



**UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS  
DEPARTAMENTO DE INGENIERIA ELECTRICA**

**“ESTRATEGIAS DE CONTROL PREDICTIVO HÍBRIDO Y SU APLICACIÓN AL  
RUTEO DINÁMICO DE VEHÍCULOS”**

**TESIS PARA OPTAR AL GRADO DE MAGISTER EN CIENCIAS DE LA  
INGENIERÍA, MENCION ELECTRICA**

**MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL ELECTRICISTA**

**ALFREDO ANTONIO NÚÑEZ VICENCIO**

**PROFESOR GUÍA:  
SRA. DORIS SÁEZ H.**

**PROFESOR CO-GUÍA:  
SR. CRISTIAN EDUARDO CORTÉS C.**

**MIEMBRO DE LA COMISIÓN:  
SR. ALDO CIPRIANO Z.**

**SANTIAGO DE CHILE  
2007**

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE MAGISTER  
EN CIENCIAS DE LA INGENIERÍA MENCIÓN ELÉCTRICA Y TÍTULO  
DE INGENIERO CIVIL ELECTRICISTA  
POR: ALFREDO ANTONIO NÚÑEZ VICENCIO  
FECHA: 25/05/2007  
PROFESOR GUÍA: SRA. DORIS SÁEZ H.  
PROFESOR CO- GUÍA: SR. CRISTIÁN CORTÉS C.

### **“Estrategias de control predictivo híbrido y su aplicación al ruteo dinámico de vehículos”**

En esta tesis se presenta una formulación analítica para el desarrollo de metodologías generales de solución utilizando estrategias de Control Predictivo Híbrido (HPC.) Se plantea esquemas de solución para el problema HPC con variables manipuladas discretas y para el problema HPC con variables manipuladas y de estado discretas. Estos se resuelven utilizando herramientas de optimización no lineal y en forma original con algoritmos genéticos.

Posteriormente, las técnicas propuestas son ejemplificadas para el control de temperatura de un estanque con calefactores on/off con resultados experimentales (variable manipulada entera) y un problema de control de nivel de un sistema de estanques cónico y cilíndrico con válvulas on/off y una filtración a una altura fija (variable de estado y manipulada entera.)

Finalmente, la última aplicación y la más importante, es el desarrollo de una estrategia HPC para el problema de programación operacional dinámica de itinerarios y rutas de vehículos en un sistema de transporte urbano de pasajeros con servicio puerta a puerta, que incluye variables manipuladas y de estado enteras. Se presenta el esquema de solución que incorpora un modelo del proceso basado en variables de estado (con variables discretas: tiempo de llegada a las paradas y ocupación de los vehículos) y restricciones en el espacio de búsqueda de soluciones según condiciones operacionales. Se generaliza la metodología de solución de control predictivo híbrido, incluyendo técnicas de clasificación difusa para determinar patrones de requerimientos según la hora del día en que se realiza el control. Los resultados comparativos se obtienen utilizando enumeración explícita y algoritmos genéticos y se desarrolla en detalle el procedimiento de sintonía de los controladores diseñados.

En un trabajo futuro se podría incorporar la modelación de la velocidad del sistema de transporte de manera de poder implementar un sistema de detección y aislamiento de fallas para problemas de congestión de tráfico.

*A mis familiares que me han estado apoyando desde el cielo y a mi Lita cariñosa.*

## **AGRADECIMIENTOS**

Quiero agradecer a quienes hicieron posible el desarrollo de esta tesis. Primero están mis padres Leticia y Guillermo que gracias a ellos pude realizar mis estudios universitarios y me apoyaron durante toda la realización de éstos. Luego a mi profesora guía Doris Sáez quien me dio la oportunidad de trabajar con ella en temas muy interesantes de investigación, me entregó muchas facilidades para realizar mi tesis y lo que más le agradezco es haber podido conocerla y todo lo que compartimos en lo cotidiano, lo cual me sirvió de experiencia en lo profesional y de ejemplo en lo personal y para futuro. También le agradezco a mi hermana Natania que me ayudó mucho cuando estaba colapsado y a mi hermano Guillermo que me ayudó sobretodo a principio de la carreta. A mis compañeros de laboratorio Roberto Zuñiga, Freddy Milla, Pablo Medina, Eduardo Sáez y Juan Solís con los cuales discutimos muchas ideas e hicieron mucho más agradable estos últimos años de carrera. Al profesor Cristián Cortés por guiarme en esta tesis y sobre todo por la buena onda y por tener justo las palabras indicadas de apoyo en momentos en que más las necesitaba. Al profesor Raúl Manasevich quien me guió como docente y me ha servido de ejemplo con su tremenda trayectoria y en la forma de guiar y relacionarse con la gente, además de su apoyo en muchas ocasiones.

Agradezco a Andrés Chong con quien empezamos en el tema de Control Predictivo Híbrido y publicamos un artículo en el ACCA, cuyos resultados fueron mejorados y se muestran en esta tesis. Debo agradecer también a Marcela Vizcay, Paul Baillaire y Ariel Muñoz, con quienes durante su práctica profesional se logró controlar y tomar datos experimentales del control de temperatura con los calefactores on-off. También a Felipe Murcia quien hizo el link con el grupo de transporte. Gracias también al profesor Igor Skrank y a Simon Oblak de la University de Ljubjana con quienes se obtuvieron los resultados mostrados en la identificación y control basado en modelos difusos para sistemas híbridos, trabajo que presenté en Vancouver en el World Congress Conference on Computational Intelligence.

También quiero agradecerle a Ximena Chacón sus palabras cuando pensaba cambiarme de especialidad y me motivó continuar en eléctrica, a Carlitos Aravena, Felipe Rios y

Claudio Garreton, compañeros de estudio en el plan común eléctrico y amigos, a Ian Pelissier por la buena onda y a los profesores Manuel Duarte, Guillermo González, Héctor Augusto, Juan Carlos Travieso, Fernando de Mayo y Aldo Cipriano quienes junto a la profesora Doris Sáez me enseñaron lo que era el área de Control Automático.

Y bueno, a todos los otros amigos que tuve en algún momento y que compartimos mucho, que no están relacionados con la tesis, pero que sin ellos mi vida hubiera sido monótona y aburrida, Fabiana, Marité, Marcelo y Larry, José Valiente, Gabriel Espejo, {Loreto, Paola, Paulina, Rodrigo}, Rigo, Leo, Oscarín, Cristian Cabello, Carola Méndez, Carmina, {muchos otros}, Adrianita, PoZeida, Zorro, Renata, Feña, Alejandra (viva Ecuador!), Cristi Oyarzún, Don Carlos Suazo, Johanna Monteiro, Loreto Boitano, y los que no me acuerde jeje.

También debo agradecer a Fondecyt y sus proyectos 1040698 "Sistemas de control predictivo híbrido con variables continuas y cuantizadas", 1030700 "Análisis, diseño y evaluación de un sistema personalizado de transporte público de alta cobertura: Adaptación al caso de la ciudad de Santiago". Además agradezco al Instituto Milenio "Complex Engineering System".

Finalmente pero no último, a Dios.

## INDICE

<b>1. INTRODUCCIÓN.....</b>	<b>8</b>
<b>2. CONTROL PREDICTIVO HÍBRIDO (HPC).....</b>	<b>10</b>
<b>2.1 Revisión bibliográfica.....</b>	<b>10</b>
<b>2.2 Formulación general del problema de Control Predictivo Híbrido .....</b>	<b>12</b>
<b>2.3 Control Predictivo Híbrido basado en modelos lineales .....</b>	<b>14</b>
<i>Ejemplo: HPC lineal con cuantización en variables manipuladas.....</i>	<i>16</i>
<b>2.4 Control Predictivo Híbrido basado en modelos difusos.....</b>	<b>20</b>
<i>Ejemplo: Control predictivo difuso con cuantización en variables manipuladas     y en variables de estado, sistemas tipo Witsenhausen.....</i>	<i>21</i>
<b>2.5 Algoritmos de Optimización Entera Mixta. ....</b>	<b>24</b>
2.5.1 Optimización Entera Mixta Lineal.....	24
2.5.2 Optimización Entera Mixta No Lineal .....	28
2.5.3 Optimización Entera Mixta basada en Algoritmos Genéticos.....	31
<b>3.- ESQUEMAS DE SOLUCIÓN PARA CONTROL PREDICTIVO HÍBRIDO.....</b>	<b>35</b>
<b>3.1 HPC basado en modelos lineales, función objetivo cuadrática y variable   manipulada entera. Aplicación: Control de Temperatura de un estanque .....</b>	<b>35</b>
3.1.1 Descripción del proceso .....	35
3.1.2 Base de evaluación.....	37
3.1.3 Resultados por simulación.....	37
3.1.4 Resultados experimentales.....	43
3.1.5 Síntesis y conclusiones.....	45
<b>3.2 HPC basado en modelos difusos, función objetivo cuadrática y variable   manipulada y de estado entera. Aplicación: Control de nivel de un sistema de   estanques híbridos. ....</b>	<b>46</b>
3.2.1 Descripción del proceso .....	46
3.2.2 Modelación Difusa para el sistema híbrido.....	47
3.2.3 Algoritmo de solución.....	51
3.2.4 Resultados por simulación.....	55
3.2.5 Síntesis y conclusiones.....	60
<b>4.- APLICACIÓN AL PROBLEMA DE RUTEO DINÁMICO DE VEHICULOS.....</b>	<b>61</b>
<b>4.1 Revisión bibliográfica.....</b>	<b>61</b>
<b>4.2 Descripción del problema.....</b>	<b>66</b>
<b>4.3 Control predictivo híbrido para el ruteo de vehículos.....</b>	<b>71</b>
4.3.1 Modelo del proceso (variables de estado tiempo y carga) .....	72

4.3.2 <i>Función Objetivo y Horizontes de predicción</i> .....	82
4.3.3 <i>Zonificación difusa</i> .....	88
4.3.4 <i>Análisis del espacio de búsqueda de soluciones y restricciones</i> .....	93
<b>4.4 Algoritmos de Optimización</b> .....	<b>95</b>
4.4.1 <i>Optimización basada en Algoritmos Genéticos (GA)</i> .....	96
4.4.2 <i>Aplicación de GA para el problema de ruteo dinámico</i> .....	97
4.4.3 <i>Ejemplo por simulación</i> .....	100
<b>4.5 Pruebas por simulación</b> .....	<b>105</b>
4.5.1 <i>Zonificación difusa</i> .....	106
4.5.2 <i>HAPC basado en GA y zonificación difusa</i> .....	109
4.5.3 <i>Disminución de áreas en zonificación clásica</i> .....	111
4.5.4 <i>Síntesis y conclusiones</i> .....	118
<b>5. CONCLUSIONES</b> .....	<b>119</b>
<b>REFERENCIAS</b> .....	<b>122</b>
<b>ANEXO: PUBLICACIONES GENERADAS</b> .....	<b>130</b>
ANEXO 1: DISEÑO DE ESTRATEGIAS DE CONTROL PREDICTIVO HIBRIDO Y SU APLICACIÓN AL CONTROL DE TEMPERATURA DE UN ESTANQUE DE NIVEL.....	131
ANEXO 2: PROBLEMA DE RUTEO DINÁMICO DE UNA FLOTA DE VEHÍCULOS CON UN ENFOQUE DE CONTROL PREDICTIVO HÍBRIDO BASADO EN ALGORITMOS GENÉTICOS. ....	139
ANEXO 3: HYBRID PREDICTIVE CONTROL BASED ON FUZZY MODEL.....	154
ANEXO 4: HYBRID ADAPTIVE PREDICTIVE CONTROL FOR THE MULTI-VEHICLE DYNAMIC PICKUP AND DELIVERY PROBLEM BASED ON GENETIC ALGORITHMS AND FUZZY CLUSTERING. ....	163
ANEXO 5: HYBRID PREDICTIVE CONTROL FOR A REAL-TIME ROUTED TRANSIT SYSTEM.....	203
ANEXO 6. CONTROL PREDICTIVO HIBRIDO PARA UN SISTEMA PERSONALIZADO DE TRANSPORTE PÚBLICO PUERTA A PUERTA PROGRAMADO EN TIEMPO REAL .....	241

## 1. INTRODUCCIÓN

El objetivo de esta tesis es presentar resultados y metodología para aplicar el enfoque de Control Predictivo Híbrido (HPC) al problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta.

La modelación del problema basada en variables de estado, la función objetivo flexible que permite incorporar predicciones, el algoritmo de solución usado y los resultados obtenidos por simulación, han sido un aporte para el área de Ingeniería Civil en Transporte. (Ver en anexo publicaciones generadas).

Sin embargo, la metodología empleada es genérica y se puede aplicar no solo a problemas de transporte, los cuales de manera muy natural presentan variables importantes enteras, sino que a procesos industriales. De hecho, la mayoría de estos procesos contienen componentes de naturaleza discreta/entera, y en la actualidad, la teoría de Control Predictivo Híbrido (HPC), aún en desarrollo, ha logrado resultados promisorios.

La tesis se presenta de tal manera que la aplicación en transporte sea consecuencia de esta metodología genérica. Para cumplir ese objetivo, en los primeros capítulos se presenta la metodología y algunos ejemplos ilustrativos, y finalmente, desde el enfoque de Control Predictivo Híbrido se muestra la solución al problema de ruteo dinámico.

En el Capítulo 2 se presenta el problema de Control Predictivo General y a partir de esta formulación se muestran algunos casos importantes como lo son HPC basado en modelos lineales y HPC para procesos no lineales basado en modelos difusos. Al final del capítulo se realiza una breve revisión de algoritmos de optimización entera mixta.

En el Capítulo 3 se aplica la metodología sistemática de resolución de HPC a los problemas de control de temperatura de un estanque con tres calefactores on/off (variable manipulada entera) y al problema de control de nivel de un sistema de estanques no lineal híbrido (variable manipulada y variable de estado discreta). Algunos de los resultados de estos ejemplos han sido presentados en congresos, debido al aporte en la innovación en



el algoritmo de optimización empleado (Algoritmos Genéticos). Los resultados experimentales que se obtuvo, escasos en la literatura actual, son una prueba que valida el método.

El Capítulo 4 muestra en detalle la solución basada en HPC para el problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta. Se emplean Algoritmos Genéticos en la optimización entera mixta no lineal y clustering difuso para predicción de demanda. Los modelos están en variables de estados y se presentan resultados por simulación.

Como anexo se reportan las publicaciones generadas como parte de esta tesis.

Finalmente, cabe mencionar que esta tesis es fruto del esfuerzo del grupo de Control en Transporte de la Universidad de Chile, que desde el año 2004 se han dedicado a explotar este nicho de investigación.

## 2. CONTROL PREDICTIVO HÍBRIDO (HPC)

### 2.1 Revisión bibliográfica

En esencia todos los procesos industriales contienen componentes continuas y discretas dados por ejemplo por válvulas on/off, switches, overrides lógicos, etc. Estos procesos que involucran tanto variables enteras como continuas ya sea en el estado o en las entradas se denominan sistemas híbridos. Los sistemas híbridos han sido ampliamente estudiados en los últimos años debido a la necesidad de nuevas herramientas que permitan abordar sus problemas relacionados como son la identificación de sistemas híbridos, el control, la detección de falla, etc.

En cuanto al control, una técnica que ha resultado exitosa es la de Control Predictivo Híbrido, en la cual se minimiza una función objetivo sujeto a las restricciones del modelo el cual incorpora las características híbridas del proceso, es decir, sus variables enteras.

Slupphuang *et al.* (1997) y Slupphaung y Foss (1997) presentan el diseño de control predictivo con variables de entrada continuas y enteras. El controlador predictivo con entradas continuas y enteras fue sintonizado usando programación entera mixta no lineal. En este caso, se resuelve el problema utilizando programación no lineal entera mixta y se evalúa favorablemente frente a una estrategia de control predictivo con separación de las variables continuas de las variables enteras. La estrategia propuesta la aplicaron por simulación al control de nivel y temperatura de un sistema de estanques.

Bemporad y Morari (1999, 2000) presentan un esquema de control predictivo híbrido que incluye restricciones operacionales. En este caso, el problema se resuelve usando programación entera mixta cuadrática (MIQP). Los algoritmos propuestos se aplican por simulación a un sistema de turbinas, el cual incorpora variables manipuladas enteras. El principal problema de MIQP es la complejidad computacional que aumenta el tiempo necesario para encontrar una solución.

Bemporad *et al.* (2002) establecen que un sistema híbrido con un controlador predictivo basado en una función objetivo cuadrática y restricciones lineales es una subclase de

sistema híbrido MLD (“Mixed Logic Dynamical”). En otras palabras, el sistema en lazo cerrado está caracterizado como un sistema híbrido.

Thomas *et al.* (2004) describe un algoritmo de Control Predictivo Híbrido basado en partición poliedral del espacio de estado. Proponen dividir el espacio de estado en regiones poliedrales en donde ocupan un modelo MLD simplificado para describir la dinámica en cada región. Con este enfoque busca disminuir el tiempo computacional y lo aplican al control por simulación de un sistema de estanques de nivel.

Becutti *et al.* (2004) proponen descomposición Lagrangeana temporal para resolver el problema de HPC, en donde ocupando propiedades de dualidad, trasladan el problema original a una secuencia de problemas de menor dimensión. Esta solución es heurística en el sentido que no logra el óptimo global, sin embargo se aproxima bastante a este.

Potočnik *et al.* (2004) proponen un algoritmo para control predictivo híbrido con entradas enteras basado en análisis de alcanzabilidad. El tiempo computacional es reducido por la construcción y corte de un árbol de evolución.

Las estrategias de HPC más usadas involucran dos algoritmos de optimización: Enumeración Explícita (EE) y Branch and Bound (BB). Ambos permiten resolver problemas de optimización entera-mixta (MIOP) (Floudos, 1995), pero el elevado esfuerzo computacional, especialmente en el caso de EE, hace que sean ineficientes para resolver problemas en tiempo real. Como solución a esto, en esta tesis se proponen Algoritmos Genéticos (GA) como herramienta eficiente para resolver el problema de optimización entera mixta no lineal.

En cuanto al problema de identificación híbrida, recientemente se han propuesto algunos métodos. Palm y Driankov (1998) presentan una identificación jerárquica obteniendo un modelo difuso con switches. El método propuesto considera una identificación difusa tipo caja negra usando clustering difuso y algunos conocimientos a priori sobre los estados discretos; sin embargo, el último paso no es generalizable. Luego, Girimonte y Babuska (2004) describieron dos métodos de selección de estructura para modelos no lineales con entradas discretas y continuas. El primer método se basa en clustering difuso, usando conjuntos difusos para obtener las entradas relevantes. El segundo método es un

algoritmo de inducción incluido en un método de búsqueda. Los resultados muestran que el clustering difuso es más rápido en términos de tiempo computacional. Sin embargo, la desventaja del método es el aumento del tiempo computacional en la medida que se aumenta el horizonte de búsqueda.

En esta tesis se propone ocupar un modelo difuso para realizar las predicciones del sistema híbrido en el esquema de control. Como el problema de optimización asociado al caso de control predictivo difuso híbrido (HFPC) es altamente no lineal, se emplea algoritmos genéticos para resolverlo. Sarimveis (2003) usa algoritmos genéticos en control predictivo difuso sin variables enteras y prueba que la técnica provee soluciones razonables en un tiempo computacional reducido.

A continuación, se presenta la formulación del problema HPC general y los casos:

- HPC con entrada entera.
- HFPC con variables de estado continuas y enteras, y entrada entera.

En el capítulo 3 se muestran resultados por simulación y experimentales en el control de dos sistemas adecuados para cada caso. En los ANEXO 1 y ANEXO 3 se encuentran las publicaciones de los resultados por simulación.

## **2.2 Formulación general del problema de Control Predictivo Híbrido**

Los sistemas híbridos se pueden denominar como sistemas dinámicos lógicos mixtos (MLD) que describen estas componentes híbridas como un sistema de ecuaciones diferenciales sujeto a inecuaciones que involucran variables binarias y continuas. Estas variables pueden ser estados y/o variables de entrada del sistema.

Los sistemas híbridos MLD se pueden definir en general como:

$$\begin{cases} x(t+1) = f(x(t), u(t), \delta(t), z(t)) \\ y(t) = g(x(t), u(t), \delta(t), z(t)) \\ h(u(t), \delta(t), z(t), x(t)) \leq 0 \end{cases} \quad (1)$$

Donde  $x(t) = [x_c^T(t), x_l^T(t)] \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$  son los estados continuos y binarios,  $u(t) = [u_c^T(t), u_l^T(t)] \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$  son las entradas o variables manipuladas continuas y binarias,  $y(t) = [y_c^T(t), y_l^T(t)] \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$  son las salidas continuas y binarias, y  $\delta(t) \in \{0,1\}^n$ ,  $z(t) \in \mathbb{R}^c$  representan variables auxiliares binarias y continuas, respectivamente. Las funciones generales  $f()$ ,  $g()$  y  $h()$  son las que definen las ecuaciones de estado, salida y restricciones del sistema en variables de estado. Las ecuaciones de estado están incluidas en el primer término de (1), la salida en el segundo y las restricciones sobre el estado, entradas y variables  $z(t)$  y  $\delta(t)$  son incluidas en el tercer término.

Para el diseño de un Control Predictivo Híbrido se necesita una función objetivo que represente de buena manera los objetivos de control del proceso. El controlador calculará las acciones de control que minimizan la función objetivo, que en general se puede escribir como:

$$\min_{\{u(t), u(t+1), \dots, u(t+N_u-1)\}} J = J(\bar{u}, \bar{\delta}, \bar{z}, \bar{x}, \bar{y}) \quad (2)$$

$J$  es cualquier función objetivo multivariable cuyas variables son los vectores:

$$\begin{aligned} \bar{u} &= [u(t), u(t+1), \dots, u(t+N_u-1)]^T \\ \bar{\delta} &= [\delta(t+1), \dots, \delta(t+N)]^T \\ \bar{z} &= [z(t+1), \dots, z(t+N)]^T \\ \bar{x} &= [x(t+1), x(t+2), \dots, x(t+N)]^T \\ \bar{y} &= [y(t+1), \dots, y(t+N)]^T \end{aligned} \quad (3)$$

$t$  es el instante de tiempo actual,  $x(t)$  el estado actual,  $N$  el horizonte de predicción,  $\{u(t), \dots, u(t+N_u-1)\}$  la secuencia de control que minimiza la función de costos (2) sujeta a (1),  $N_u$  el horizonte de control,  $u(t+k)$  es constante para  $k \geq N_u$ .

### 2.3 Control Predictivo Híbrido basado en modelos lineales

Los sistemas híbridos MLD, pueden ser modelados como un sistema de ecuaciones lineales sujeto a inecuaciones que involucran variables binarias y continuas. A partir del caso general (1), escogiendo  $f()$  y  $g()$  lineales y  $h()$  función lineal afín, se puede llegar a la expresión para el caso lineal propuesta por Bemporad y Morari (1999):

$$\begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ -E_5 \leq E_1u(t) - E_2\delta(t) - E_3z(t) + E_4x(t) \end{cases} \quad (4)$$

Donde  $x(t) = [x_c^T(t), x_l^T(t)] \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$  son los estados continuos y binarios,  $u(t) = [u_c^T(t), u_l^T(t)] \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$  son las entradas o variables manipuladas continuas y binarias,  $y(t) = [y_c^T(t), y_l^T(t)] \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$  son las salidas continuas y binarias, y  $\delta(t) \in \{0,1\}^n, z(t) \in \mathbb{R}^r$  representan variables auxiliares binarias y continuas, respectivamente.  $A, B_1, B_2, B_3, C, D_1, D_2, D_3, E_1, E_2, E_3, E_4$  y  $E_5$  son matrices que forman las ecuaciones y restricciones del sistema en variables de estado. Las ecuaciones de estado están incluidas en el primer término de (1), la salida en el segundo y las restricciones sobre el estado, entradas y variables  $z(t)$  y  $\delta(t)$  se incluyen en el tercer término.

El Control Predictivo Híbrido puede ser diseñado basado en la minimización de cualquier función objetivo que sea coherente con los objetivos del controlador. En general la función

objetivo  $J$  puede ser multivariable como se muestra (2) y el problema de optimización tendrá como vectores de variables los mostrados en (3). El algoritmo de optimización entregará una secuencia de acciones de control:

$$\vec{u} = [u(t), u(t+1), \dots, u(t+N_u-1)]^T \quad (5)$$

de la cual, según el horizonte deslizante, solo se extrae y se aplica al sistema su primera componente  $u(t)$  la cual representa a la acción de control adecuada que lleva al sistema a su siguiente estado  $x(t+1)$  minimizando el funcional objetivo (2).

Para propósitos de regulación y seguimiento de referencia, se plantea la siguiente función objetivo cuadrática:

$$J = \sum_{k=0}^{N_u-1} \|u(t+k) - u_e(t+k)\|_{Q_1}^2 + \sum_{k=1}^N \|\delta(t+k) - \delta_e(t+k)\|_{Q_2}^2 + \|z(t+k) - z_e(t+k)\|_{Q_3}^2 + \|x(t+k+1) - x_e(t+k+1)\|_{Q_4}^2 + \|y(t+k) - y_e(t+k)\|_{Q_5}^2 \quad (6)$$

Donde  $u_e, \delta_e, z_e, x_e$  y  $y_e$  son vectores con valores de las variables en el punto de equilibrio o valores de trayectorias de referencia. El operador satisface  $\|\vec{h}\|_{Q_n}^2 = (\vec{h})^T \cdot Q_n \cdot \vec{h}$ ,  $Q_1, Q_2, Q_3, Q_4$  y  $Q_5$  son matrices de peso.

El resultado de la optimización de la función objetivo (6) sujeto a las restricciones del modelo (4) dará como resultado  $\vec{u}^* = [u^*(t), u^*(t+1), \dots, u^*(t+N_u-1)]^T$  que representa a la secuencia de acciones de control que mueven el estado desde la condición inicial  $x(k)$  al estado de equilibrio o a la trayectoria  $x_e$  y minimiza la función de costos (6) sujeta a (4). Según el horizonte deslizante, solo se extrae y se aplica al sistema su primera componente  $u^*(t)$  la cual representa a la acción de control adecuada que lleva al sistema a su siguiente estado  $x(t+1)$  minimizando el funcional objetivo (6).

En resumen, en el diseño de una estrategia de control predictivo híbrido, tanto variables manipuladas como estados pueden ser enteros. Estos problemas de optimización son difíciles de resolver analíticamente y se han desarrollado muchos métodos numéricos tanto analíticos (enumeración explícita, branch and bound, descomposición de bender, etc.) como heurísticas (algoritmos genéticos, enjambre de partículas, etc.) para resolver problemas específicos.

### **Ejemplo: HPC lineal con cuantización en variables manipuladas**

En este caso se considera que las variables de estado del sistema son continuas  $x(t) = [x_c^T(t)] \in \mathbb{R}^{n_c}$  y que la variable manipulada puede tomar valores tanto continuos como binarios  $u(t) = [u_c^T(t), u_l^T(t)] \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ . No se consideró necesario incluir en la salida los valores de las entradas aplicadas, y que esta puede indicar algunos valores de variables de estado. Por lo tanto la salida sólo tomará valores continuos  $y(t) = [y_c^T(t)] \in \mathbb{R}^{p_c}$ .

En una primera aproximación se abordó una familia de problemas que satisfacen las siguientes ecuaciones para las matrices definidas para (3):

$$B_2 = B_3 = E_2 = E_3 = 0 \quad (\text{No hay variables auxiliares})$$

$$D_1 = D_2 = D_3 = 0 \quad (\text{La salida será una variable de estado})$$

La ecuación (1) quedará expresada por:

$$\begin{cases} x(t+1) = Ax(t) + B_1u(t) \\ y(t) = Cx(t) \\ -E_5 \leq E_1u(t) + E_4x(t) \end{cases} \quad (7)$$

Donde el primer termino es la ecuación en variables de estado, el segundo la salida del sistema y el tercero restricciones de operación tanto para la variable manipulada como



para las variables de estado (entre ellos también pueden incorporarse restricciones para la salida).

El controlador predictivo generalizado (GPC: "Generalized Predictive Controller") propuesto por Clarke (1987) se diseña considerando un modelo ARIX del proceso ("AutoRegressive Integrated with exogenous variable"). A partir de la ecuación (7) y tomando transformada zeta, se tiene que la función de transferencia entre la salida y la entrada esta dada por:

$$G(z) = \frac{y(z)}{u(z)} = C(z \cdot I - A)^{-1} B_1 \quad (8)$$

$G(z)$  puede ser una matriz según el número de salidas y entradas que tenga el sistema. Cada una de sus componentes será una función de transferencia SISO que representará la respuesta al impulso cuando se consideran todas las entradas y salidas nulas con excepción de la entrada y salida correspondiente a la fila y columna que se tome. Se considera el caso en que hay una sola salida ( $p_c=1$ ) y una sola entrada ( $m_c=0$  y  $m_l$  libre), obteniéndose el siguiente modelo ARIX:

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + \frac{e(t)}{\Delta}$$

Donde  $A(z^{-1})$  y  $B(z^{-1})$  son polinomios,  $G(z) = \frac{B(z^{-1})}{A(z^{-1})}$ ,  $\Delta = 1 - z^{-1}$ ,  $y(t)$  es la variable controlada,  $u(t)$  es la variable manipulada y  $e(t)$  es ruido blanco con esperanza igual a cero.

La salida predicha a  $j$  pasos, considerando las señales de salida hasta  $t$  y las acciones de control futuras, está dada por:

$$y_p(t+j) = G_j \Delta u(t+j-1) + F_j y(t)$$

con  $G_j = E_j(q^{-1})B(q^{-1})$ ,  $E_j$  y  $F_j$  polinomios únicos definidos por el polinomio  $A$  y el intervalo de predicción  $j$ .

El algoritmo de Control Predictivo Generalizado (GPC), calcula las acciones de control futuras, minimizando la siguiente función de costos:

$$J = \sum_{k=N_1}^N \left( \hat{y}(t+k) - \bar{r}(t+k) \right)^T \cdot \left( \hat{y}(t+k) - \bar{r}(t+k) \right) + \lambda \sum_{k=N_1}^{N_u} \left( \Delta \bar{u}(t+k-1) \right)^T \cdot \left( \Delta \bar{u}(t+k-1) \right) \quad (9)$$

Donde  $N_1$  y  $N$  corresponden a los horizontes de predicción,  $N_u$  es el horizonte de control,  $\hat{y}(t+k)$  es el vector de predicción a  $k$  pasos para la variable controlada,  $\bar{r}(t+k)$  es el vector de referencias futuras y  $\lambda$  es un factor de peso. Esta función objetivo es un caso particular de la función objetivo planteada en (6), donde las matrices ajustadas según la dimensión de cada variable son:

$$Q_1 = \lambda \cdot T_{(m_c+m_l+1) \times (m_c+m_l+1)}$$

$$Q_2 = 0_{\eta \times \eta}$$

$$Q_3 = 0_{r_c \times r_c}$$

$$Q_4 = 0_{(n_c+n_l) \times (n_c+n_l)}$$

$$Q_5 = I_{(p_c+p_l) \times (p_c+p_l)}$$

La matriz  $Q_1$  genera las diferencias en la variable manipulada mediante la función matricial  $T$  (un ejemplo se mostrará a continuación) y las dimensiones de las matrices están dadas por las dimensiones de las variables de estado ( $n_c$ ), de entrada ( $m_c \times m_l$ ) y de salida ( $p_c$ ). Por facilidad se incorpora dentro del vector de acciones de control a  $u(t-1)$  que es una variable conocida y necesaria para calcular  $\Delta u(t)$ . Para un ejemplo donde el horizonte de control es 9, la matriz  $Q_1$  de transformación tiene la forma:

$$Q_1 = \lambda \cdot T_{(m_c+m_f+1) \times (m_c+m_f+1)} = \lambda \cdot T_{10 \times 10} = \lambda \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Con estas matrices la función objetivo queda de la forma:

$$\min_{\{u(t), u(t+1), \dots, u(t+N_u-1)\}} J = \sum_{k=1}^{N_u} \|u(t+k-1) - u_e\|_{Q_1}^2 + \sum_{t=1}^N \|y(t+k) - y_e\|_{Q_2}^2 \quad (10)$$

En la práctica, se requiere mantener las variables del proceso en rangos que aseguren el buen comportamiento de los equipos y evitar situaciones críticas. Por ejemplo, los actuadores tienen restricciones de límite y velocidad. Además, los puntos de operación del proceso están determinados por objetivos económicos, que usualmente llevan al proceso a operar cerca de las restricciones. Los sistemas de control, en especial, los sistemas de control predictivo se anticipan a estas restricciones y corrigen las acciones de control.

Las restricciones típicas son de rango en la variable manipulada, en la variación de la variable manipula o en el rango de las variables controladas, es decir:

$$\begin{aligned} u_{\min} &\leq u(t+i-1) \leq u_{\max} \\ \Delta u_{\min} &\leq \Delta u(t+i-1) \leq \Delta u_{\max} \\ y_{\min} &\leq \hat{y}(t+j) \leq y_{\max} \\ j &= N_1, \dots, N_2 \text{ e } i = N_1, \dots, N_u. \end{aligned} \quad (11)$$

Estas restricciones suelen ser incorporadas en el modelo, ocupando el tercer término de la ecuación en variables de estado (7).

Finalmente, utilizando algoritmos de programación cuadrática u otro algoritmo especializado, se resuelve el problema de optimización en el cual se minimiza la función objetivo (10), sujeta a las restricciones dadas por las predicciones y las restricciones definidas en (11) (Tsang y Clarke, 1988). En la Figura 1 se muestra un esquema del funcionamiento de una planta híbrida con el controlador propuesto.

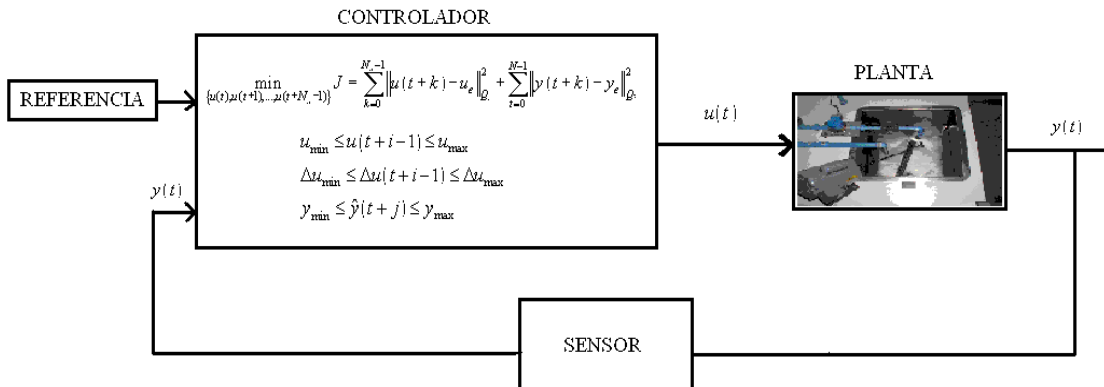


FIGURA 1. ESQUEMA LAZO CERRADO CON CONTROL PREDICTIVO HÍBRIDO.

## 2.4 Control Predictivo Híbrido basado en modelos difusos

La estrategia Control Predictivo Híbrido basado en modelo difusos (HFPC) es una generalización del control predictivo basado en modelos (MPC) en donde el modelo predicción incorpora tanto las variables discretas/enteras como continuas (Núñez et al 2006). No se encontraron otras referencias respecto a este tema puntual, pero en la literatura se puede hallar una amplia gama de trabajos en el tema de Control Predictivo basado en modelos difusos. Una completa recopilación y aportes en este tema se encuentra en las tesis de magíster de D. Sáez (1995) en donde se estudiaron diversos algoritmos de control, entre ellos: control lineal cuadrático realimentado en el estado, control predictivo generalizado, y control basado en un modelo difuso. La principal contribución de esa tesis es la derivación original de dos algoritmos de control basados en modelos difusos: el control lineal cuadrático difuso y el control predictivo generalizado difuso.

Se propone a continuación ocupar un control predictivo híbrido basado en algún modelo difuso que incluya las características híbridas del sistema y que sea capaz de resolver el siguiente problema de optimización entera mixta no lineal.

$$\min_{\{u(k), u(k+1), \dots, u(k+N_u-1)\}} J = J_1 + \lambda J_2$$

$$J_1 = \sum_{j=N_1}^{N_y} (\hat{y}(k+j) - r(k+j))^2, J_2 = \sum_{j=N_1}^{N_u} \Delta u(k+j-1)^2 \quad (12)$$

donde  $J$  es la función objetivo,  $\hat{y}(k+j)$  corresponde a la predicción a  $j$  pasos para la variable controlada dada por el modelo difuso,  $r(k+j)$  es la referencia,  $\Delta u(k+j-1)$  es el incremento de la acción de control y  $\lambda$  es un factor de peso.  $N_1$ ,  $N_y$  y  $N_u$  son el horizonte de predicción y de control respectivamente. El resultado de la optimización será una secuencia de acciones de control  $\{u(k), \dots, u(k+N_u-1)\}$ .

El problema de identificación difusa del sistema híbrido puede ser no trivial. Si bien no es el objetivo de esta tesis indagar en el problema de identificación, se presenta a continuación una alternativa para el caso de control de sistemas tipo Witsenhausen.

Una vez resuelto el problema de identificación, es posible resolver el problema HFPC con variable manipulada entera ocupando Algoritmos Genéticos tal y como se muestra en el capítulo 2.2 mediante un ejemplo de control de estanques híbridos.

**Ejemplo: Control predictivo difuso con cuantización en variables manipuladas y en variables de estado, sistemas tipo Witsenhausen.**

En esta tesis se abordan sistemas con estados que presentan variables enteras y continuas, en los cuales el estado continuo se mantiene continuo incluso cuando los estados discretos cambian. La transición del sistema ocurrirá cuando uno o más estados continuos satisfagan condiciones definidas para cada transición. Este tipo de sistemas híbridos se conocen como del tipo Witsenhausen (Witsenhausen, 1966). Se describe en forma general como:

$$\begin{aligned}x_{k+1} &= f_{q_k}(x_k, u_k) \\q_k &= g(x_k, q_{k-1})\end{aligned}\tag{13}$$

donde  $x_k \in R^n$  es el vector de estado,  $u_k \in R^m$  es el vector de entradas y  $q_k \in S$  con  $S=\{1, \dots, s\}$  es el vector de estado de switching para cada región. Esto significa que los estados del sistema híbrido están descritos en cualquier instante de tiempo  $k$  por el conjunto de estados  $(x_k, q_k)$  en el dominio  $R^n \times S$ . En general el estado discreto (o de switch)  $q_k$  depende del estado  $x_k$  y del estado discreto previo  $q_{k-1}$ . El comportamiento local del sistema está descrito por las funciones  $f_{q_i}$  para cada estado discreto, y  $g$  es la función de transición entre regiones para los estados discretos o de switch.

El esquema de control de este tipo de sistema es un caso particular del caso Control Predictivo Híbrido general planteado con las ecuaciones (1).

Para efecto de identificación, los sistemas de tipo Witsenhausen pueden ser representado mediante el siguiente sistema difuso de dos niveles, los cuales fueron descritos por Tanaka *et al.* (2001). Esos dos niveles son el nivel de switch entre regiones y el nivel de modelo difuso local. La forma clásica del sistema difuso de dos niveles es:

$$\begin{aligned}x_{k+1} &= \sum_{i=1}^s \sum_{j=1}^{R_i} v_i(q_k) \beta_{ij}(z_k) (a_{ij} x_k + b_{ij} u_k + r_{ij}) \\v_i(q_k) &= \begin{cases} 1, & q_k \in S_i \\ 0, & q_k \notin S_i \end{cases} \\ \beta_{ij}(z_k) &= \frac{\prod_{r=1}^p A_{ij,r}(z_{k,r})}{\sum_{j=1}^{R_i} \prod_{r=1}^p A_{ij,r}(z_{k,r})}\end{aligned}$$

donde  $s$  es el número de regiones de switch,  $r_i$  es el número de reglas en la  $i$ -ésima región de switch  $S_i$ ,  $v_i(q_k)$  es un valor de peso que activa cada región de switch el cual es descrito por el estado de switch actual  $q_k$  y define la región actual, y  $\beta_j(z_k)$  es una función de pertenencia para cada modelo local definido por el vector de premisas

$z_k = [z_{k,1} \ z_{k,2} \ \dots \ z_{k,p}]^T$  y los conjuntos difusos locales  $A_{ij,r}$  ( $r = 1, \dots, p$ ), con  $A_{ij,k}(z_{k,j})$  el grado de pertenencia de la variable premisa ( $z_{k,j}$ ) al conjunto de pertenencia  $A_{ij,k}$ .

La estructura de las ecuaciones dinámicas para el caso de dos niveles es muy adecuada cuando las regiones de switch se conocen exactamente. Sin embargo, se busca manejar un caso genérico en el cual las regiones de switch son desconocidas y no se posee información avanzada al respecto. Se supondrá que lo único conocido son datos de entrada y salida. Entonces, se propone estructurar un modelo difuso continuo de un solo nivel e imponer la característica híbrida (regiones de switch) dentro de las funciones de pertenencias.

El modelo puede ser descrito entonces por:

$$x_{k+1} = \sum_{i=1}^{s.R_i} \mu_i(z_k) (a_i x_k + b_i u_k + r_i)$$

$$\mu_i(z_k) = \frac{\prod_{r=1}^p B_{i,r}(z_{k,r})}{\sum_{i=1}^{s.R_i} \prod_{r=1}^p B_{i,r}(z_{k,r})} \quad (14)$$

donde  $\mu_i(z_k)$  es el grado de pertenencia del producto entre el valor de peso que activa cada región de switch y la función de pertenencia del modelo local definido, que queda definido por el vector de premisas  $z_k$ . La naturaleza híbrida del sistema se incorporará entonces en la forma de las funciones de pertenencia. El criterio ocupado para su distribución está basado en el análisis de valores y vector propios de las matrices de covarianzas de los clusters, obtenida a partir del algoritmo de cluster realizado. Sean los centros de los clusters  $v_i$ , los valores propios de los cluster  $\{\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im}\}$ , y sus vectores propios  $\{\phi_{i1}, \phi_{i2}, \dots, \phi_{im}\}$ , donde los valores propios y los correspondientes vectores propios están ordenados en orden descendiente según los valores propios. Por medio del análisis de los vectores propios mas importantes (componentes principales en cuyas direcciones se entrega la mayor cantidad de información), se puede detectar la región de switching. El criterio usado es conocido como el procedimiento de unir clusters similares, el cual fue

ocupado exitosamente para detectar las regiones de switch en el caso presentado en el capítulo 2.2.

## **2.5 Algoritmos de Optimización Entera Mixta.**

Los problemas de Optimización Entera Mixta se pueden clasificar según su linealidad. De acá surgen distintos métodos para resolver problemas de Optimización Entera Mixta Lineal (MILP) que se aplican cuando tanto la función objetivo como las restricciones del problema son lineales con respecto a sus variables de optimización. Pero en general el problema de control predictivo híbrido es no lineal.

En el caso de HPC con variables manipuladas enteras y continuas, con propósitos de regulación y seguimiento y modelo de predicción ARIX (lineal), la función objetivo planteada es cuadrática y por lo tanto ese problema deberá ser resuelto ocupando algún método de Optimización Entera Mixta No Lineal (MINLP). En este caso particular es posible explotar la convexidad del problema en la medida que las restricciones operacionales sean convexas, ya que la función objetivo cuadrática es convexa al igual que las restricciones dadas por el modelo lineal.

En el caso de HFPC, tanto la función objetivo como el modelo son no lineales. Además, y en general, el problema será no convexo debido al modelo difuso y sus funciones de pertenencia.

Se presenta a continuación una breve revisión de los algoritmos más importantes de optimización entera mixta. Se muestra el caso lineal para efectos de que se comprenda la dificultad del problema, la cual se amplifica cuando se resuelve el caso no lineal.

### **2.5.1 Optimización Entera Mixta Lineal**

La formulación del MILP con variables binarias es:



$$\begin{aligned}
& \min (c^T x + d^T y) \\
& s.t. \quad Ax + By \leq b \\
& \quad x \in X \subseteq R^n \\
& \quad y \in \{0,1\}^q
\end{aligned} \tag{15}$$

Donde  $x$  es un vector de  $n$  variables continuas,  $y$  es un vector de  $q$  variables binarias,  $c$  y  $d$  son vectores de parámetros de tamaño  $(n \times 1)$  y  $(q \times 1)$  respectivamente,  $A$  y  $B$  son matrices constantes de dimensiones apropiadas y  $b$  es un vector de  $p$  desigualdades.

La mayor dificultad de los problemas MILP se deben a la naturaleza combinatorial del dominio de las variables  $y$ . Cualquier selección de 0 o 1 para los elementos del vector  $y$  resulta un problema de programación lineal LP para las variables  $x$  el cual puede ser resuelto y se puede encontrar su óptimo global.

Una opción para resolver este problema es seguir la técnica de enumeración explícita (EE), enumerando todas las posibles combinaciones de variables binarias para los elementos del vector  $y$ . Desafortunadamente, el tiempo computacional con esa técnica crece exponencialmente al aumentar el número de variables binarias. Por ejemplo, si consideramos 100 variables binarias, el número de combinaciones es de  $2^{100}$ . Como resultado podríamos tener que resolver  $2^{100}$  LP. Entonces la técnica de enumeración explícita se puede tornar prohibitiva para problemas con muchas variables binarias.

Se ha realizado análisis de complejidad para muchos tipos de problemas de programación entera mixta. En particular, el problema (15) pertenece a la clase de problemas NP-Hard, que son problemas cuyo espacio de soluciones factible crece combinatorialmente al agregar una nueva variable de optimización. En un problema de Control Predictivo Híbrido es común variar el horizonte de control. Cuando este horizonte aumenta, se agrega variables de optimización entera al problema por lo que crece combinatorialmente.

A pesar de la naturaleza combinatorial de los modelos MILP de la forma (15), se han propuesto y aplicado exitosamente muchos algoritmos en problemas de aplicación de mediana y larga escala.

Los algoritmos propuestos, se pueden clasificar como:

- (i) Métodos de Branch and Bound, donde se emplea un árbol de combinaciones 0-1 y la región factible de solución es particionada en subdominios sistemáticamente y se generan cotas superiores e inferiores a diferentes niveles del árbol con el fin de aplicar ciertos criterios para no recorrer el árbol completo.
- (ii) Métodos de Planos Cortantes, donde la región factible no es dividida en subdominios sino que se crean nuevas restricciones (cuts) de manera de reducir la región factible hasta encontrar una solución binaria óptima.
- (iii) Métodos de descomposición, donde la estructura matemática es explotada mediante partición de las variables, dualidad y métodos de relajación.
- (iv) Métodos basados en lógica, donde técnicas de inferencia simbólica o restricciones disyuntivas son utilizadas para expresarlas en términos de variables binarias.

Dentro de las cuatro clases de algoritmos, el método de Branch and Bound será explicado con mayor detalle.

### **Método de Branch and Bound**

El problema (15) se puede resolver utilizando el método de optimización Branch and Bound (Floudas, 1995).

En este método, se aplica reglas para evitar enumerar todas las combinaciones posibles de las variables enteras o binarias y se basa en resolver problemas continuos que buscan encontrar soluciones enteras mejores a la mejor solución factible que se haya localizado hasta el momento, explorando todo el dominio del problema original, si es necesario. Esto permite ahorrar un apreciado esfuerzo computacional.

El método Branch and Bound para el caso de problemas de optimización convexos (entre ellos el caso lineal y el de Control Predictivo Híbrido con funcional cuadrático y restricciones lineales), se puede resumir como:

1. Resuelva el problema de optimización relajado con todas las variables continuas, es decir, sin considerar que las variables “y” son binarias (enteras) (Problema P0). Si el resultado es que todas las variables y tienen valores binarios, ésta es la solución óptima. En caso contrario se crean dos subproblemas. Se escoge la variable continua más fraccionaria ( $y_{i1}$ ) y se construye el subproblema 1 (P1) agregando la restricción  $y_{i1} \leq 1$  y el subproblema 2 (P2) agregando la restricción  $y_{i1} \leq 0$ .
2. Seleccione el siguiente subproblema a resolver. Se resuelve el problema de optimización con restricciones. Si la función objetivo evaluada en el óptimo es mayor que el función objetivo evaluada en la mejor solución entera encontrada, entonces se obtiene como salida del problema la mejor solución factible encontrada y se regresa al problema anterior. Al hacer esto se eliminan las posibles ramas que se generarían a partir de este problema. Si no se cumple lo anterior se escoge la variable más fraccionaria ( $y_{i2}$ ) y se crean dos nuevos subproblemas agregando las restricciones  $y_{i2} \leq 1$  para el primer subproblema y agregando la restricción  $y_{i2} \leq 0$  para el segundo subproblema.
3. Si hay subproblemas por resolver, se regresa al paso 2. Si no hay subproblemas por resolver, entonces la solución óptima es la mejor entera que se haya encontrado hasta el momento. Si no hay soluciones el problema es infactible.

En la Figura 2 se muestra un diagrama del árbol completo de decisión del método de Branch and Bound. Es importante señalar que salvo en el peor caso, no todos los nodos serán recorridos en la medida que se vayan eliminado ramas infactibles o subóptimas.

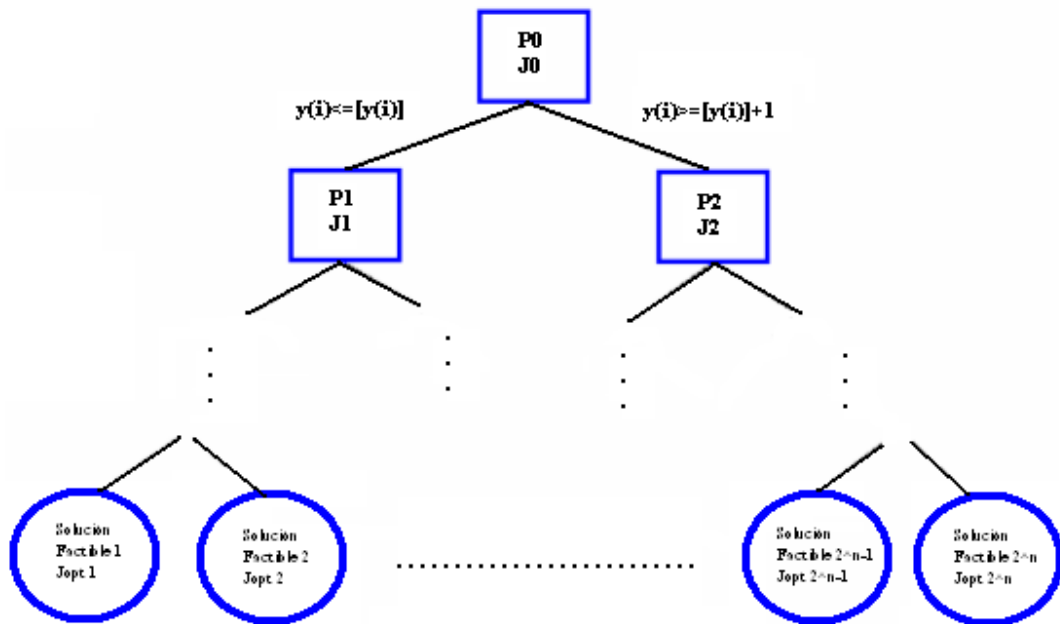


FIGURA 2. ESQUEMA ÁRBOL COMPLETO PARA ALGORITMOS BRANCH AND BOUND.

### 2.5.2 Optimización Entera Mixta No Lineal

La formulación del MINLP con variables binarias es:

$$\begin{aligned}
 &\min f(x, y) \\
 &s.t. \quad g(x, y) \leq 0 \\
 &\quad x \in X \subseteq R^n \\
 &\quad y \in \{0, 1\}^q
 \end{aligned} \tag{16}$$

donde  $x$  es un vector de  $n$  variables continuas,  $y$  es un vector de  $q$  variables binarias,  $f(x,y)$  es la función no lineal que se desea optimizar y  $g(x,y)$  un vector con las restricciones no lineales.

Métodos de enumeración explícita se hacen ineficientes si es que el dominio de las variables enteras no se acota.

En orden cronológico de desarrollo, se encuentran los siguientes métodos (Floudos, 1995):

- (i) Descomposición de Benders Generalizada (GBD), donde se crean dos secuencias de actualización de las cotas superior (no creciente) e inferior (no decreciente) que convergen en un número finito de iteraciones. La cota superior corresponde a la solución de subproblemas en las variables continuas arreglando las enteras mientras que las cotas inferiores se encuentran en métodos basados en la teoría dual.
- (ii) Branch and Bound (BB), donde comienza con la relajación del problema (soluciones que se buscan continuas) y presenta a continuación una enumeración en donde se arreglan una variable entera en cada nodo. Dependiendo del tipo de función objetivo, se puede determinar en un tiempo inferior a la enumeración de todas las soluciones factibles, la solución del problema de optimización.
- (iii) Aproximación exterior (OA), enfocado a problemas con restricciones de desigualdad no lineales, crea secuencias de cotas inferiores y superiores como GBD, pero tiene la característica propia de usar la solución del problema de encontrar la cota superior y linealizar la función objetivo y las restricciones alrededor del punto. Las cotas inferiores en OA se basan en la acumulación de funciones objetivos linealizadas y restricciones, alrededor de los puntos encontrados por la solución primal.
- (iv) Aproximación factible (FA), redondea el problema relajado a una solución entera, y ocupando ciertos criterios logra encontrar óptimos locales.
- (v) Aproximación exterior con relajaciones de igualdad (OA/ER), extiende OA para manejar restricciones de igualdad no lineales, relajándolas en desigualdades según el signo de sus multiplicadores.

- (vi) Aproximación exterior con relajaciones de igualdad y penalización aumentada (OA/ER/AP), introduce una función de penalización en el subproblema de la cota inferior de OA/ER.
- (vii) Aproximación exterior generalizada (GOA), extiende OA a problemas no lineales.
- (viii) Descomposición cruzada generalizada (GCD), usa información primal y dual explotando ventajas de GBD.

Cabe mencionar que estos algoritmos son clasificados según:

- (i) Algoritmos basados en descomposición (GBD y GCD)
- (ii) Métodos basados en linealizaciones (OA, OA/ER, OA/ER/AP, GOA)

Ambos tipos de algoritmos resuelven MINLP por procesos iterativos. El problema es descompuesto en un subproblema NLP (problema no lineal) en el cual se mantienen valores de las variables enteras fijas y un problema maestro MILP. El subproblema NLP optimiza las variables continuas y proveen de una cota superior a la solución MINLP, mientras que el problema maestro MILP tiene el rol de predecir una nueva cota inferior para el problema MINLP, así como también una nueva variable entera para cada iteración. La búsqueda termina cuando la cota inferior predicha es igual o excede a la actual cota superior. El esquema se muestra en la siguiente Figura 3.

La principal diferencia entre GBD y OA es en la definición del problema MILP maestro. En el algoritmo GBD, el problema MILP maestro está dado por una representación dual de un espacio continuo, mientras que en el método OA, está dado por una aproximación primal. En general, el método OA requiere menos iteraciones y entonces la solución de menos subproblemas NLP, pero los problemas MILP requieren más tiempo computacional en comparación con GBD. (Grossmann 1990).

Para tener suficientes condiciones de convergencia, los métodos de solución requieren que el problema MINLP satisfaga algunas condiciones de forma convexa.

En términos de performance algorítmica y computacional con MINLP, problemas muy largos (cientos de variables binarias, miles de variables continuas y cientos de restricciones) han sido resueltos (Sahinidis y Grossman, 1989). Ninguno de los métodos ha mostrado ser constantemente superior en todas las aplicaciones.

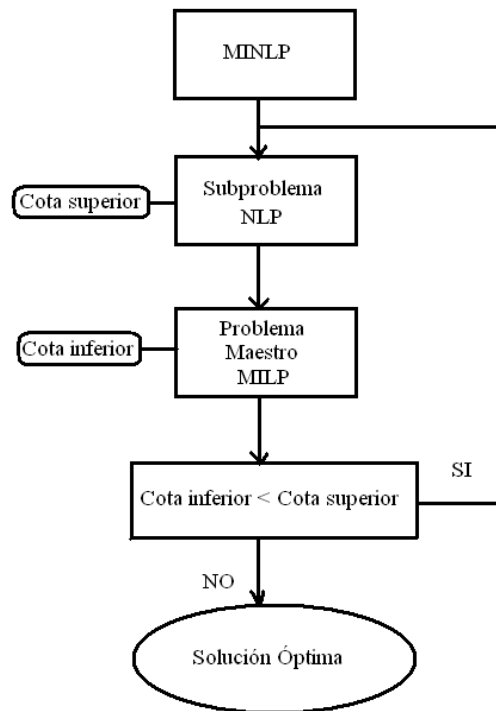


FIGURA 3. ESQUEMA SOLUCIÓN PROBLEMA MINLP.

### 2.5.3 Optimización Entera Mixta basada en Algoritmos Genéticos

Las heurísticas son procedimientos que buscan una buena solución al problema de Optimización Entera Mixta No Lineal en un tiempo computacional reducido y cercano al óptimo global. Es muy improbable que una heurística pueda replicar la solución exacta del problema de optimización; no obstante, una buena heurística puede lograr soluciones muy cercanas al óptimo buscado.

Para reducir el tiempo computacional estos métodos evitan cálculos de gradientes y realizan una búsqueda de soluciones basada en ciertas reglas.

Entre estas heurísticas se pueden mencionar Redes Neuronales y Algoritmos Evolutivos los cuales incluyen Algoritmos Genéticos, Programación Evolucionaria, Estrategias Evolutivas y Programación Genética.

Para poder aplicar estas heurísticas se requiere en primer lugar tener una codificación de las soluciones factibles. Esta codificación le asignará a cada variable de optimización entera un gen el cual puede tomar los valores dentro de un cierto rango en el cual le es permitido variar. Las variables que no sean enteras serán consideradas como variables libres para efecto de la selección de individuos, pero al momento de evaluar el *"fitness"* (o función objetivo) asociado a un cierto individuo se deberá en primer lugar resolver el problema de optimización no lineal sujeto a restricciones de individuo fijo y luego con la solución del problema evaluar la función objetivo global. Dado que se cuenta con software adecuado para resolver problemas de optimización no lineales continuos, la selección de individuos evitará tener que resolver problemas no lineales enteros mixtos. El esquema de solución es similar entonces al de la Figura 3, con la salvedad que el problema maestro será reemplazado por un algoritmo evolutivo que seleccione las mejores soluciones encontradas y produzca nuevas posibles generaciones y se tendrán la cantidad de individuos propuesta de problemas NLP en paralelo. Ver Figura 4.

Este esquema presenta la ventaja de que no requiere del problema maestro MILP, y que no trabaja con iteraciones secuenciales como lo hace el esquema de la Figura 3, por lo que se puede formular con búsqueda en paralelo de soluciones (para problemas muy complejos se puede ocupar un cluster de computadores, donde cada uno resuelva el problema para un individuo de la generación).

Estos algoritmos varían en la forma en que se producen las nuevas generaciones y a continuación dos métodos para hacerlo: la primera forma es la convencional en donde se aplican operadores de "mutación" y "crossover" para producir nuevas generaciones, y la segunda es ocupando el algoritmo PSO en donde se modifica la distancia que cada partícula (individuo) se mueve en cada dimensión por iteración.



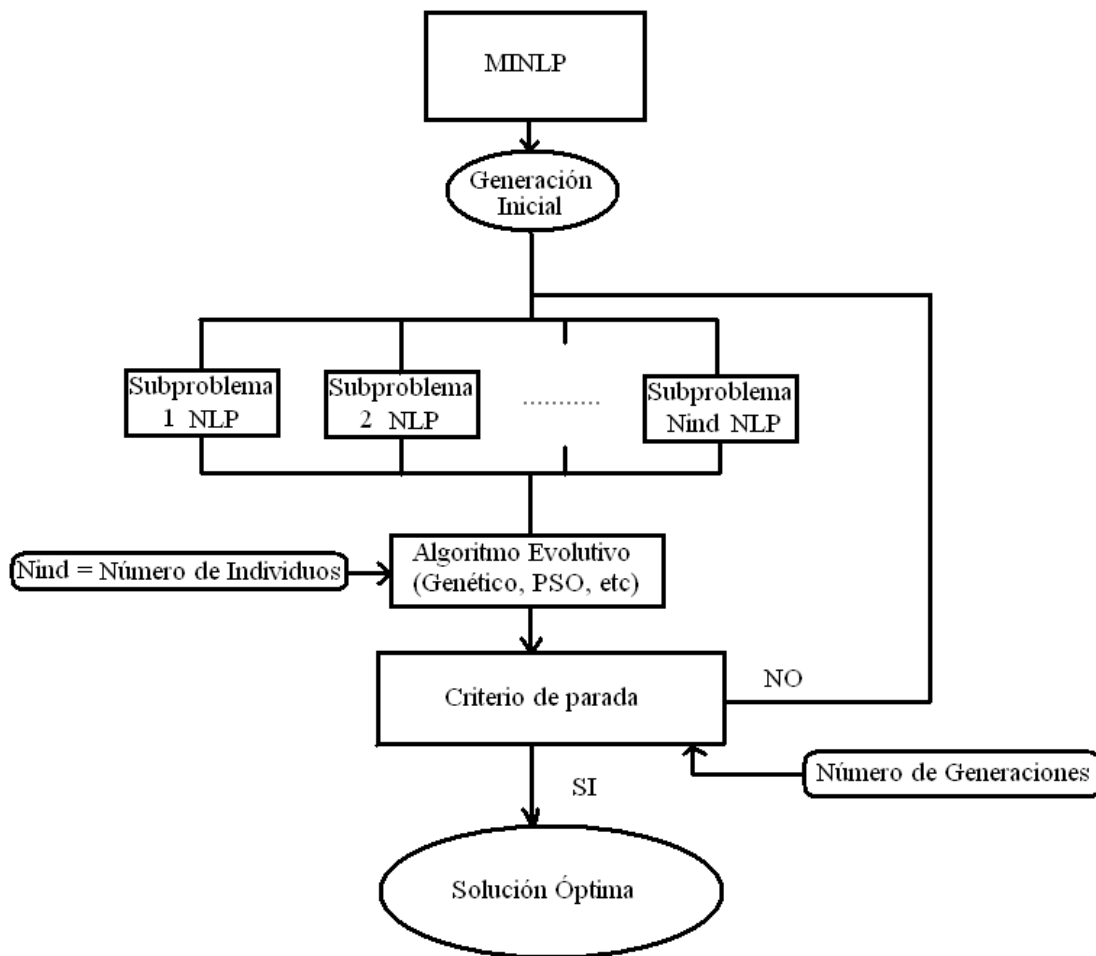


FIGURA 4. ESQUEMA SOLUCIÓN PROBLEMA MINLP USANDO ALGORITMOS EVOLUTIVOS.

### Algoritmos Genéticos convencionales

El problema de optimización se resolverá utilizando Algoritmos Genéticos (GA). Este método es muy flexible sobre todo en problemas de Optimización Entera-Mixta debido a que la codificación binaria de las variables enteras no es necesaria. Los genes de los individuos (soluciones posibles) están dados directamente por las variables enteras de optimización que en este caso son las distintas paradas que se deben recorrer y el vehículo que cumplirá el requerimiento. El cálculo de gradiente no es necesario lo que permitirá ahorrar un significativo tiempo computacional. El algoritmo genético simple que se ocupará se puede resumir como (Man *et al.*, 1998):

1. Inicializar una población aleatoria de individuos, que corresponden a posibles soluciones.
2. Evaluar la función objetivo para cada individuo de la población actual.
3. Seleccionar padres aleatorios.
4. Aplicar operadores genéticos como entrecruzamiento (*crossover*) y/o mutación a los padres, para una nueva generación.
5. Evaluar la función objetivo de todos los individuos de la nueva generación
6. Escoger los mejores individuos según los mejores valores de la función objetivo.
7. Reemplazar los individuos más débiles de la generación anterior por los mejores de la nueva generación generada en el punto 6.

Si el valor de la función objetivo alcanza una cierta tolerancia o el número máximo de generaciones se ha alcanzado, entonces se debe detener el algoritmo. En otro caso, se regresa al punto 2. En la Figura 5 se muestra un diagrama de flujo del algoritmo GA simple, recién explicado en sus siete pasos principales.

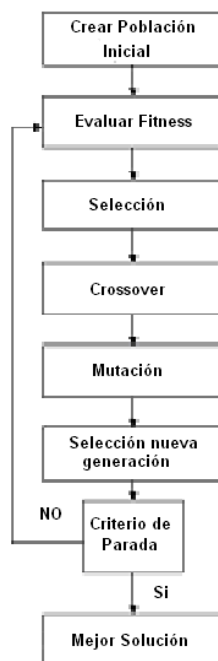


FIGURA 5. DIAGRAMA DEL ALGORITMO GENÉTICO SIMPLE.

### 3.- ESQUEMAS DE SOLUCIÓN PARA CONTROL PREDICTIVO HÍBRIDO

#### 3.1 HPC basado en modelos lineales, función objetivo cuadrática y variable manipulada entera. Aplicación: Control de Temperatura de un estanque

##### 3.1.1 Descripción del proceso

El esquema de solución del problema HPC basado en modelos lineales, función cuadrática objetivo y variable manipulada entera se ejemplifica con la aplicación de Control de Temperatura de un estanque de nivel. (Sáez *et al.* 2004)

En la Figura 6 se muestra un sistema de control de temperatura de un estanque localizado en el laboratorio de Electrotecnologías de la Universidad de Chile. El sistema consiste en un estanque cuadrado con tres calefactores on/off los cuales entregan una potencia máxima de 2000[W] cada uno.

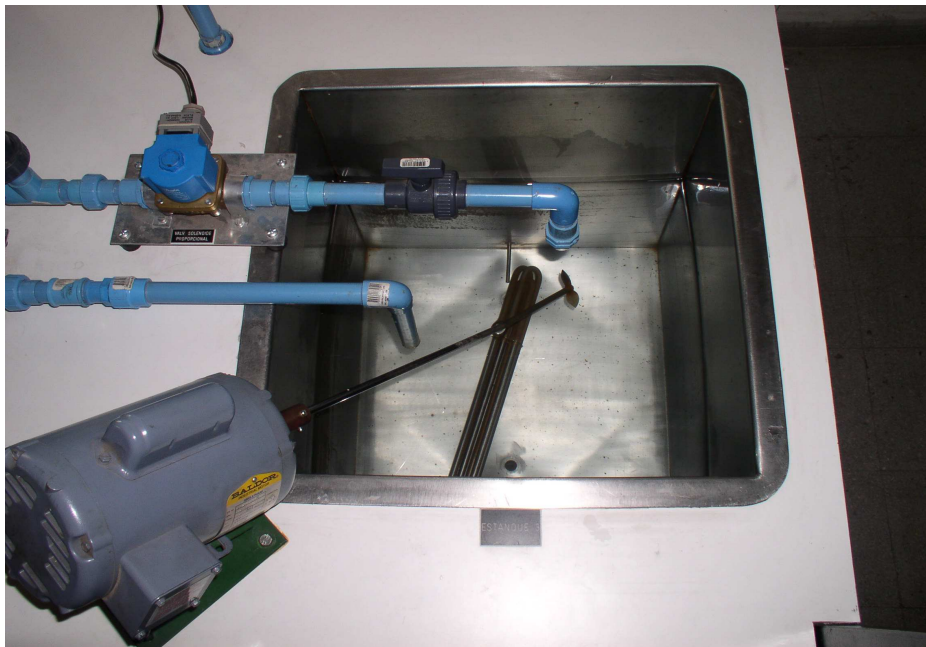


FIGURA 6. ESTANQUE DE AGUA CON CALEFACTORES ON/OFF. LABORATORIO DE ELECTROTECNOLOGÍAS UNIVERSIDAD DE CHILE.

En este caso, el objetivo de control es que la temperatura del agua siga una referencia mediante el prendido y el apagado de los calefactores on/off.

La variable controlada es la temperatura del agua y la variable manipulada está dada por el estado de los calefactores. El sistema es entonces híbrido SISO con variable manipulada entera.

Las entradas posibles o potencias posibles para entregar con los tres calefactores son:  $u(t) = 0, 2000, 4000, 6000 [W]$ , los cuales corresponden a la potencia agregada de los calefactores prendidos. Por otro lado, la temperatura del estanque es medida en el fondo del estanque con un termocupla localizada cerca de los calefactores. Además, un mezclador es usado para mantener la temperatura del estanque homogénea.

Experimentalmente, bajo recirculación de agua y aplicando un escalón de potencia mediante un calefactor, la dinámica del proceso puede ser aproximada mediante el siguiente modelo de primer orden con retardo:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{K}{\tau s + 1} e^{-\tau_d \cdot s} \quad (17)$$

donde  $K = 0.0067 [^{\circ}C/W]$  es la ganancia estática,  $\tau = 598.75 [s]$  es la constante de tiempo y  $\tau_d = 2 [s]$  es el retardo.

El modelo ARIX considerado en la ecuación (18) es obtenido mediante la discretización de la ecuación (17) usando un periodo de muestreo de 30 segundos.

$$(1 - 0.95q^{-1})y(t) = (30.61 \cdot 10^{-5} + 2.13 \cdot 10^{-5} q^{-1})u(t-1) + \frac{e(t)}{\Delta} \quad (18)$$

donde  $y(t)$  es la temperatura del agua en el instante  $t$ ,  $u(t)$  es la potencia entregada por los calefactores que toma valores discretos,  $q^{-1}$  es un operador de retardo ( $q^{-1}x(t) = x(t-1)$ ),  $\Delta = 1 - q^{-1}$ ,  $e(t)$  es un ruido blanco.

La solución convencional predictiva podría considerar el problema como un continuo y entonces la solución continua para la variable manipulada podría ser discretizada en los cuatro niveles usando PWM (Pulse Wise Modeling), obteniendo una solución sub-óptima. El control predictivo híbrido (HPC) propuesto permite incluir la característica discreta de la variable manipulada en el problema de optimización mediante la obtención del óptimo de la solución óptima global.

### **3.1.2 Base de evaluación**

Para las distintas estrategias HPC, los horizontes  $N_1$ ,  $N_y$ ,  $N_u$  de la función objetivo (9) son definidos basados en la constante de tiempo del proceso de temperatura estable. En este caso, los parámetros fueron elegidos son  $N_1 = 1$  y  $N=N_y=N_u= 5, 7$  y  $10$ . El factor de peso es sintonizado mediante un análisis de sensibilidad y se toma igual a  $\lambda = 0.1$ . Para el análisis comparativo de las estrategias HPC, la media del funcional objetivo es evaluado.

Además, el tiempo total requerido por cada controlador es medido. Se utilizó para realizar las pruebas un Pentium® 4, CPU de 2.80 GHz, con un 2,79Ghz, 1.00 GB de RAM.

Para la estrategia de control predictiva híbrida basada en algoritmos genéticos (HPC&GA), la probabilidad de mutación es 0.001, la de *crossover* es 0.7 y el criterio de parada corresponde al alcance de un máximo número de generaciones. Estos parámetros se dejan fijos y con valores similares a los típicos en la literatura, sin embargo es posible realizar un análisis de sensibilidad y sintonizarlos de mejor manera. En esta tesis sin embargo, los parámetros escogidos para sintonizar serán número de generaciones y de número de individuos.

### **3.1.3 Resultados por simulación**

Las pruebas por simulación usando Algoritmo Genético Simple son realizadas 20 veces bajo las mismas condiciones en cuanto a número de individuos y de generaciones. Los

siguientes resultados que se presentan corresponden a los valores medios de esas 20 pruebas.

En la Figura 7 la evolución de la función objetivo (9) como función del número de generaciones para diferentes números de individuos es mostrada. La Figura 8 presenta un detalle de la Figura 7 entre 20 y 30 número de generaciones. A partir de esas figuras, el número óptimo de generación se fija en 25 con 20 individuos.

La Figura 9 muestra el tiempo computacional en función del número de generaciones y de individuos. Los tiempos computacionales resultantes son lineales con respecto al número de generaciones y sus pendientes aumentan a medida que incrementamos el número de individuos. De la Figura 9 se extrae que para todos los casos testeados, el tiempo computacional es razonablemente pequeño. Para el número óptimo de generaciones e individuos previamente escogidos, el tiempo computacional aproximado es de 8 segundos.

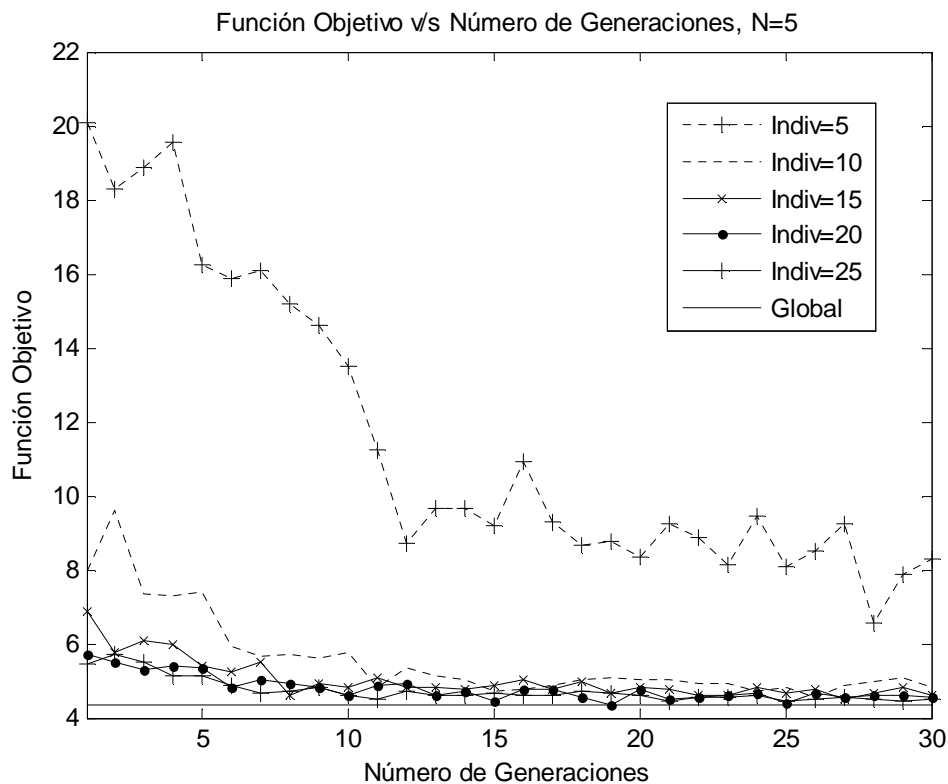


FIGURA 7. EVOLUCIÓN DE LA FUNCIÓN OBJETIVO.

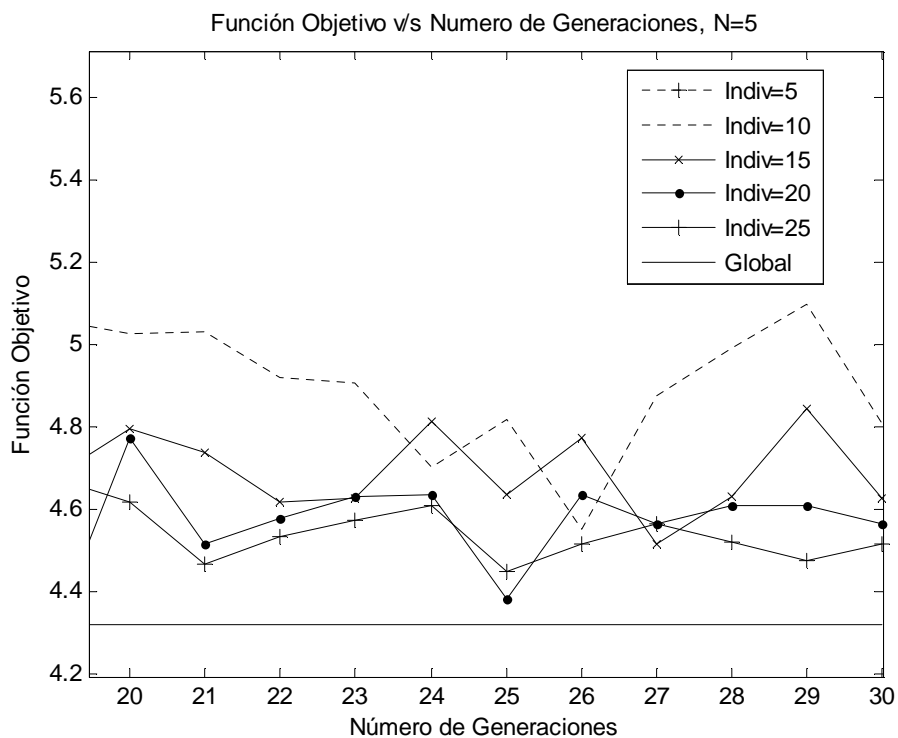


FIGURA 8. DETALLE DE LA EVOLUCIÓN DE LA FUNCIÓN OBJETIVO.

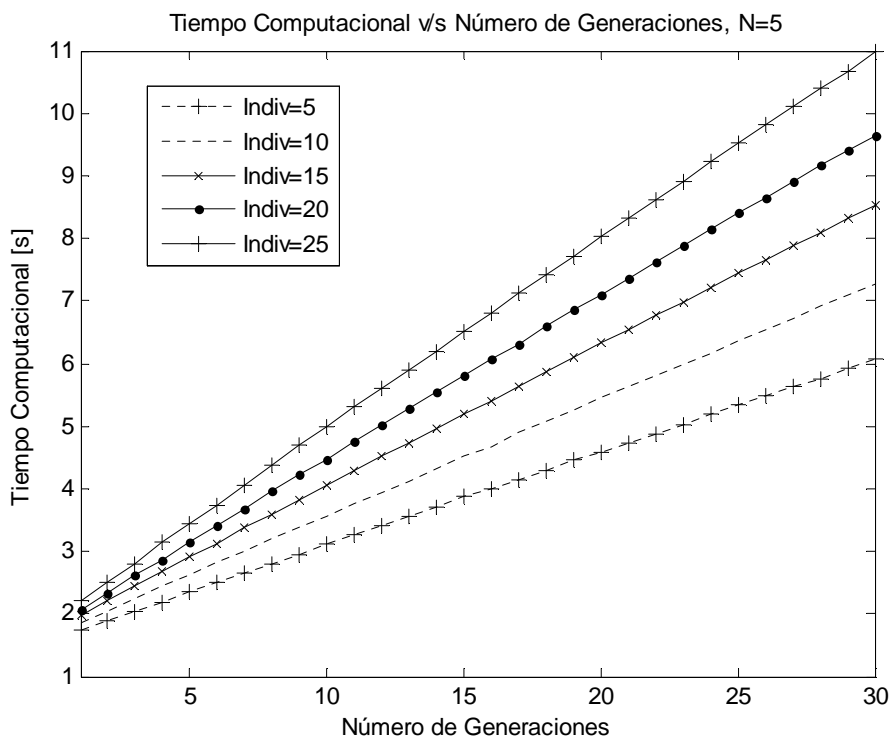


FIGURA 9. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL.

La Figura 10 muestra la respuesta del sistema dada por la medición de temperatura durante 4000[s] de simulación, usando las diferentes estrategias HPC. La Figura 11 tiene el detalle de la Figura 10 desde lo 900 hasta los 1800[s] de simulación. La Figura 12 muestra la potencia total entregada por los calefactores on/off. En este caso, HPC&GA considera 20 individuos y 25 generaciones.

Las respuestas del sistema son idénticas cuando se ocupa Enumeración Explícita (HPC&EE) y Branch and Bound (HPC&BB). Debido a que HPC&GA ocupa una heurística para la búsqueda de soluciones, algunas diferencias se pueden apreciar en la variable manipulada mostrada. Sin embargo, pese a que no alcanza el óptimo global, la forma en que se programan los Algoritmos Genéticos ayuda a encontrar una solución muy cercana al óptimo global.

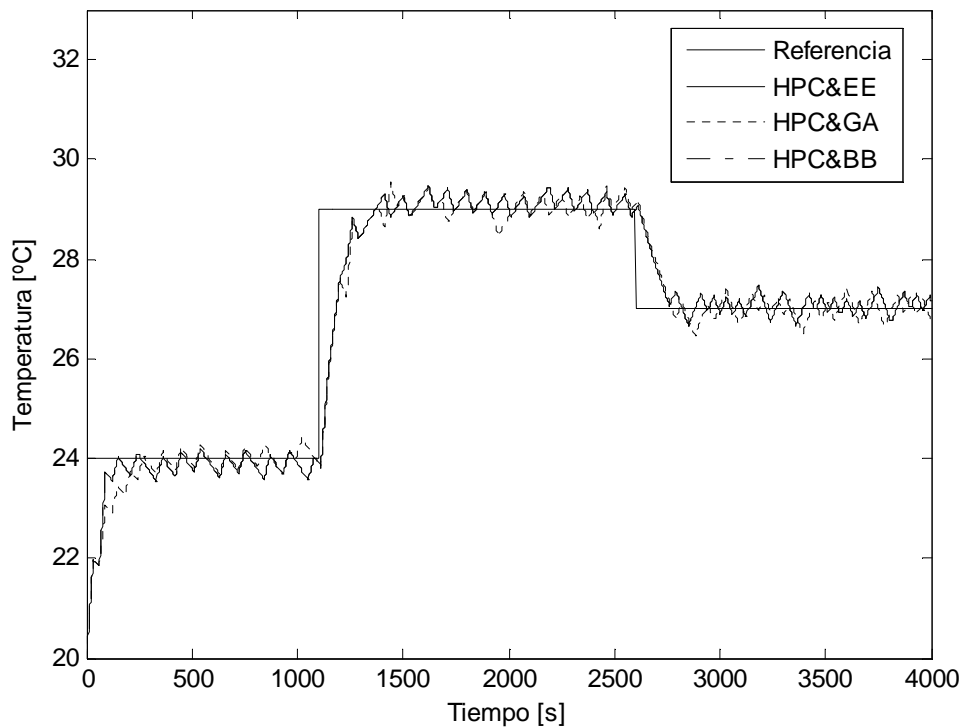


FIGURA 10. PRUEBAS POR SIMULACIÓN. RESPUESTA DE LA TEMPERATURA EN EL ESTANQUE.



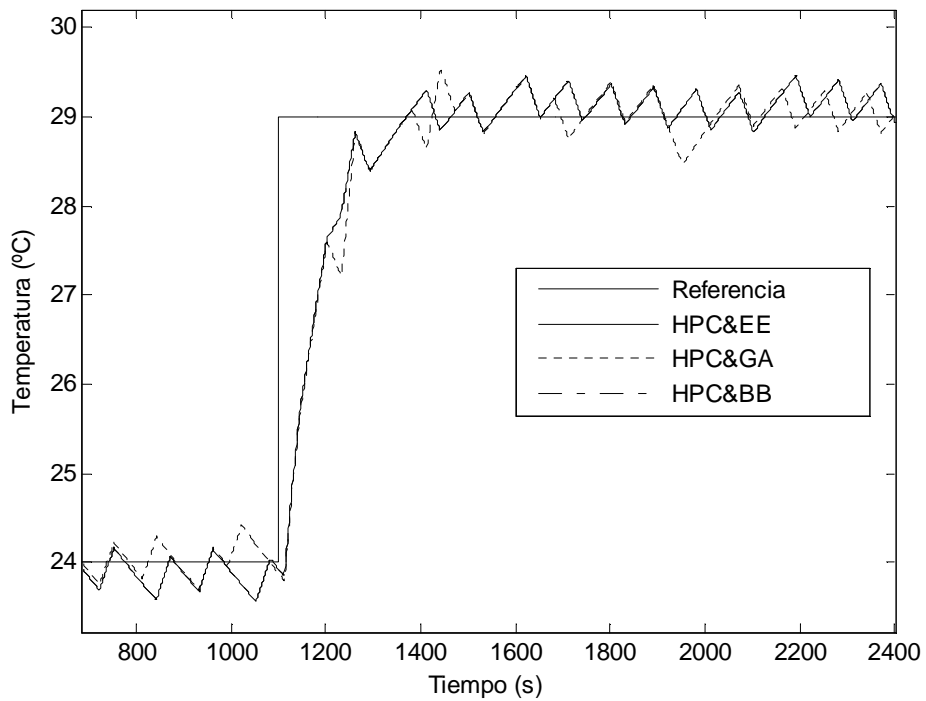


FIGURA 11. PRUEBAS POR SIMULACIÓN. DETALLE DE LA RESPUESTA DE LA TEMPERATURA EN EL ESTANQUE.

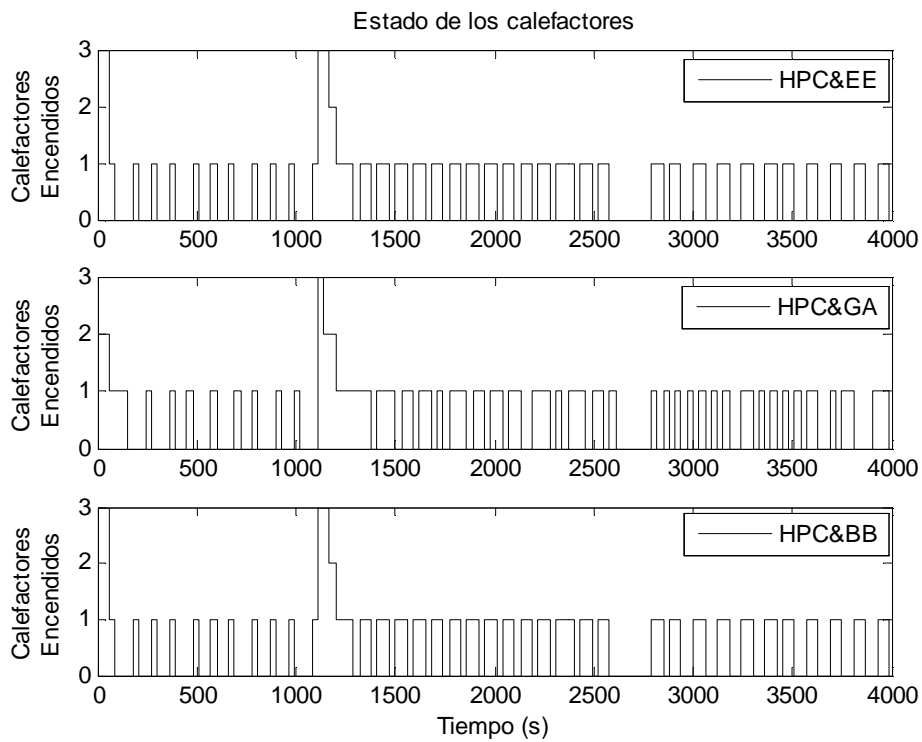


FIGURA 12. PRUEBAS POR SIMULACIÓN. POTENCIA TOTAL ENTREGADA POR LOS CALEFACTORES.

La Tabla 1 muestra los valores medios de la función objetivo definido en (9), usando distintos horizontes de predicción y de control. El tiempo computacional requerido para las mismas pruebas se muestra en la Tabla 2.

A partir de la Tabla 1 y debido a que el problema es convexo, el valor medio de la función objetivo es el mismo para HPC&BB y HPC&EE cuando  $N=5$  y  $N=7$ . Para  $N=10$  el problema de optimización usando HPC&EE no puede ser resuelto en un tiempo razonable. La estrategia HPC&GA encuentra una solución cercana a la optima global debido a que el valor medio de la función objetivo es un poco mayor.

Como se muestra en la Tabla 2, el tiempo computacional usando HPC&EE crece exponencialmente cuando el horizonte de predicción y control aumentan. Además, el tiempo computacional usando HPC&BB aumenta 10 veces cuando el horizonte de control y predicción aumentan al doble. Sin embargo, cuando se ocupa HPC&GA, el tiempo computacional se mantiene acotado y reducido sin importar el horizonte de predicción. Esto se debe a la forma en que se debe a que el criterio de parada se tomó como el número máximo de generaciones.

TABLA 1. VALORES MEDIOS DE LA FUNCIÓN OBJETIVO

	(N2=Nu=5, $\lambda=0.1$ )			(N2=Nu=7, $\lambda=0.1$ )			(N2=Nu=10, $\lambda=0.1$ )		
	J <sub>1</sub>	J <sub>2</sub>	J	J <sub>1</sub>	J <sub>2</sub>	J	J <sub>1</sub>	J <sub>2</sub>	J
HPC&BB	3.8500	4.8955	4.3396	10.814	4.6119	11.276	20.372	6.5448	21.027
HPC&EE	3.8500	4.8955	4.3396	10.814	4.6119	11.276	***	***	***
HPC&GA	4.0855	5.8657	4.6721	12.955	8.8806	13.843	21.518	14.814	22.999

TABLA 2. TIEMPO COMPUTACIONAL.

Estrategia de Control	Tiempo [s] (N=5, $\lambda=0.1$ )	Tiempo [s] (N=7, $\lambda=0.1$ )	Tiempo [s] (N=10, $\lambda=0.1$ )
HPC&BB	19,703	57,328	211,027
HPC&EE	10,339	167,051	***
HPC&GA (20,25)	8,412	17,891	21,689

### 3.1.4 Resultados experimentales

De manera similar a los resultados por simulación, los parámetros de la función objetivo para el problema experimental fueron:  $N_x=1$ ,  $N_y=5$ ,  $N_u=5$  y  $\lambda=0.1$ . Para la estrategia HPC&GA se consideraron 20 individuos y 25 generaciones. Es importante mencionar que debido a que el agua de recirculación se tomó directamente desde la cañería, la perturbación dada por su temperatura no fue controlable por lo que cada realización del experimento sufrió de distintos niveles de perturbación no medible ni controlable. Los resultados que se muestran a continuación corresponden al promedio de 10 realizaciones que se hicieron experimentalmente. Las figuras 13 y 14 muestran los resultados experimentales para las estrategias de HPC&BB y HPC&GA para el control de temperatura del estanque.

De la Figura 13 se extrae que la estrategia HPC&GA tiene una respuesta muy similar a la de HPC&BB, la cual entrega el óptimo global a cada instante. La respuesta HPC&GA tiene menor sobreoscilación que la otra estrategia, sin embargo, genera una acción de control que tiene mayores oscilaciones (y por ende mayor  $J_2$ ).

La Tabla 3 muestra el valor medio de la función objetivo definida en (9).

De los gráficos y la tabla se determina que ambas estrategias logran seguir la referencia de buena manera, pero HPC&GA usa más energía de control al realizar más variaciones en la variable manipulada.

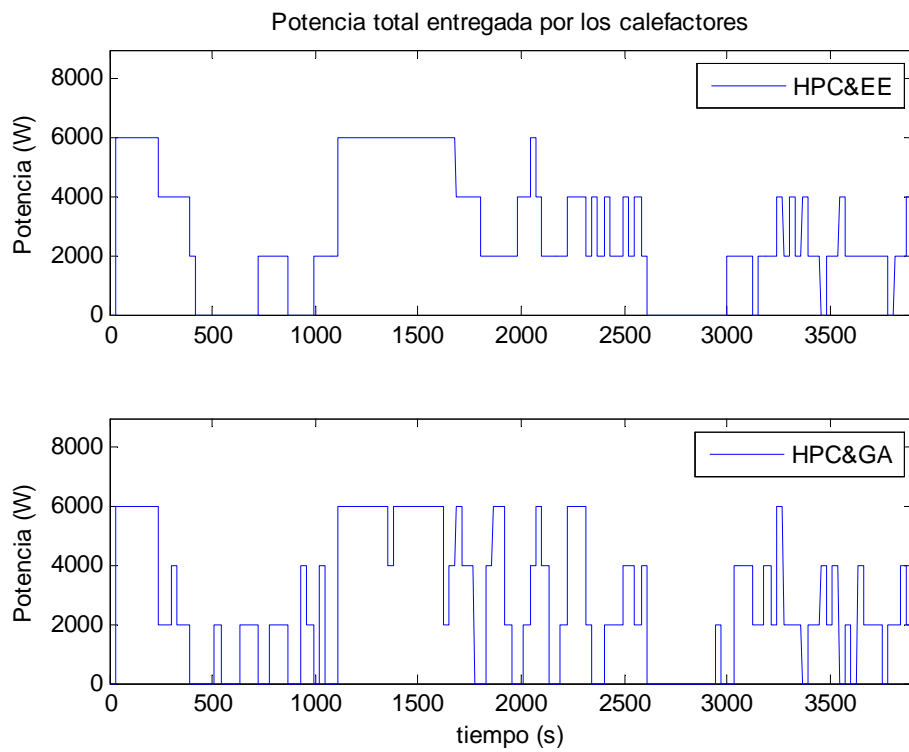


FIGURA 13. RESULTADOS EXPERIMENTALES CON DISTINTAS ESTRATEGIAS HPC: VARIABLE MANIPULADA.

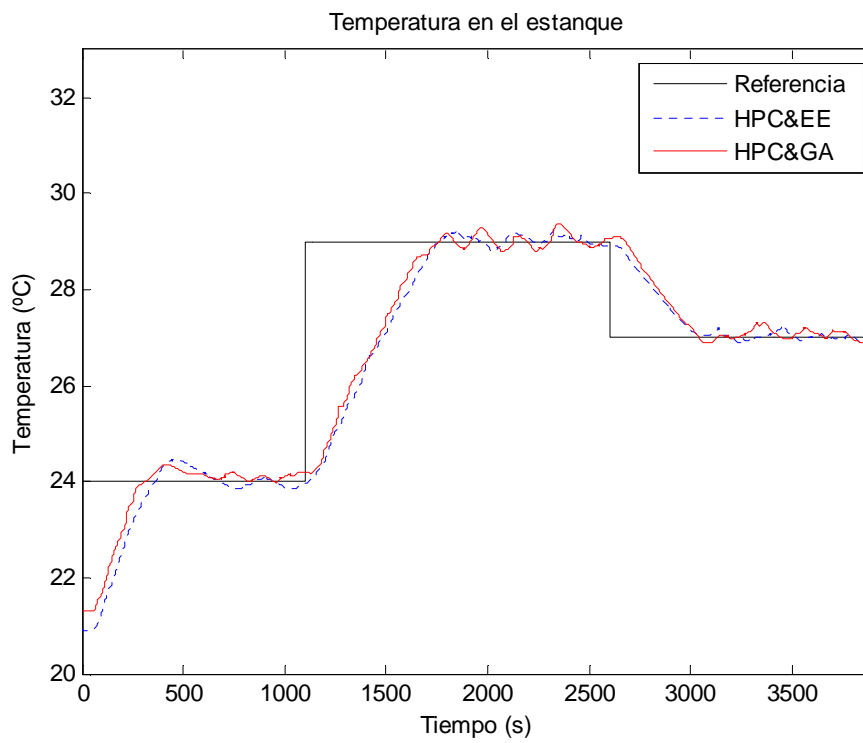


FIGURA 14. RESULTADOS EXPERIMENTALES CON DISTINTAS ESTRATEGIAS HPC: VARIABLE CONTROLADA.

TABLA 3. VALORES MEDIOS FUNCIÓN OBJETIVO (9), RESULTADO EXPERIMENTAL

	$(N_2=Nu=5, \lambda=0.1)$		
	$J_1$	$J_2$	$J$
GA 20,25	11.524	14.869	13.01
BB	9.531	8.098	10.34

### 3.1.5 Síntesis y conclusiones

Se presentó un método nuevo de solución del problema de control predictivo híbrido basado en algoritmos genéticos.

La principal ventaja del control predictivo está relacionada con la inclusión de las variables enteras dentro de la formulación del problema de optimización lo que representa de mejor forma al proceso real. Entonces, se evitan las simplificaciones y la solución óptima global en cada instante se calcula y se aplica al proceso.

Algunas de las alternativas de solución del problema de control predictivo híbrido son enumeración explícita (HPC&EE) y Branch and Bound (HPC&BB) entre otras. HPC&EE requiere de un alto esfuerzo computacional y por ejemplo, no fue capaz de resolver el problema de control de temperatura real en un tiempo razonable por lo que no se pudo implementar experimentalmente. En HPC&EE el tiempo computacional aumenta exponencialmente con el número de predicciones. HPC&BB es una mejor alternativa en comparación a HPC&EE pero si el horizonte de control y de predicción aumentan mucho, el tiempo computacional se verá incrementado también. Por otro lado, HPC&GA ha mostrado ser eficiente y permite alcanzar soluciones cercanas a la óptima. Además, muestran un buen desempeño en cuanto a tiempo computacional, y entonces permite encontrar buenas soluciones en tiempo reducido. Esta última característica es una consecuencia de evitar cálculo de gradiente.

Todo lo mencionado fue probado además experimentalmente, en donde los resultados mostraron ser similares con HPC&BB y HPC&GA en el control de temperatura del estanque.

### 3.2 HPC basado en modelos difusos, función objetivo cuadrática y variable manipulada y de estado entera. Aplicación: Control de nivel de un sistema de estanques híbridos.

#### 3.2.1 Descripción del proceso

El comportamiento del sistema de estanques mostrado en la Figura 15, está definido por las siguientes ecuaciones diferenciales no lineales y ecuaciones algebraicas que definen las regiones de cambios (*switching*) (Núñez *et al.* 2006).

$$\begin{aligned} \frac{dh_1}{dt} \cdot \pi \cdot \frac{R_1^2}{H_1^2} h_1^2 &= K_{CP} \cdot u + \phi_{ONOFF2} - \overbrace{V_1 h_1}^{\phi_{V1}} - \phi_{ONOFF1} \\ \frac{dh_2}{dt} \cdot \pi \cdot R_2^2 &= \overbrace{V_1 h_1}^{\phi_{V1}} + \phi_{ONOFF1} - \overbrace{V_2 h_2}^{\phi_{V2}} - \phi_{ONOFF2} \end{aligned} \quad (19)$$

Si  $(h_2 \geq H_{2min})$  y  $(h_1 < H_{1max})$  entonces  $\phi_{ONOFF2} = K_{ONOFF2}$

Si  $(h_1 \geq H_{1max})$  y  $(h_2 < H_{2max})$  entonces  $\phi_{ONOFF1} = K_{ONOFF1}$

donde  $h_1$  y  $h_2$  son el nivel de líquido en el estanque cónico y en el cilíndrico respectivamente y  $H_{1min}, H_{1max}, H_{2min}, H_{2max}$  definen los niveles de *switching*.

La variable controlada es este caso será el nivel del estanque cónico  $h_1$ , y la variable manipulada corresponderá al voltaje de la bomba de entrada ( $u$ ), la cual tiene niveles discretos que van desde 0 hasta 100, en pasos de 5. Se asume también que se miden ambos niveles  $h_1$  y  $h_2$ .

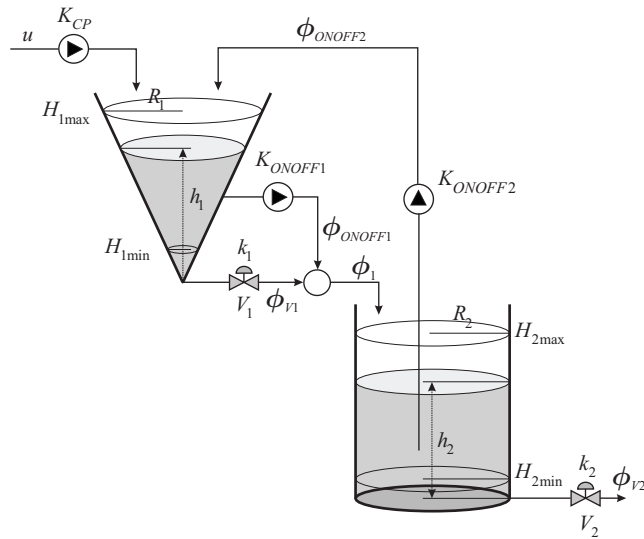


FIGURA 15. SISTEMA DE ESTANQUES HÍBRIDO.

### 3.2.2 Modelación Difusa para el sistema híbrido

El comportamiento del sistema híbrido será modelado considerando una estructura para el modelo difuso de la forma (13). Para realizar la identificación se toman datos de entrada y salida del proceso. Las señales ocupadas para la excitación de la bomba de entrada y la salida de la planta se muestran en las figuras 16 y 17. Las señales fueron muestreadas con un tiempo de muestreo  $T_s = 10$  s.

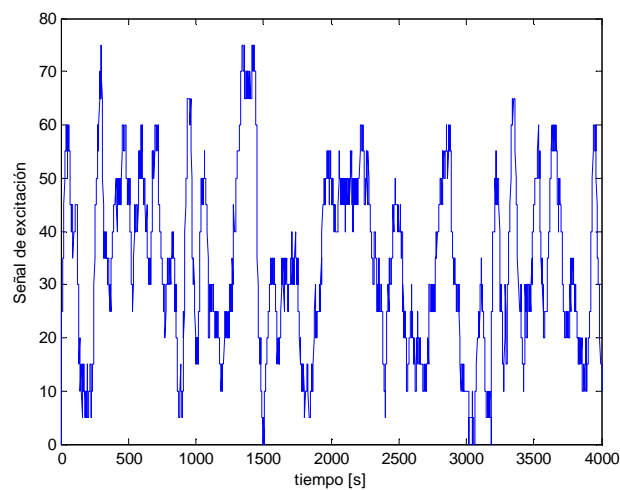


FIGURA 16. SEÑAL DE EXCITACIÓN DE ENTRADA.

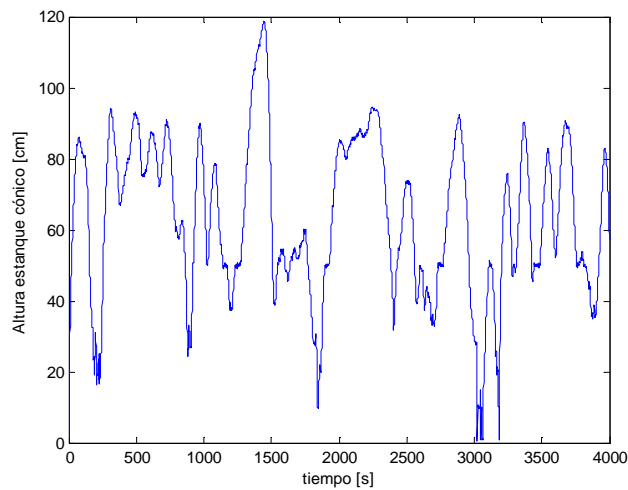


FIGURA 17. SEÑAL DE SALIDA  $H_t$ .

El diseño de la distribución de las funciones de pertenencia es un elemento clave en el procedimiento de modelación. En nuestro caso se obtiene mediante el análisis de los vectores propios de las matrices de covarianza de los clusters. Se obtienen los clusters ocupando las señales de entrada y salida obtenidas de las variables medidas. Los datos deben ser tomados tratando de obtener información respecto a todos los clusters posibles o comportamiento del sistema, mediante el recorrido de todos los puntos de operación posibles (distintas regiones de switching). El análisis de las componentes principales es realizado mediante el cálculo de los vectores propios. El radio entre los vectores propios es calculado para cada cluster.

El análisis de los vectores propios para todos los clusters se muestra en la Figura 18. Se muestra que en algunos clusters, los vectores principales cambian drásticamente. Este cambio tiene relación con los cambios en la dinámica del sistema e indican potenciales zonas de switching del sistema. La idea es poner dos funciones de pertenencia alrededor de cada cambio de dirección de los vectores propios (mínimo y máximo eigenvector ratio). Este se hace debido a que a la región de switching no puede ser exactamente definida (especialmente en el caso de registros ruidosa). Esta idea involucra una banda de tolerancia alrededor de la zona de switching. En la Figura 18 se muestran las correspondientes funciones de pertenencia propuestas.



La estructura del modelo difuso es definida como en la ecuación (14), que se recuerda a continuación:

$$x_{k+1} = \sum_{i=1}^{s-R_i} \mu_i(z_k) (a_i x_k + b_i u_k + r_i)$$

$$\mu_i(z_k) = \frac{\prod_{r=1}^p B_{i,r}(z_{k,r})}{\sum_{i=1}^{s-R_i} \prod_{r=1}^p B_{i,r}(z_{k,r})}$$

donde la premisa es  $z_k = h_{1,k}$  y el vector de consecuencias es igual a  $x_c = [h_{1,k} \ u_k \ 1]^T$ . Los parámetros del modelo difuso ( $\theta_i = [a_i \ b_i \ r_i]$ ), son obtenidos por estimación basada en mínimos cuadrados, y se muestran en la Tabla 4.

TABLA 4. PARÁMETROS MODELO DIFUSO

$i$	$a_i$	$b_i$	$r_i$
1	0.8376	0.3403	0.0386
2	0.9764	0.0522	0.0511
3	0.9873	0.0290	0.0305
4	0.9747	0.0196	0.7656
5	0.9933	0.0125	-0.0136
6	0.9946	0.0091	0.0265
7	0.9987	0.0066	-0.2163
8	1.0015	0.0045	-0.4334

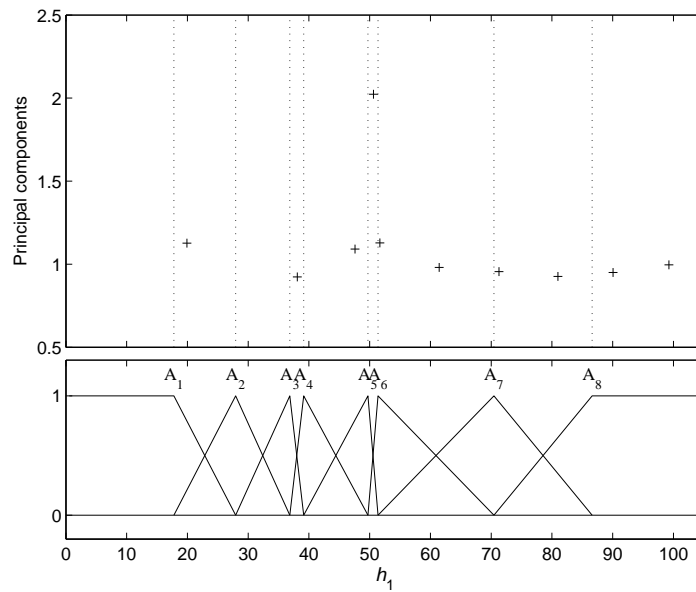


FIGURA 18. ANALISIS DE VECTORES PROPIOS Y FUNCIONES DE PERTENENCIA.

La validación del modelo difuso diseñado se muestra en la Figura 19. El modelo propuesto entrega una buena estimación de la salida del proceso, e inherentemente incorpora la característica híbrida del sistema (zona de switching).

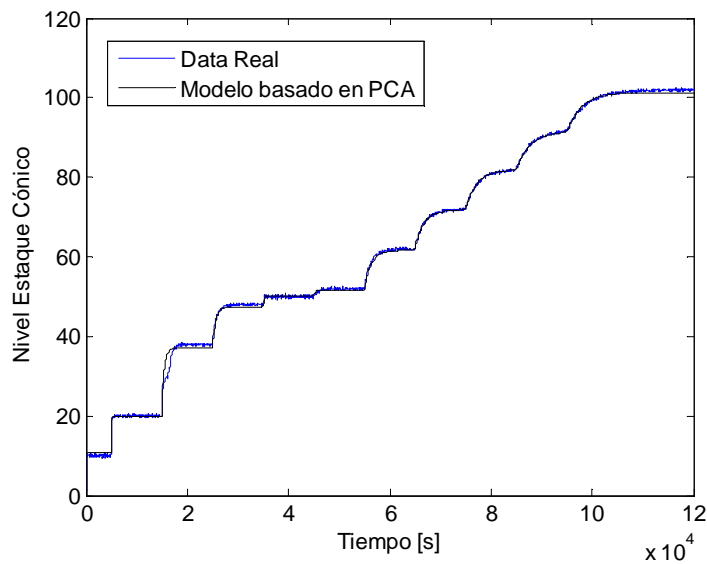


FIGURA 19. VALIDACIÓN DEL MODELO DIFUSO.

### 3.2.3 Algoritmo de solución

Se ocupan Algoritmos Genéticos para resolver el problema de optimización debido a que es entero no lineal tanto en la función objetivo como en las restricciones. Una de las ventajas que se tiene es que no será necesario estimar gradientes, lo cual relaja el esfuerzo computacional (Man et al, 1998).

Una solución potencial del algoritmo genético (secuencia de control) se llamará individuo. El individuo puede ser representado por un conjunto de parámetros relacionados a los genes de un cromosoma y se pueden describir en una forma binaria o entera. Un individuo representará una posible secuencia de acciones de control  $\{u(k), \dots, u(k + N_u - 1)\}$ , en donde cada elemento representa a un gen y el largo del individuo es igual al horizonte de control.

Usando el algoritmo genético convencional, los mejores individuos tienen mayor probabilidad de ser elegidos asegurando buena descendencia o nueva generación. Los mejores padres se seleccionan con alta probabilidad y se recombinan para producir descendencia. En la recombinación se aplican dos operadores genéticos fundamentales, el de crossover y el de mutación. Para el mecanismo de crossover se intercambian porciones de dos individuos con cierta probabilidad para la producción de dos nuevos individuos. El operador de mutación altera cualquier porción del individuo aleatoriamente con cierta probabilidad (Man et al, 1998).

Para efectos de claridad en cuanto a la derivación de la ley de control basada en el algoritmo genético simple (SGA), se describe el siguiente ejemplo. Asuma que la variable manipulada pertenece al rango  $[0,100]$  y está cuantizada por pasos de 5, y entonces hay 21 posibles entradas en cada instante. Entonces, el procedimiento para este ejemplo está dado por los siguientes pasos:

1. Inicializar una población aleatoria de individuos, es decir, crear soluciones o secuencias de acción de control al azar que sean factibles para el problema de control predictivo difuso híbrido. En este caso, debido a que el horizonte de control es 6, hay  $21^6$  individuos posibles. En el siguiente ejemplo, se muestra una población de tamaño 7 individuos por generación.

Población $i$ $\Leftrightarrow$	}	Individuo 1	}	$\Leftrightarrow$	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>45</td><td>50</td><td>60</td><td>50</td><td>80</td><td>50</td></tr></table>	45	50	60	50	80	50
		45			50	60	50	80	50		
		Individuo 2			<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>15</td><td>25</td><td>20</td><td>05</td><td>85</td><td>65</td></tr></table>	15	25	20	05	85	65
		15			25	20	05	85	65		
		Individuo 3			<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>45</td><td>50</td><td>60</td><td>50</td><td>80</td><td>50</td></tr></table>	45	50	60	50	80	50
		45			50	60	50	80	50		
		Individuo 4			<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>35</td><td>55</td><td>00</td><td>50</td><td>80</td><td>45</td></tr></table>	35	55	00	50	80	45
35	55	00	50	80	45						
Individuo 5	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>55</td><td>40</td><td>100</td><td>55</td><td>65</td><td>15</td></tr></table>	55	40	100	55	65	15				
55	40	100	55	65	15						
Individuo 6	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>80</td><td>50</td><td>60</td><td>75</td><td>30</td><td>05</td></tr></table>	80	50	60	75	30	05				
80	50	60	75	30	05						
Individuo 7	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>20</td><td>15</td><td>40</td><td>00</td><td>10</td><td>00</td></tr></table>	20	15	40	00	10	00				
20	15	40	00	10	00						

Por ejemplo, Individuo 1 significa que el vector de secuencias de acción de control futuras es:

$$\text{Individuo 1} = [u(k) \ u(k+1) \ u(k+2) \ u(k+3) \ u(k+4) \ u(k+5)]^T = [45 \ 50 \ 60 \ 50 \ 80 \ 50]^T$$

Note que si el individuo 1 resulta ser el óptimo, o mejor adaptado, la acción de control que se aplicará en el instante actual será:

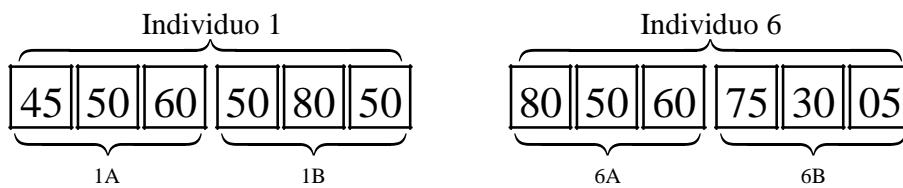
$$u^*(k) = 45$$

2. Evaluar la función objetivo de todos los individuos de la población usando el funcional (14). Note que la predicción  $\hat{y}(k+j)$  se calcula recursivamente usando las acciones de control futuras que son conocidas para un individuo dado. En general:

$$\hat{y}(k+j) = f(\hat{y}(k+j-1), \dots, u(k+j-1), \dots)$$

donde  $f$  es la función no lineal definida por el modelo difuso.

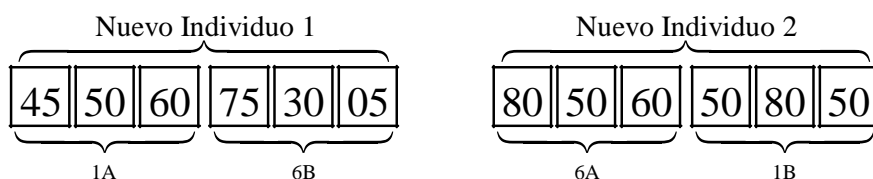
3. Seleccionar padres aleatorios de la población. Esta selección se realiza aleatoriamente y se recomienda usar selección estocástica universal (SUS) en la cual un padre que tiene mejor función objetivo tiene mayor probabilidad de ser seleccionado con lo cual se potencia la población manteniendo a los mejores. Por ejemplo, los individuos 1 y 6 son seleccionados como padres.



4. Aplicar los operadores genéticos de crossover y mutación a los padres seleccionados para producir una nueva población. Cada operador tiene una probabilidad de ocurrencia asociada. La probabilidad de crossover indica cual es la probabilidad de que se aplique el operador crossover a dos padres seleccionados. En el caso de que no se efectúe esta operación ambos padres pasan a la siguiente generación. En el caso de la probabilidad de mutación, esta indica cual es la probabilidad que tiene un bit o un gen de cada nuevo cromosoma en mutar aleatoriamente. Ambas probabilidades deben ser sintonizadas de acuerdo a las características de cada problema.

En cuanto al operador de crossover, no solo se debe fijar su probabilidad de ocurrencia, sino que también el tipo de crossover que se aplica. Este puede variar en cuanto a puntos de crossover y si es que estos puntos son fijos o aleatorios. En el ejemplo a continuación el punto de crossover se seleccionó y se mantuvo fijo entre el tercer y el cuarto gen del cromosoma o individuo.

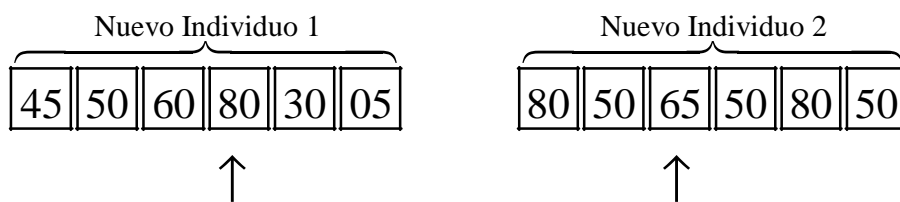
Luego de aplicar crossover a los padres seleccionados 1 y 6, los hijos resultantes de la mezcla resultan ser:



Notar la porción denotada 1A corresponde a la primera porción del individuo 1, 6B a la segunda porción del individuo 6, 6A la primera porción del individuo 6 y 1B la segunda del individuo 1.

En cuanto al operador de mutación, en el primer algoritmo genético planteado por Holland, su probabilidad de ocurrencia era sintonizada muy baja por lo que afectaba muy poco en la producción de nuevos individuos. Sin embargo, dependiendo del problema que se resuelva, esta probabilidad puede ser sintonizada y lograr mejores resultados en menor tiempo. Cada gen puede o no sufrir mutación, y una vez que esto queda definido, aquellos genes que serán mutados (o cambiados) tomarán un valor dentro los factibles según una distribución uniforme.

En el ejemplo mostrado a continuación, los nuevos individuos 1 y 2 sufren mutación en los genes número 4 y 3 respectivamente. Podría ocurrir también que no sufran de mutación o bien la mutación ocurra en múltiples genes. Una vez seleccionados los genes en los que se aplicará mutación, con una distribución uniforme se selecciona los nuevos valores. En el ejemplo el cuarto gen del nuevo individuo 1 es cambiado de 75 a 80, y del nuevo individuo 2 el tercer gen cambia de 60 a 65.



5. Se evalúa la función objetivo dada por la función objetivo (12) para todos los nuevos individuos generados.

6. Se ubican los nuevos individuos dentro de la población inicial y mediante SUS se seleccionan a los mejores individuos de acuerdo a un ranking basado en sus funciones objetivo. Hay varias posibilidades para realizar el ranking y se recomienda seleccionar aquella que permita mantener la diversidad en la población. Esto es crítico en problemas en donde la función objetivo de un buen individuo puede ser en cuanto a las unidades propias del problema muy superior a las de un individuo peor adaptado, en donde al realizar SUS la probabilidad de seleccionar al peor es tan baja que el mejor individuo pronto repleta la población, perdiendo la posibilidad de indagar nuevas soluciones y cayendo en posibles mínimos locales.

7. Basado en la selección por ranking, los individuos más débiles son reemplazados por los más fuertes y se mantiene el número de individuos por generación.

8. El algoritmo se detiene si la función objetivo del mejor individuo alcanza cierta tolerancia o bien el número máximo de generaciones es alcanzada. Esto corresponde al criterio de parada y en el caso de Control Predictivo Híbrido, en donde el tiempo computacional disponible para obtener una acción de control es limitado, se sugiere ocupar un número máximo de generaciones debido a la capacidad de manejar el tiempo computacional. En el caso que ningún criterio de parada se cumpla, se regresa al paso 3.

El enfoque de solución del problema HFPC basado en Algoritmos Genéticos provee una solución sub-óptima factible para la ley de control, cercana a la óptima global en cada instante. En resumen, los parámetros que se deben sintonizar al ocupar el método GA son el número de individuos, el número de generaciones, la probabilidad de crossover, la probabilidad de mutación y el criterio de detención.

#### **3.2.4 Resultados por simulación**

Los parámetros de sintonía de la función objetivo (12) están dados por  $N_1 = 1$ ,  $N = N_y = N_u = 3$ , and  $\lambda = 0.001$ . El tiempo total computacional requerido para la estrategia HFPC será evaluado en un computador Pentium® 4, 2.80 GHz, y 1.00 GB RAM. El tiempo total de simulación será 6000 s.

Para HFPC-GA la probabilidad de mutación será igual a 0.001, la de crossover igual a 0.7 y el criterio de parada será el número máximo de generaciones. Se fijaron los primeros parámetros y se realizó una sensibilidad para el número de generaciones.

La Figura 20 muestra la función objetivo en función del número de generaciones (y para distintos números de individuos). Basados en la figura, 150 generaciones y 14 individuos son seleccionados para realizar las pruebas. La Figura 21 muestra el tiempo computacional en función del número de generaciones (y para distintos números de individuos). El tiempo computacional es linealmente dependiente del número de generaciones y su pendiente crece cuando se aumenta el número de individuos. A partir

de la figura el tiempo computacional es razonablemente menor al tiempo de simulación (6000 s). Esto significa que todas las estrategias propuestas se pueden aplicar en tiempo real debido al tiempo computacional consumido. Para 150 generaciones con 14 individuos, el tiempo computacional fue aproximadamente 600s (un décimo del tiempo total de simulación) así también el tiempo consumido en cada iteración es mas pequeño que el tiempo de muestreo.

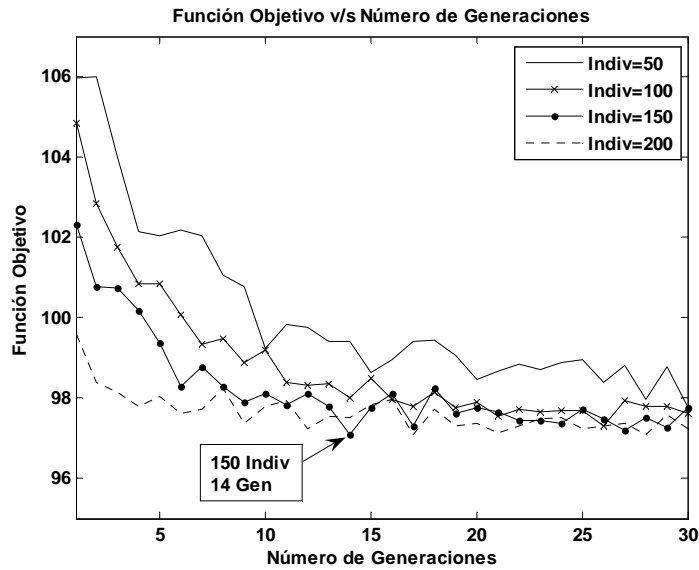


FIGURA 20. EVOLUCIÓN DE LA FUNCIÓN OBJETIVO.

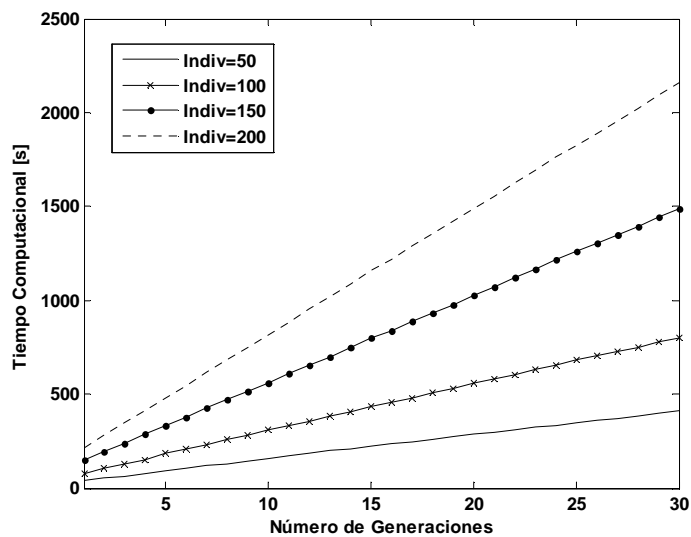


FIGURA 21. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL.



Con los valores óptimos de 150 generaciones con 14 individuos, los resultados de HFPC-GA son obtenidos. Figuras 22 y 23 muestran la variable controlada (nivel del estanque cónico  $h_i$ ) y la variable manipulada (voltaje discreto de la bomba  $u$ ), respectivamente para HFPC-GA y HFPC-EE. La última estrategia evalúa todas las acciones de control factibles a cada instante mientras que HFPC-GA considera solo algunos elementos del espacio de búsqueda, como se explicó anteriormente. Las figuras 24 y 25 muestran las respuestas en detalle para 3500 a 5000 s.

En la Tabla 5 el valor medio de la función objetivo (14) y el tiempo computacional total para la misma simulación son mostrados. La Tabla 6 muestra los estadísticos de la variable controlada y la manipulada.

Como HFPC-GA es un algoritmo de búsqueda heurístico, algunas diferencias con HFPC-EE para la variable manipulada (y por ende la controlada) pueden ser vistas en las figuras 22, 23, 24 y 25 (especialmente en la Figura 25). Sin embargo, la respuesta HFPC-GA es cercana a la solución óptima global dada por HFPC-EE como se muestra en las figuras 8 y 9, así como en la Tabla 5.

Se obtiene una considerable reducción del tiempo computacional usando HFPC-GA en comparación con HFPC-EE. Como se muestra en las tablas 5 y 6, se reducen ligeramente los índices estadísticos para la variable manipulada ( $J_2$  y  $std(\Delta u)$ ).

Sin embargo, el error de seguimiento obtenido es peor tanto en términos de función objetivo ( $J_1$ ) como en desviación estándar ( $std(y-r)$ ) debido a que el algoritmo HFPC-GA no realiza un búsqueda exhaustiva de soluciones.

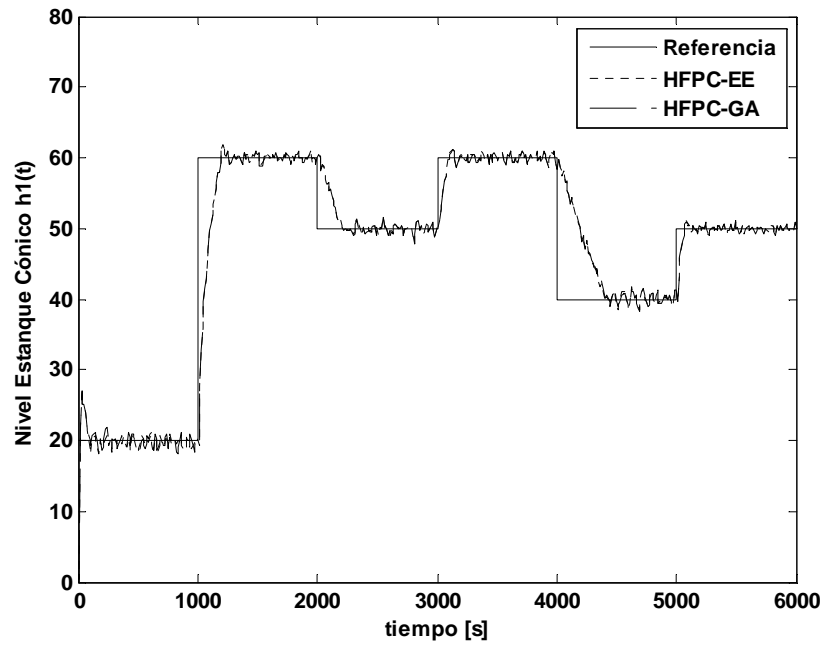


FIGURA 22. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE CONTROLADA.

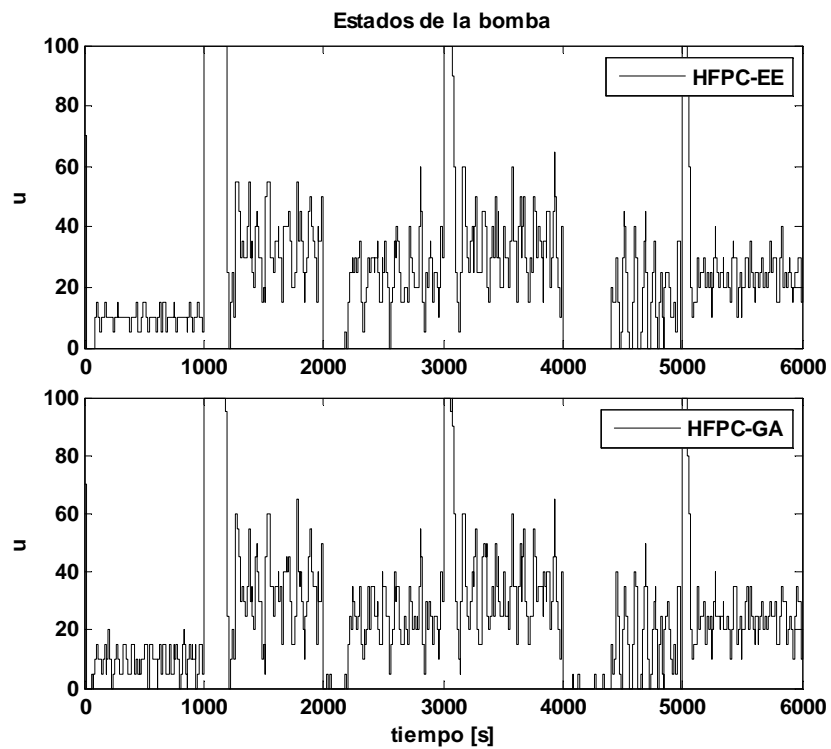


FIGURA 23. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE MANIPULADA.

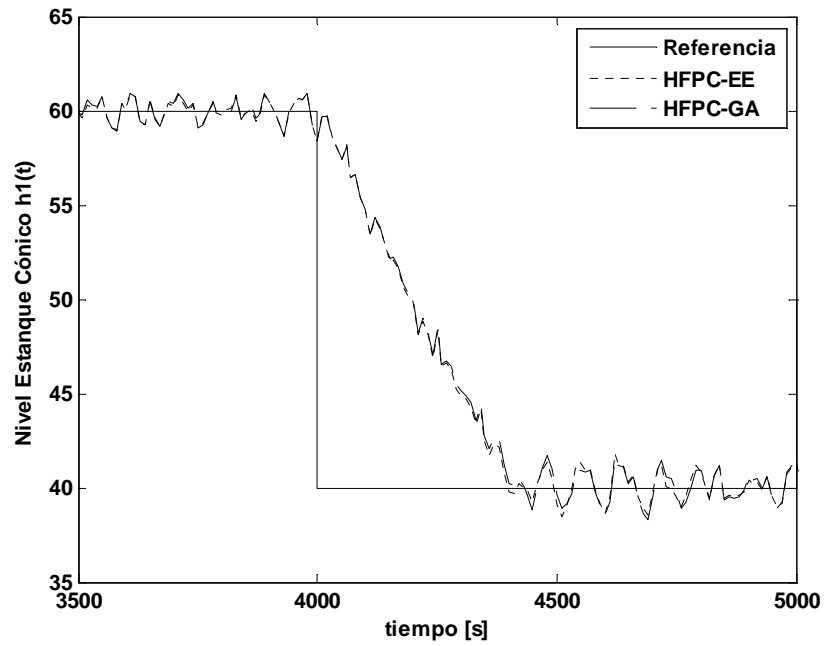


FIGURA 24. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE CONTROLADA. DETALLE DE FIG. 22

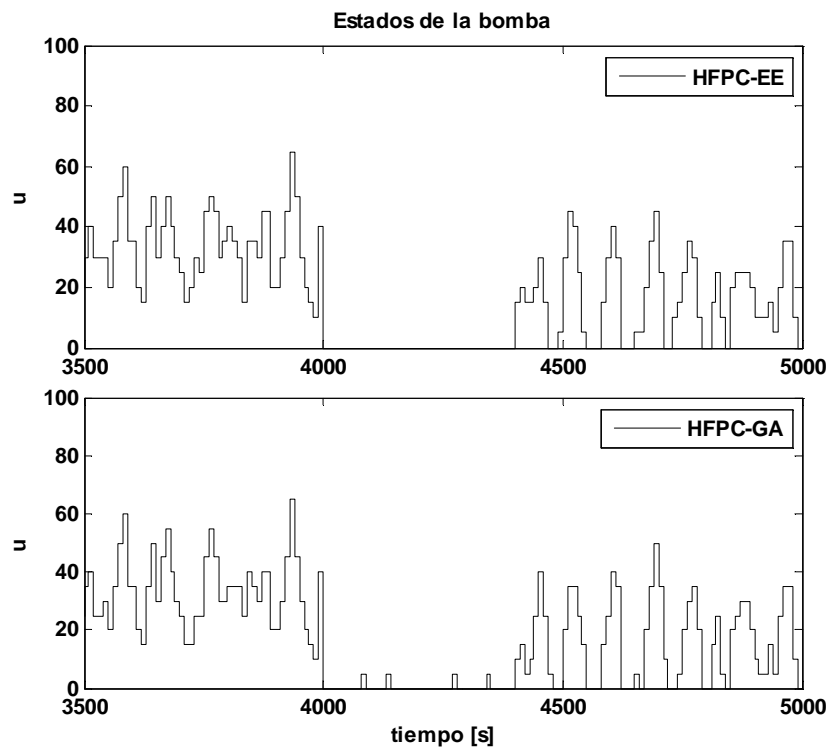


FIGURA 25. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE MANIPULADA. DETALLE DE FIG. 23.

TABLA 5. VALORES MEDIOS DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.

$N_2=N_u=3, \lambda=0.001$	$J_1$	$J_2$	$J$	Tiempo Computacional
HFPC-EE	94.3	487.8	94.8	2850 [s]
HFPC-GA (150,14)	94.8	450.1	95.4	598 [s]

TABLA 6. DESVIACIÓN STANDARD DE (Y-R) Y  $\Delta u$ .

$N_2=N_u=3, \lambda=0.001$	y-r	$\Delta u$
HFPC-EE	5.2095	12.9979
HFPC-GA (150,14)	5.2299	12.4367

### 3.2.5 Síntesis y conclusiones

Se presentó un nuevo enfoque para controlar sistemas no lineales con variables de estado y manipuladas tanto continuas como enteras. Usando una estrategia de control predictivo híbrido y ocupando modelación difusa se pudo resolver el problema de control de un sistema con estados tanto discretos como continuos y variables manipuladas discretas.

El elemento clave de la modelación difusa de sistemas híbridos está en la identificación y detección de las zonas de switch, el cual es realizado mediante el uso de clustering difuso y análisis de componentes principales. La optimización de la función objetivo es en este caso un problema del tipo NP-Hard, el cual pudo ser resuelto eficientemente mediante algoritmos genéticos. El algoritmo de control propuesto HFPC-GA fue exitosamente probado en el control de sistema de estanques híbrido tanto en tiempo computacional como en precisión.

En resumen, el principal aporte del trabajo es la combinación del uso de PCA análisis en la modelación difusa y algoritmos genéticos en obtener la ley de control predictiva híbrida.

## **4.- APLICACIÓN AL PROBLEMA DE RUTEO DINÁMICO DE VEHICULOS**

En este capítulo se presenta una solución basada en técnicas de Inteligencia Computacional para el problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta. Este problema se formula como un problema de Control Predictivo Híbrido basado en variables de estado para el sistema flota-ciudad-clientes. El tiempo y la carga son variables de estado, la secuencia de los vehículos corresponde a la variable de entrada o manipulada y las solicitudes por servicio son perturbaciones.

Para resolver el problema de ruteo dinámico se plantea un controlador predictivo híbrido que asigna las rutas a los vehículos minimizando una función de costos esperada a medida que ocurren solicitudes por servicio en cualquier instante de tiempo. Cuando ocurre un nuevo requerimiento, el controlador ocupa como información las secuencias (rutas) anteriores, las variables de estado y un modelo de predicción de llamadas. Debido al elevado tiempo computacional que toma encontrar la solución óptima global se realiza la optimización en un sub-espacio donde las soluciones satisfacen la restricción de mantener el orden de las rutas preestablecidas para cada vehículo. Como algoritmo de optimización se propone algoritmos genéticos que entregan soluciones cercanas al óptimo dado por enumeración explícita, pero en un tiempo computacional reducido. Para el modelo de predicción de demanda se propone Fuzzy Clustering para zonificar el área urbana de una forma más flexible y que permita reflejar la real distribución tanto espacial como temporal de las posibles llamadas futuras.

### **4.1 Revisión bibliográfica**

El problema de recoger y dejar pasajeros (sistemas dial-a-ride), con o sin ventanas de tiempo, ha sido ampliamente estudiado en la literatura reciente por varios autores, de donde varias formulaciones así como métodos de solución han sido presentados para tratar con ese tipo de problemas (Savelsbergh y Sol, 1995.) Se han propuesto muchos métodos exactos y heurísticos para resolver en tiempo real, tanto el problema determinístico como estocástico, en donde no solo el despachador maneja información confiable a priori con relación a los requerimientos por servicios, sino que trata con

requerimientos desconocidos que entran al sistema en tiempo real mientras los vehículos están en operación.

En los últimos años el interés de estudiar las versiones dinámicas y estocásticas del problema dial-a-ride ha aumentado, principalmente debido al acceso a comunicaciones y tecnologías de información, así como por el interés actual de despachar y rutear sistemas en tiempo real. De acuerdo con Gendreau et al (1996), los problemas de ruteo estocástico de vehículos son caracterizados por demanda estocástica (Gendreau et al., 1995; Yang et al., 2000; Secomandi, 1998, 2000), clientes estocásticos (Waters, 1989), o una mezcla de ambos (Bertsimas et al., 1990; Bertsimas, 1992; Gendreau et al., 1995, 1996). En el caso de versiones dinámicas del problema de ruteo de vehículos, se enfocará el análisis en el problema de recoger y dejar pasajeros (dynamic pick-up and delivery problem DPDP), cuyo resultado final es un conjunto de rutas para todos los vehículos que dinámicamente se van moviendo en el tiempo (Psaraftis, 1980, 1988; Gendreau et al., 1999; Kleywegt y Papastavrou, 1998). Recientemente, se observa aplicaciones reales de estos sistemas usando tecnología de punta (Madsen et al. 1995; Dial, 1995)

Con respecto al uso de información futura para mejorar las decisiones de despacho, en el contexto del problema dinámico de recoger y entregar pasajeros, hay pocas referencias con ejemplos de investigación en esta área. Powell y su equipo (Spivey y Powell, 2004) resuelven el problema de asignar conductores dinámicamente a cargas que aparecen aleatoriamente en el tiempo motivados en aplicaciones de ruteo de camiones que llevan cargas que llegan de aviones transcontinentales. Ichoua et al. (2005) desarrollaron una estrategia basada en conocimiento probabilístico de las llegadas de futuras solicitudes por servicio para manejar una flota de vehículos en un despacho en tiempo real. La nueva estrategia introduce clientes predichos en las rutas de los vehículos para lograr una mejor cobertura del territorio. Los métodos de solución encontrados en esta línea de investigación son diversos, con formulaciones basadas tanto en modelos de redes dinámicas (Powell, 1988) y esquemas tanto dinámicos como estocásticos de programación (Godfrey y Powell, 2002; Topaloglu y Powell, 2005).

Recientemente, Cortés y Jayakrishnan (2004) se dieron cuenta que el problema puede ser modelado bajo un esquema de control predictivo híbrido adaptivo (Hybrid Adaptive Predictive Control - HAPC), considerando que los potenciales re-ruteos de los vehículos

podrían afectar las decisiones de despacho actual, por medio de un costo extra de insertar solicitudes de servicio en tiempo real en las rutas predefinidas mientras los vehículos están en movimiento. Cortés et al. (2005) escribieron una formulación formal del problema DPDP como uno del tipo HAPC, basado en modelos en variables de estado. El estado del sistema está definido en términos del tiempo de llegada a las paradas y la carga del vehículo entre paradas (variables de estado estocásticas). La entrada del sistema (acción de control) son las decisiones de ruteo, las salidas del sistema son los tiempos efectivos de llegada a las paradas y la demanda por servicio es modelada como una perturbación. Los autores usan un modelo discreto con un periodo de muestreo variable en el tiempo, con una función objetivo para el despacho que incluye el efecto de predicción por medio de probabilidades obtenidas de data histórica que muestra los patrones típicos de demanda.

Cortés et al. (2005) pusieron a prueba su modelo con unos datos históricos simulados, concluyendo que al incorporar el poder predictivo al sistema, se logra obtener mejores desempeños en cuanto a la satisfacción del cliente, principalmente debido a ahorros en el tiempo efectivo de espera, comparando con modelos de decisión miopes de despacho (sin poder predictivo). Sin embargo, en los trabajos en congreso, los autores resuelven el problema con un método ineficiente de enumeración explícita (EE), lo cual es muy restrictivo para resolver problemas a gran escala con una intensidad de demanda alta. Además, y con respecto al modelo predictivo, los autores usan una técnica de agrupación muy básica para estimar probabilidades espaciales y temporales de puntos futuros de recogida y entrega.

Se reconoce que en Cortés et al. (2005) hay una contribución relevante en cuanto a formulación del DPDP con un modelo HAPC, así como en la medida de beneficios de su estrategia comparada contra la alternativa miope para un ejemplo simulado. Sin embargo, al enfocarse en los temas no estudiados en detalle por los autores es posible mejorar y sistematizar el esquema de modelación propuesta. Entonces, con el desarrollo de algoritmos eficientes para resolver el modelo, es posible probar una formulación de la función de costos más general y calibrarla vía análisis de sensibilidades. Además, otros parámetros específicos del modelo (asociados a la zonificación espacial y cálculo de las probabilidades) pueden ser mejor estimados usando algoritmos más sofisticados.

Teniendo en cuenta esas limitaciones, en la presente tesis se presenta un algoritmo de solución más eficiente (basado en técnicas heurísticas de Computación Evolutiva) así como un estimador de patrones de viaje más preciso basado en data histórica (Zonificación Difusa). En resumen, las mayores contribuciones que se presentan en este tema son tres. La primera, en el desarrollo de una formulación analítica del espacio de estado del problema, el cual permite al despachador aplicar directamente una variedad de métodos de optimización numérica para enfrentar diferentes condiciones del problema. La segunda es la zonificación difusa la cual permite calcular probabilidades y patrones de viaje a partir de data histórica bajo escenarios más realistas. Finalmente, y basados en la formulación analítica, Algoritmos Genéticos (GA) se proponen y se prueban basados en ejemplos por simulación.

A continuación, se presenta una revisión de la literatura en el uso de métodos heurísticos y metaheurísticos para la resolución de diferentes tipos de problemas de ruteo de vehículos (Vehicle Routing Problem - VRP), tanto para el caso dinámico como estático (Le Bouthillier y Crainic, 2005; Tarantilis, 2005; Tarantilis et al., 2005; Li et al., 2005; Bianchessi y Righini, 2006).

En los últimos años, se han desarrollado muchas modificaciones del bien conocido método de búsqueda Tabú para resolver variantes del VRP, como búsqueda Tabú Granular y búsqueda Tabú basada en memoria adaptiva (Gendreau et al., 1999; Toth y Vigo, 2003; Tarantilis, 2005). Otro método heurístico para el caso dinámico del VRP es uno basado en prioridades propuesto por Tighe et al. (2004).

Como el VRP es un problema NP-Hard, los algoritmos genéticos (GA) basados en técnicas evolucionarias han sido analizados en la literatura especializada. Específicamente, GA han sido aplicados a diferentes versiones del VRP, considerando varias representaciones de los cromosomas y operadores genéticos según el problema en particular. Skrllec et al. (1997) proponen resolver el problema VRP para un vehículo con capacidades ocupando optimización basada en GA logrando reducir el tiempo computacional cuando se ocupa una cierta selección heurística inteligente de la población inicial. Por otro lado, en Filipec et al. (1998) la misma propuesta fue aplicada al problema VRP con varios vehículos.



Zhu (2003) describe algoritmos genéticos especializados basados en parámetros adaptivos para resolver el caso estático del VRP con ventanas de tiempo que previenen de la búsqueda de soluciones una convergencia prematura (obteniendo mínimos locales) y mejorando los resultados cuando se comparan con el típico método de GA. Tong et al. (2004) ocupa el método de optimización GA para resolver el caso estático del VRP con ventanas de tiempo bajo tamaño de flota incierto. Para resolver este problema, una codificación de genes especial asociada al número de vehículos y rutas es presentada. Recientemente, Haghani y Jung (2005) aplicaron el método de optimización basado en GA para el caso dinámico del VRP para múltiples vehículos con tiempos de viajes dependientes del tiempo y ventanas de tiempo suaves. Esta propuesta provee resultados promisorios en términos de tiempo computacional.

Jih y Yung-Jen (1999) y Osman et al. (2005), muestran una comparación favorable para GA contra programación dinámica en términos de tiempo computacional. GA fue usado para resolver el caso dinámico del VRP con ventanas de tiempo y restricciones de capacidad mientras que la programación dinámica se ocupó de resolver un VRP multi-objetivo. Un método híbrido incluyendo ambos algoritmos es descrito, logrando resultados precisos y en un razonable tiempo computacional.

Con respecto a otras heurísticas usadas en el contexto del caso dinámico del VRP, se han aplicado nuevas metaheurísticas inspiradas en la conducta de colonias de hormigas reales (ant colony methods) para resolver esos problemas (Montemanni et al. 2005; Dréo et al. 2006). Esos métodos son especialmente apropiados para resolver eficientemente problemas de optimización combinatorial y son caracterizados por la combinación de un enfoque constructivo y uno basado en mecanismos de aprendizaje (Dorigo y Stutzle, 2004). Montemanni et al. (2005) también aplican el método de colonia de hormigas en un caso de estudio real obteniendo resultados promisorios. Dréo et al. (2006) presentan buenos resultados para el caso estático del VRP por medio de la optimización del tamaño de la flota así como los planes de rutas de los vehículos. Las dos metaheurísticas descritas (GA y colonia de hormigas) se han aplicado solamente para el caso miope dinámico del VRP, sin considerar escenarios futuros de demanda para mejorar las decisiones actuales de despacho. En esta tesis se muestra una aplicación de GA en una formulación no miope para el caso dinámico del VRP, basado en un esquema de HAPC.

La estructura de este capítulo es la siguiente. En una primera parte se presenta el modelo en variables de estado para carga y tiempo de viaje, en el cual la estocasticidad en los tiempos de viaje se debe a la demanda. Se muestra en detalle la formulación en variables de estado para el DPDP. A continuación, se propone una zonificación basada en clustering difuso para obtener un flexible y sistemático uso de los patrones históricos de llamadas y su calibración. En la siguiente sección, se propone el uso de GA para resolver el problema mostrado en la parte anterior. A continuación, las ventajas en términos de flexibilidad en la formulación que permiten obtener soluciones genéricas cuando se usa el nuevo controlador predictivo difuso basado en un esquema de solución con GA se cuantifican mediante experimentos de simulación.

## **4.2 Descripción del problema**

Uno de los objetivos de la tesis es formular el caso general de MDPDP con el enfoque de Control Predictivo Híbrido (HPC), reconociendo que el proceso de ruteo dinámico incluye variables de estado tanto continuas como enteras así como variables de entrada enteras.

Específicamente, en esta tesis se desarrolla una estructura que incorpora estocasticidad en las reglas de ruteo en el contexto de problema dinámico de recoger y dejar pasajeros. El objetivo aquí es rutear pasajeros, sin embargo el esquema puede ser ocupado también para el ruteo de cargas.

La formulación está basada en la siguiente premisa: el tiempo esperado de viaje (o de espera) de todos los pasajeros ya asignados será fuertemente afectado por cualquier decisión futura que se realice para satisfacer un nuevo requerimiento de servicio. Se introduce una predicción estocástica en las reglas de ruteo para decidir el orden de la asignación de manera de incorporar de una forma más realista el tiempo efectivo de viaje (o de espera) experimentado por los usuarios así como el del operador en la formulación del costo de decisión, el cual eventualmente podría cambiar alguna decisión de ruteo previamente tomada, obteniendo mejores soluciones que sean más cercanas al óptimo dinámico del sistema.

La idea de usar un esquema basado en control predictivo híbrido es la de poder obtener despachos más precisos con un costo reducido bajo demanda estocástica, generada dinámicamente en el tiempo sin tener en el módulo de despacho ningún conocimiento previo de esos nuevos requerimientos en el momento de la llamada por servicio. Bajo ese esquema de modelación, el uso de información pasada sobre la dinámica del sistema puede permitir predecir su comportamiento, información con la cual se pueden lograr mejores decisiones de ruteo al ser incorporada en la función objetivo que captura las componentes de costos más relevantes. Esta información podría eventualmente ser actualizada online lo cual permitiría adaptar el modelo de predicciones. Esto podría ser beneficioso si se aplica control durante escenarios en los cuales la información histórica parece no tener relación con lo que ocurre en el sistema actualmente.

Asuma que se tiene un área de influencia  $A$ , con un servicio de tránsito en una red de largo  $D$  en unidades de distancia. Suponga que se tiene un conjunto de vehículos  $V$  de tamaño de flota  $F$  y capacidad máxima de pasajeros. La flota de vehículos se encuentra en operación viajando dentro del área de cobertura según reglas de ruteo predefinidas. La demanda por servicio es desconocida y generada dinámicamente en tiempo real (asumimos una tasa de llegada  $\mu$  en llamadas por unidad de tiempo). Las decisiones de ruteo deben ser tomadas en tiempo real para manejar la demanda con los vehículos disponibles (flota fija).

En cualquier instante de tiempo  $k$  asuma que el vehículo  $j \in V$  ha sido asignado a la secuencia de tareas que incluyen recogidas y entregas, que puede ser representada por una función  $S_j(k) = [s_j^0(k) \ s_j^1(k) \ \dots \ s_j^i(k) \ \dots \ s_j^{w_j}(k)]^T$ , en la cual el  $i$ -ésimo elemento de la secuencia representa a la parada  $i$ -ésima en la ruta del vehículo  $j$ ,  $w_j$  es el número de paradas. La condición inicial (denotada por  $s_j^0(k)$ ) debe ser actualizada para indicar la posición del vehículo  $j$  en cualquier instante de tiempo  $k$ . La acción de control completa o variable manipulada  $u(k) = S(k)$  puede ser visualizada como la decisión de un módulo despachador, el cual toma las decisiones de ruteo dinámico. En el instante  $k$ , esa acción de control es representada por un conjunto de secuencias asignadas a todos los vehículos en el instante  $k$ . Analíticamente,

$$u(k) = S(k) = [S_1(k), \dots, S_j(k), \dots, S_F(k)] \quad (20)$$

Los vehículos viajarán de acuerdo a su secuencia anterior  $S(k-1)$  mientras no se reciban nuevas llamadas que provoquen re-ruteos. Cuando un nuevo requerimiento por servicio (llamada) aparece, el controlador (o central despachadora), calcula las secuencias para el siguiente instante  $S(k)$  (acción de control) para toda la flota de vehículos incluyendo las paradas requeridas por el nuevo usuario. Entonces, cada secuencia  $S_j(k)$  se mantiene fija durante todo el intervalo de tiempo  $(k, k+1)$ , a no ser que el vehículo pase por alguna parada programada durante dicho intervalo, en cuyo caso su secuencia disminuirá en largo mostrando que la tarea programada ya fue realizada.

Las decisiones de despacho son tomadas por el controlador en tiempo real basada en un criterio de inserción específico usado por el módulo despachador. En esta aplicación, la regla es escoger aquella  $S_j(k)$  que reporta el mínimo costo de inserción, minimizando una función objetivo  $J$ , la cual en nuestro caso penalizará el tiempo de viaje y de espera de los pasajeros. La acción de control seleccionada deberá no solo insertar dentro de sus rutas a la nueva llamada sino que además de ser factible deberá ser la que le signifique al sistema un mínimo costo  $S_j^*(k)$ .

Un modelo dinámico con un paso fijo (dado por el tiempo de muestreo) no es directamente aplicable a este caso, debido a que el controlador predictivo toma las decisiones de ruteo en el momento en que una nueva llamada entra en el sistema. Entonces, en este esquema es necesario formular el problema en términos de paso de tiempo variable, el cual estará dado por el intervalo de tiempo que ocurre entre dos requerimientos consecutivos.

Las decisiones de ruteo dinámicas asociadas a las especificaciones de la función de costo dependen fuertemente del estado del sistema en el instante  $k$  (es decir, cuando el requerimiento  $k$  entra al sistema). Los atributos del vehículo incluyen todas las características cuando se deja la parada  $i$ -ésima de su secuencia. En esta formulación el vector de atributos incluyen el instante de tiempo  $T_j^i(k)$  y la carga del vehículo  $L_j^i(k)$ , después que el vehículo  $j$  deja la parada  $i$ . En un contexto más general, otras variables

podrían ser agregadas para obtener una mejor imagen de los estados de los vehículos, como por ejemplo la velocidad del vehículo  $j$  para capturar congestión de tráfico no recurrente, el tiempo acumulativo de viaje experimentado por todos los pasajeros en los vehículos antes de dejar la parada  $i$ , y por ejemplo alguna medida del espacio de disponibilidad en otros vehículos para pasajeros potenciales en las proximidades del un segmento  $(i, i+1)$  (Cortes, 2003; Cortés y Jayakrishnan, 2004).

En la Figura 26 se muestra una representación gráfica de una secuencia típica de un vehículo calculada en cierto instante  $k$ .

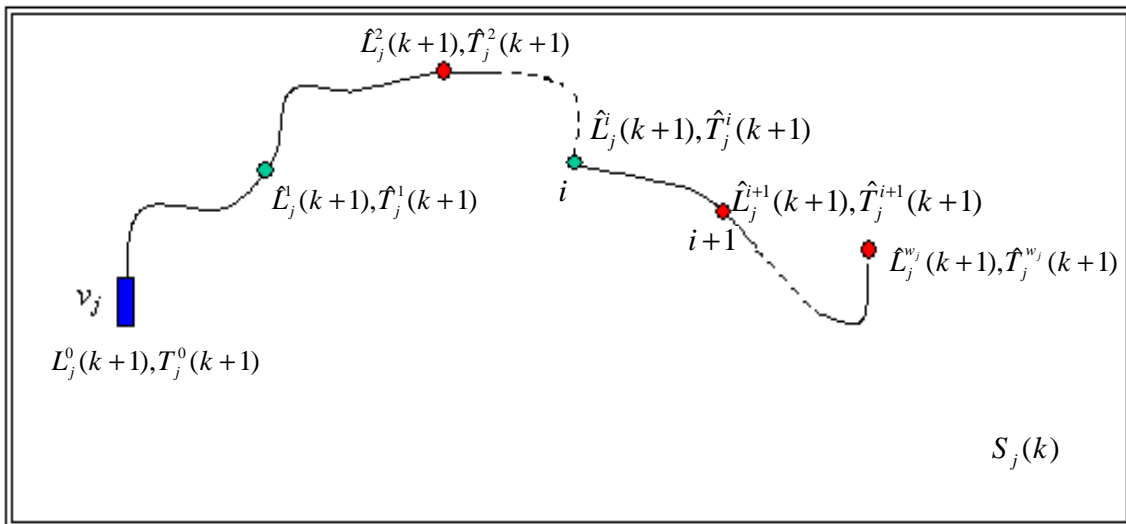


FIGURA 26. RUTA TÍPICA DE UN VEHÍCULO EN EL INSTANTE K.

Como se mencionó anteriormente, cuando una nueva llamada entra al sistema, el controlador, decide cual vehículo en servicio servirá el nuevo requerimiento dentro de su ruta, en base a la función de costos del sistema, la cual depende de estimaciones de los tiempos efectivos de viaje y de espera de los pasajeros. Para evaluar el desempeño del controlador, los tiempos efectivos de espera y de viaje de cada pasajero deben ser calculados solamente cuando el requerimiento sea completamente satisfecho (recogida y entrega).

El vector de salidas  $y(k) = T(k) = [T_1(k), \dots, T_j(k), \dots, T_F(k)]$  es calculado midiendo los tiempos observados de viaje entre segmentos  $T_j^{(i,i+1)}(k) = T_j^{i+1}(k) - T_j^i(k)$ , hasta que

ocurre una nueva llamada para todos los vehículos y todos los pares de paradas consecutivas  $(i, i+1)$ . La decisión de despacho dinámica depende mayormente de una función de costo en términos de los segmentos de viaje  $T_j^{(i,i+1)}(k)$ . Entonces, se justifica el usar los tiempos efectivos de viaje y de espera de los pasajeros como un buen indicador del desempeño del sistema sobre las decisiones que fueron tomadas.

Analíticamente, la ecuación (1) puede ser escrita para el MDPDP, reconociendo la dependencia entre el proceso de ruteo y las siguientes variables asociadas:

$$\begin{aligned} x(k) &= [L(k), T(k)]^T \\ y(k) &= T(k) \\ u(k) &= S(k) \end{aligned} \quad (21)$$

donde

$$\begin{aligned} T(k) &= [T_1(k), \dots, T_j(k), \dots, T_F(k)] \\ L(k) &= [L_1(k), \dots, L_j(k), \dots, L_F(k)] \end{aligned} \quad (22)$$

Como se discutirá a continuación, las funciones  $f$  y  $g$  de la ecuación (1) son no lineales para el sistema de ruteo propuesto.

El esquema de control predictivo híbrido propuesto en esta tesis se representa por el diagrama de bloques mostrado en la Figura 27.

En la figura, el controlador predictivo híbrido es representado por el módulo despachador, el cual toma decisiones de ruteo en tiempo real basado en información que tiene del sistema de ruteo (proceso) y los valores esperados para tiempos de viaje y atributos de la flota (modelo). Un mecanismo adaptativo también es agregado en la figura, representando la necesidad de adaptar los parámetros del modelo. A continuación, se describe el modelo en variables de estado, resaltando el tratamiento de las componentes de tanto el tiempo de salida como de la carga. De aquí en adelante, el objetivo es definir una formulación en variables de estado para el problema dinámico de recoger y dejar pasajeros, y descrito en la siguiente sección.

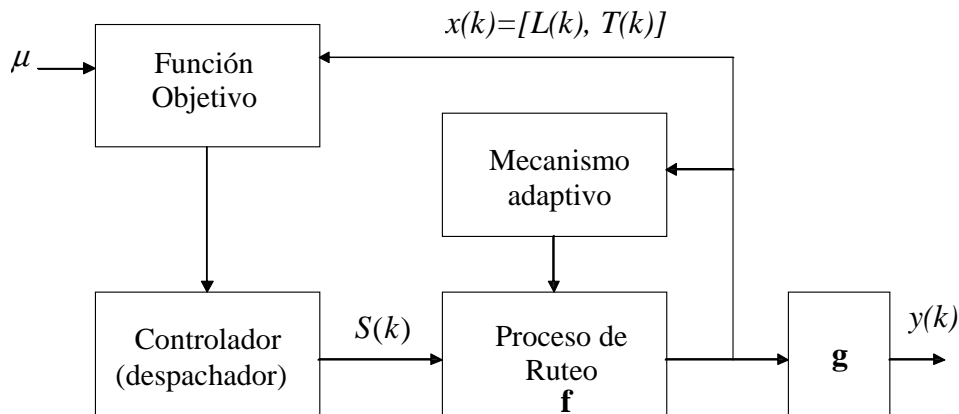


FIGURA 27. DIAGRAMA DE BLOQUES COMPLETO PARA EL ENFOQUE HÍBRIDO PREDICTIVO PARA EL CALCULO DE RETARDOS OCASIONADOS POR RE-RUTEOS.

### 4.3 Control predictivo híbrido para el ruteo de vehículos

En esta sección, se formaliza el problema dinámico de recoger y entregar (Dynamic Pickup and Delivery Problem DPDP) bajo un esquema de Control Predictivo Híbrido Adaptivo (Hybrid Adaptive Predictive Control HAPC). El sistema se formaliza en términos de variables de estado y una función objetivo. El tamaño de la flota es asumida conocida ya que se supone que el sistema se encuentra bien dimensionado en función de la demanda que debe satisfacer y un cierto nivel de calidad de servicio exigido por los usuarios. La función de costo no incluye ventanas de tiempo ni para las paradas de recogidas o entregas, sin embargo es un tema muy interesante de trabajo futuro. Se aproxima el costo de operación por el tiempo total que el vehículo viajó y el costo de los usuarios considerado por tanto tiempo de espera y de viaje.

El servicio consiste en recoger y dejar pasajeros a coordenadas específicas dentro del área urbana, las cuales son conocidas solamente después del correspondiente requerimiento recibido en tiempo real por el despachador. Los vehículos tienen que ser rápidamente re-ruteados (desde sus tareas iniciales) de manera de ubicar el nuevo requerimiento dentro de la ruta de alguno de los vehículos mientras ellos están en movimiento. Las decisiones de ruteo son tomadas basándose en la minimización de una función objetivo que depende de las variables de estado asociadas al estatus en tiempo real de los vehículos. Esas variables de estado pueden incluir todas las características

importantes del vehículo, las cuales son en nuestro caso, el tiempo de arribo esperado y la carga esperada. Además, asumimos que hay data histórica disponible, tanto de las posiciones de recogida como de entrega (en términos de coordenadas) así como los instantes de tiempo en que ocurrieron las llamadas. Esta información es fundamental en el modelo predictivo explicado a continuación.

En la sección 4.3.1 se formula el modelo dinámico, como una extensión del propuesto por Cortés et al. (2005), para el problema específico de rutear una flota de  $F$  vehículos con capacidad limitada, para servir una demanda en tiempo real, distribuida en un área urbana. Las condiciones del tiempo de viaje y la estructura de la red se simplifican considerando velocidad constante de los vehículos cuando se mueven de una parada a otra. En la sección 4.3.2, se resume la función objetivo general propuesta para la formulación en Cortés et al. (2005), y se muestra en detalle los casos de dos y tres pasos en la predicción. Finalmente, se modela la política operacional a través de una formulación analítica para mostrar la estructura del problema general y visualizar sus ventajas.

#### 4.3.1 Modelo del proceso (variables de estado tiempo y carga)

En el contexto de la teoría de control, los sistemas híbridos se caracterizan por tener variables tanto continuas como enteras. Específicamente, los sistemas híbridos pueden expresarse como un sistema en variables de estado no lineal dado por:

$$\begin{aligned}x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k))\end{aligned}\tag{23}$$

Donde  $x(k)$  son las variables de estado continuas y/o discretas (enteras),  $u(k)$  son las entradas o variables manipuladas continuas y/o enteras e  $y(k)$  define las salidas del sistema ya sean continuas y/o discretas. En general, un diseño basado en Control Predictivo Híbrido minimiza la siguiente función objetivo, (Bemporad and Morari, 1999):

$$\min_{\{u(k), u(k+1), \dots, u(k+N-1)\}} J(u(k), \dots, u(k+N-1), \hat{x}(k+1), \dots, \hat{x}(k+N), \hat{y}(k+1), \dots, \hat{y}(k+N))\tag{24}$$



Donde  $J$  es una función objetivo,  $k$  es el tiempo actual,  $N$  el horizonte de predicción,  $\hat{x}(k+t)$ ,  $\hat{y}(k+t)$  son el vector de espacio de estado y la salida del sistema esperada en el instante  $k+t$  respectivamente, y  $\{u(k), \dots, u(k+N-1)\}$  representa la secuencia de control, la cual corresponde al vector de variables de optimización. Una vez que la expresión (24) es optimizada, solamente el primer elemento del vector de control es usado y se actualiza las condiciones del sistema, basándose en el horizonte deslizante.

La modelación del DPDP requiere un paso o tiempo de muestreo variable ( $\tau$ ), a diferencia de los esquemas tradicionales HPC en los cuales los pasos suelen ser fijos. En este caso, los eventos del sistema son provocados por los requerimientos por servicio específicos que ocurren en instantes de tiempo no fijos, lo cual justifica el uso de pasos variables.

En cualquier instante  $k$ , cada vehículo que pertenece a la flota tiene asociada una secuencia de tareas (paradas). Analíticamente  $S_j(k)$  representa la secuencia de paradas asignadas al vehículo  $j$  en el instante  $k$ . Como se dijo en la sección anterior, las variables de estado consideradas aquí son los tiempos de salida esperada y la carga esperada cuando los vehículos dejan cada parada perteneciente a sus secuencias asignadas. En resumen,  $T_j^i(k)$  y  $L_j^i(k)$  representan el tiempo estimado de llegada y la carga cuando el vehículo  $j$  abandona la parada  $i$ , calculados en el instante de tiempo  $k$ , respectivamente. El conjunto de secuencias  $S(k) = [S_1(k), \dots, S_j(k), \dots, S_F(k)]$  asociadas a los vehículos corresponden a la variable manipulada  $u(k)$  y los requerimientos por servicio corresponden a perturbaciones en el modelo.

En resumen, en el instante de tiempo  $k$ , cada vehículo  $j$  tiene asociada una secuencia de paradas asignadas  $S_j(k)$ , y dos vectores que contienen la hora estimada de partida  $T_j^i(k)$  y la carga estimada al salir de cada parada  $L_j^i(k)$ , cada vector con dimensión  $w_j(k)+1$ , donde  $w_j(k)$  es el número de paradas asignadas al vehículo  $j$  en el instante  $k$ . Analíticamente:

$$S_j(k) = \begin{bmatrix} S_{01}(k) & S_{02}(k) & P_j^0(k) & label_0(k) \\ S_{11}(k) & S_{12}(k) & P_j^1(k) & label_1(k) \\ S_{21}(k) & S_{22}(k) & P_j^2(k) & label_2(k) \\ \vdots & \vdots & \vdots & \vdots \\ S_{w_j(k)1}(k) & S_{w_j(k)2}(k) & P_j^{w_j(k)}(k) & label_{w_j(k)}(k) \end{bmatrix}, \quad (25)$$

$$T_j(k) = \begin{bmatrix} T_j^0(k) \\ T_j^1(k) \\ T_j^2(k) \\ \vdots \\ T_j^{w_j(k)}(k) \end{bmatrix}, L_j(k) = \begin{bmatrix} L_j^0(k) \\ L_j^1(k) \\ L_j^2(k) \\ \vdots \\ L_j^{w_j(k)}(k) \end{bmatrix} \quad (26)$$

Donde  $S_{i1}(k)$  y  $S_{i2}(k)$  indican si la siguiente parada  $i$  es una recogida o una entrega,  $P_j^i(k)$  cuantifica la posición geográfica de la parada  $i$  asignada al vehículo  $j$  en términos de coordenadas espaciales, y  $label_i(k)$  indica al pasajero que está llamando.  $P_j^0(k)$ ,  $L_j^0$ ,  $T_j^0$  (primera fila de la matriz secuencia) representa las condiciones iniciales del vehículo en el instante  $k$  (posición espacial, carga y tiempo de reloj de referencia).

En la Figura 28, se muestra un ejemplo para una secuencia específica, indicando la secuencia  $S_1(k)$ , la carga esperada  $L_1(k)$  (pasajeros/vehículo) y tiempo de arribo esperado a las paradas (minutos)  $T_1(k)$  en el instante  $k$ . De aquí en adelante, y por simplicidad, las paradas por recogidas y entregas serán representadas con un signo mas (+) o un signo menos (-) como superíndice al pasajero  $label_i(k)$  al cual pertenece la parada. En otras palabras, para el cliente  $r$ , su recogida y su entrega serán denotadas por  $r^+$ ,  $r^-$  respectivamente.

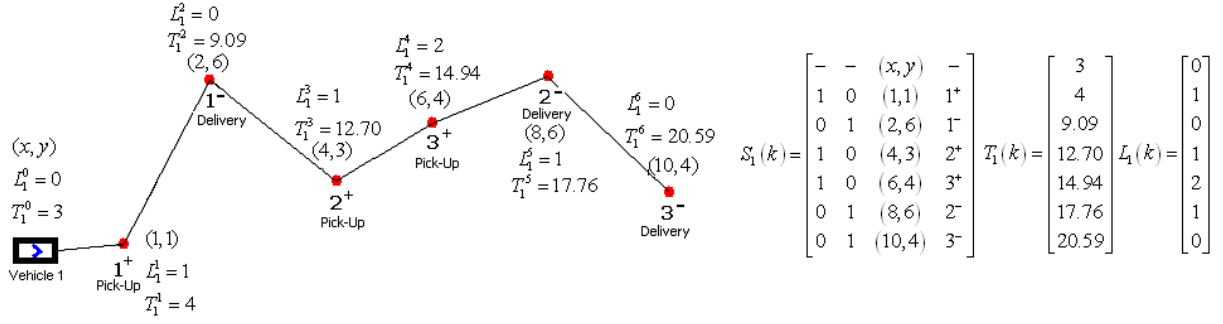


FIGURA 28. EJEMPLO DE SECUENCIA DEL VEHÍCULO 1 Y VARIABLES DE ESTADO ASOCIADAS.

En Cortés et al (2005) se propuso las siguientes ecuaciones dinámicas para describir el proceso:

$$\hat{x}(k+1) = \begin{bmatrix} E\{L(k+1)/k\} \\ E\{T(k+1)/k\} \end{bmatrix} = \begin{bmatrix} \hat{L}(k+1) \\ \hat{T}(k+1) \end{bmatrix} = \begin{bmatrix} f_L(L(k), S(k)) \\ f_T(T(k), S(k)) \end{bmatrix} \quad (27)$$

$$y(k) = T(k) = [T_1(k), \dots, T_j(k), \dots, T_F(k)]$$

donde

$$\begin{aligned} T(k) &= [T_1(k), \dots, T_j(k), \dots, T_F(k)] \\ L(k) &= [L_1(k), \dots, L_j(k), \dots, L_F(k)] \\ S(k) &= [S_1(k), \dots, S_j(k), \dots, S_F(k)] \end{aligned} \quad (28)$$

Las funciones  $f_L$  y  $f_T$  en la ecuación (27), corresponden al modelo en variables de estado que se detalla a continuación.

$$E\{L(k+1)/k\} = \hat{L}(k+1) = f_L(L(k), S(k)) = A_L L(k) + B_L(S(k)) \quad (29)$$

En donde cada vector de carga se puede descomponer como:

$$L_j(k) = \left[ L_j^0(k) \quad L_j^1(k) \quad \cdots \quad L_j^{w_j}(k) \right]_{(w_j+1) \times 1}^T \quad (30)$$

Tal y como se mencionó antes,  $L_j^i(k)$  representa la carga estimada del vehículo  $j$ , cuando deja la parada  $i$  (estimada en el tiempo  $k$ ), y  $w_j$  es el número de paradas que pertenecen a la secuencia del vehículo. Las matrices correspondientes al modelo en variables de estado, se calculan como:

$$B_L(S(k))_{(w_j+3) \times 1} = B_L^2 \cdot (S(k) \cdot B_L^1)$$

$$A_L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)} \quad ; \quad B_L^1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}_{1 \times 4}$$

$$B_L^2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

La matriz de secuencia  $S(k)$  y la carga esperada  $\hat{L}(k+1)$  pueden cambiar de dimensiones dinámicamente. Entonces las dimensiones de las matrices  $A_L, B_L^1, B_L^2$  son variables y dependen de la matriz secuencia, con lo que se obtiene un modelo en variables de estado adaptivo.  $B_L^1$  es diseñado para quitar las dos últimas columnas de la secuencia, las cuales no son necesarias para calcular los cambios en la carga instantánea a instante.

La característica de coordenadas para las paradas en la matriz de secuencia es necesaria para estimar el tiempo de viaje entre las paradas. Es necesario que por otro lado, las secuencias que se impongan al sistema sean factibles, por ejemplo, que cumplan con la restricción de precedencia. Comprobar que las secuencias sean factibles no es función de

las ecuaciones dinámicas sino que del despachador, quien debe ingresar al sistema variables de decisión factibles.

La ecuación dinámica para el tiempo de salida esperado se obtiene con la misma metodología.

$$E\{T(k+1)/k\} = \hat{T}(k+1) = f_T(T(k), S(k)) = A_T \cdot T(k) + B_T(S(k)) \quad (31)$$

Cada vector de tiempo se puede descomponer como:

$$T(k) = \left[ T_j^0(k) \quad T_j^1(k) \quad \dots \quad T_j^{w_j}(k) \right]_{(w_j+1) \times 1}^T$$

Donde  $T_j^i(k)$  representa el tiempo de salida del vehículo  $j$  a la parada  $i$  estimada en el instante  $k$  y  $w_j$  es el número de paradas pertenecientes a la secuencia del vehículo deseada. Las matrices correspondientes al modelo en variables de estado de la ecuación (31) son:

$$B_T(S(k))_{(w_j+3) \times 1} = \text{dist} \left( B_T^1 \cdot (S(k)) \cdot B_T^2 \right)$$

$$A_T = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)} \quad ; \quad B_T^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{4 \times 1}$$

$$B_T^1 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

La función *dist* del vector de coordenadas resultantes, se ocupará de estimar tiempos de viajes entre pares de coordenadas. Por el momento se considerará que el tiempo de viaje es proporcional a la distancia entre dos puntos. Esa constante de proporcionalidad se supondrá constante y deberá representar a la velocidad promedio de los vehículos dentro de la red urbana, incorporando dentro de esa constante todos los efectos estocásticos que afectan a los tiempos de viajes reales, como semáforos, congestiones, etc. Es evidente que esta simplificación es importante y en un trabajo futuro se relajará, incorporando como variable de estado a la velocidad, pudiendo entonces imponer estocasticidad en los tiempos de viajes.

Al igual que para las ecuaciones del modelo dinámico de la carga, las matrices  $A_T, B_T^1, B_T^2$  cambiarán sus dimensiones dinámicamente, dependiendo de la matriz secuencia del vehículo. Estas últimas definiciones completan el modelo dinámico adaptivo de las variables de estado del proceso de ruteo.

A parte de satisfacer las ecuaciones dinámicas, el proceso tiene restricciones lógicas que le dan complejidad al proceso. Es así que la secuencia de los vehículos y las variables de estado tienen que cumplir un conjunto de restricciones que dependen en las condiciones reales del DPDP modelado. Entre estas restricciones, se puede identificar la de precedencia, restricciones de capacidad, entre otras, las cuales son escritas como restricciones lógicas como se muestra a continuación.

**Restricción 1.-**  $S_{1i}$  y  $S_{2i}$  son binarias y distintas, debido a que una parada puede ser o una recogida o una entrega, pero nunca ambas.

$$S_{i1}, S_{i2} \in \{0,1\}, S_{i1} \neq S_{i2} \Leftrightarrow S_{i1} + S_{i2} = 1.$$

**Restricción 2.-** De precedencia. La entrega de un pasajero no puede ocurrir antes que su recogida.

$$\forall i, j, i < j \leq w_j(k), label_i(k) = label_j(k) \Rightarrow s_{1i} = 1 \wedge s_{1j} = 0$$

Si dentro del conjunto de etiquetas un pasajero aparece una sola vez, significa que actualmente es usuario del servicio y que por lo tanto su parada asociada corresponde a una entrega.

$$\forall i, label_i(k) \neq label_j(k) \quad \forall j \leq w_j(k) \Rightarrow s_{1i} = 0$$

Es fácil deducir de esta restricción que todas las secuencias terminan en una entrega.

$$S_{n1} = 0, S_{n2} = 1.$$

**Restricción 3.-** Un destino  $P_j^i(k)$  correspondiente a un pasajero  $label_i(k)$  debe ser visitado solo una vez. Esta restricción es tratada analíticamente en las ecuaciones (32)-(36).

**Restricción 4.-** Consistencia. Una vez que un pasajero es recogido por un vehículo, el (o ella) debe ser entregado en su destino por el mismo vehículo. Esta restricción se trata también analíticamente en las ecuaciones (32)-(36).

**Restricción 5.-** Restricción de carga máxima. Un vehículo no podrá llevar más pasajeros que su carga máxima, esto es  $L_j^i(k+t) \leq L_{\max}$ .

En general, los tiempos de viaje en áreas urbanas presentan una gran variabilidad durante un día normal, debido a muchos factores como accidentes, tráfico y condiciones climáticas (Ceylan and Bell, 2004). Por simplicidad, ocuparemos una red conceptual con norma euclídeana como estimador de las distancias y se considerará velocidad constante para los vehículos. Notar que en el modelo, la única fuente de estocasticidad que afecta a los tiempos efectivos de viaje y de espera y es provocado por los nuevos requerimientos que ocurren en tiempo real, los cuales obligan a los vehículos adquirir nuevas rutas para satisfacer la nueva demanda con el respectivo aumento del tiempo esperado de recogida o de entrega de los pasajeros. Fuentes de estocasticidad adicional pueden ser incorporadas en el modelo cambiando las variables de estado (agregando nuevas variables) y cambiando la función objetivo (modificando ciertas componentes dependiendo del nuevo modelo en variables de estado).

En esta tesis, se trabaja con la variable manipulada  $S(k)$  para tomar las decisiones extrayendo de ella una matriz binaria de activaciones  $G=(g_{ir})_{\substack{i=1..n \\ r=1..n}}$  la cual está asociada con las paradas  $P_j^i(k)$ .

Se tiene que  $n = w_j(k)$  y cada elemento de la matriz  $g_{ir} \in \{0,1\}$  representa la  $r$ -ésima activación de la parada  $i$ .

Una parada  $P_j^i(k)$  asociada al pasajero  $label_i$  asignado al vehículo  $j$ , puede ser escrita como una combinación lineal de todas las paradas conocidas ( $f_1, f_2, \dots, f_n$ ) usando los factores de activación  $g_{ir}$ . Analíticamente:

$$P_j^i(k) = g_{i1}f_1 + g_{i2}f_2 + \dots + g_{ii}f_i + \dots + g_{in}f_n \quad (32)$$

donde

$$g_{ir} = \begin{cases} 0 & f_r \text{ no es la parada } i \\ 1 & f_r \text{ es la parada } i \end{cases} \quad (33)$$

Entonces, el vector de posiciones  $P_j(k)$ , excluyendo la condición inicial  $P_j^0(k)$ , puede ser escrito como sigue:

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1(n-1)} & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2(n-1)} & g_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1} & g_{(n-1)2} & \cdots & \cdots & g_{(n-1)(n-1)} & g_{(n-1)n} \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{n(n-1)} & g_{nn} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G \cdot f \quad (34)$$



Con estas ecuaciones (34), la restricción 3 puede ser escrita en términos de variables lógicas. Las siguientes ecuaciones en términos de las  $g_{ir}$  representan la restricción 3 que dice que una parada debe ser visitada solo una vez.

$$g_{i1} + g_{i2} + \dots + g_{in} = 1, \quad \forall i = 1, \dots, n \quad (35)$$

$$g_{1r} + g_{2r} + \dots + g_{nr} = 1, \quad \forall r = 1, \dots, n \quad (36)$$

Dentro del conjunto de paradas, se puede reconocer cuales de ellas son recogidas y cuales son entrega. Ocupando esa información, se escriben mediante relaciones analíticas entre elementos de la matriz  $G$  tanto la restricción de precedencia como todas las otras. Por ejemplo, usando la matriz  $G$  la carga esperada se puede escribir como la suma de la carga inicial más todas las activaciones de las recogidas anteriores a la parada menos las activaciones de todas las entregas anteriores, como se muestra en (37):

$$\hat{L}_j(k+t) = \left[ L_j^0(k) \quad \dots \quad L_j^0(k) + \sum_{m=1}^i \left( \sum_{k_p: f_{k_p} \text{ is a pickup}} g_{mk_p} - \sum_{k_d: f_{k_d} \text{ is a delivery}} g_{mk_d} \right) \quad \dots \quad 1 \quad 0 \right]^T \quad (37)$$

Usando (37), la restricción de carga máxima (restricción 5) puede ser escrita ocupando los factores de activación de la matriz  $G$ . Analíticamente:

$$L_j^0(k) + \sum_{m=1}^i \left( \sum_{k_p: f_{k_p} \text{ is a pickup}} g_{mk_p} - \sum_{k_d: f_{k_d} \text{ is a delivery}} g_{mk_d} \right) \leq L_{\max} \quad i = 2, \dots, n-1 \quad (38)$$

Además, y para completar el modelo en variables de estado, el tiempo de llegada puede ser expresado como función de la matriz  $G$  según se muestra en (39).

$$\hat{T}(k+t) = \left[ T^0(k) \quad T^0(k) + G^1 Q (G^2)^T \quad \dots \quad T^0(k) + \sum_{r=1}^{i-1} G^r Q (G^{r+1})^T \quad \dots \quad T^0(k) + \sum_{r=1}^{w_j(k+t)-1} G^r Q (G^{r+1})^T \right]^T \quad (39)$$

Donde  $G^r$  denota la fila  $r$ -ésima de  $G$ ,  $Q$  es una matriz que contiene los tiempos de viajes entre todas las paradas (a partir de estimaciones basados en distancia euclídeana y velocidad constante).

En este modelo, tanto matrices como vectores cambian de dimensión de iteración en iteración. Esto viene de la necesidad de capturar el efecto dinámico provocado por la operación real, en donde paradas ya visitadas no se requieren para tomar decisiones a futuro (si pueden ser ocupadas como data histórica). Es así que se desarrolló un mecanismo de expansión y reducción de matrices (mecanismo adaptativo). La idea es disminuir o incrementar el vector de posiciones mostrado en la ecuación (34), reduciendo o incrementando también los vectores de carga y tiempo estimados. Por ejemplo, cuando cierto vehículo acepta un nuevo requerimiento por servicio, las dimensiones del vector de posiciones aumentan en dos filas, incorporando la parada para recoger y para dejar al nuevo pasajero. Adicionalmente, cuando un vehículo alcanza cualquier parada, ese punto tiene que ser removido del vector de posición inicial, reduciendo su dimensión en una columna.

#### **4.3.2 Función Objetivo y Horizontes de predicción**

En la sección anterior, se desarrolló la formulación analítica del modelo en variables de estado. A continuación se presenta el concepto de función de costos para incorporar una medición del comportamiento del sistema. El objetivo del controlador predictivo híbrido será minimizar esta función de costos con lo cual se seleccionarán las mejores rutas de acuerdo a esta minimización. La función de costos escogida para realizar las pruebas por simulación incorpora tanto el tiempo de espera como el tiempo de viaje experimentado por cada pasajero que solicita servicio. Además, se agrega un costo por servicio equivalente al tiempo total de viaje que recorre cada vehículo, con lo cual se incluye el tiempo de los trayectos en que el vehículo viaja sin pasajeros a bordo en la decisión.

Lo más importante en la definición de la función objetivo será el definir un horizonte de predicción razonable  $N$ , el cual depende del problema en estudio y de la intensidad de eventos desconocidos que ocurran en el sistema en tiempo real. En los casos en que la

decisión tomada en el instante  $k$  considera un horizonte de predicción mayor que uno, el controlador le agrega a la decisión la característica predictiva. Esto, en la medida que la predicción sea buena, provocará que el sistema funcione mejor dinámicamente ya que la decisión que se toma en el instante actual afectará a todos los eventos que se generen a futuro y por consiguiente se reducirá de mejor manera la función de costos. Tomando en cuenta lo anterior, el controlador (o central despachadora) calculará las decisiones para el horizonte de control completo  $N$ , es decir,  $\{S(k), \dots, S(k+N-1)\}$ , mediante predicciones basadas en datos históricos, y aplicará al sistema solamente la secuencia en el siguiente instante  $S(k)$ , de acuerdo al funcionamiento del horizonte deslizante. El funcionamiento del ruteo de los vehículos, dependerá entonces de que tan bien el sistema predice el impacto de posibles re-ruteos debido a inserciones provocadas por requerimientos por servicio futuros y desconocidos.

La función objetivo para un horizonte de predicción genérico  $N$ , se escribe de la siguiente manera:

$$\underset{\{S(k), S(k+1), \dots, S(k+N-1)\}}{\text{Min}} J = \sum_{t=1}^N \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h^{\Delta T(k+t)} \cdot \left( (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2), h} \right) \quad (40)$$

$$C_j(k+t) \Big|_{S_j(k+t-2), h} = \sum_{i=1}^{w_j(k+t)} \left\{ \underbrace{\left[ \hat{L}_j^{i-1}(k+t) + 1 \right] \left( \hat{T}_j^i(k+t) - \hat{T}_j^{i-1}(k+t) \right)}_{\text{J travel time}} + \underbrace{z_j^i(k+t) \alpha \left( \hat{T}_j^i(k+t) - T_j^0(k+t) \right)}_{\text{J waiting time}} \right\} \Big|_{S_j(k+t-2), h} \quad (41)$$

donde  $k+t$  es el instante en el que el  $t$ -ésimo requerimiento entra en el sistema, medido desde el requerimiento del instante  $k$ .  $h_{\max}(k+t)$  es el número de posibles requerimientos en el instante  $k+t$ ,  $p_h^{\Delta T(k+t)}$  es la probabilidad de ocurrencia del  $h$ -ésimo requerimiento tipo (asociado a un par específico de zonas, como se discutirá mas adelante) durante el intervalo de tiempo  $\Delta T(k+t)$ , notando que  $\Delta T(k+t)$  especifica un cierto instante de tiempo en el cual el paso  $k+t$  pertenece.  $C_j(k+t) \Big|_{S_j(k+t-2), h}$  en (41) es la función de costos del vehículo  $j$  en el instante  $k+t$ , el cual depende de la secuencia previa en el instante

$k+t-1$   $S_j(k+t-2)$  y un nuevo requerimiento potencial  $h$  el cual ocurre con probabilidad  $p_h^{\Delta T(k+t)}$ ,  $w_j(k+t)$  es el número de paradas estimadas para el vehículo  $j$  en el instante  $k+t$ ,  $S_j(k+t-2), h$  es la nueva secuencia dado que  $h$  ocurrió. La función de costos  $C_j(k+t)$  en (41), contiene dos componentes: tiempo de espera y tiempo de viaje. Ambas se escriben en función de la carga y el tiempo de salida y se calculan entre paradas consecutivas. Notar que se considera un muestreo con tiempo variable dado por el tiempo entre dos requerimientos consecutivos. Este paso variable fue sintonizado usando análisis de sensibilidad. Para explicitar la flexibilidad de la formulación y la consistencia económica, el costo de espera se ponderó por un peso  $\alpha$ , y un factor para el tiempo de espera en (41) como sigue:

$$z_j^i(k+t) = \begin{cases} 1 & \text{si la parada } i \text{ en el instante } k+t \text{ es una recogida} \\ 0 & \text{En otro caso} \end{cases} \quad (42)$$

Además, la expresión (41) depende de la matriz secuencia  $S(k)$ , y se escribe en términos de la matriz  $G$  y sus componentes según muestra (43).

$$C_j(k+t) \Big|_{S_j(k+t-2), h} = \sum_{i=1}^{w_j(k+t)} \left[ L_j^0(k) + 1 + \sum_{m=1}^{i-1} \left( \sum_{k_p: f_{k_p} \text{ is a pickup}} g_{mk_p} - \sum_{k_d: f_{k_d} \text{ is a delivery}} g_{mk_d} \right) \right] \left[ G^{(i-1)} Q (G^i)^T \right] + \alpha \cdot z_j^i(k+t) \left( \sum_{r=1}^{i-1} G^r Q (G^{r+1})^T \right) \quad (43)$$

Las probabilidades de ocurrencia asociadas a cada escenario son parámetros en la función objetivo y deben ser calculadas en base a data en tiempo real, data histórica o una combinación de ambas. En esta aplicación en particular, se propone como método novedoso para el cálculo de estas probabilidades la zonificación difusa, método con el cual sistematizamos el manejo de la data histórica para la detección de patrones de viaje (implementación “off line”).

La estrategia a un paso significa que el horizonte de predicción es  $N = 1$ , y  $h_{\max}(k+1) = 1$  debido a que el nuevo requerimiento es sólo uno y es conocido, y entonces tiene

probabilidad igual a 1, obteniéndose la siguiente expresión para la función objetivo, usando (40):

$$\begin{aligned}
J &= \sum_{t=1}^1 \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h(k+t) \cdot (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2),h} \\
&= \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+1)=1} p_h(k+1) \cdot (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),h} \\
&= \sum_{j=1}^F \overbrace{p_1(k+1)}^{=1} \cdot (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1)} = \\
J &= \sum_{j=1}^F \left( C_j(k+1) - \overbrace{C_j(k)}^{\text{constante conocida}} \right) \Big|_{S_j(k-1)} \tag{44}
\end{aligned}$$

La diferencia  $(C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1}$  significa que el costo se evalúa considerando la acción de control en el instante anterior, representado por  $S_j(k-1)$ . Conceptualmente,  $J$  representa el costo de inserción cuando el sistema acepta una nueva llamada, la cual se genera aleatoriamente en tiempo real, y considerando toda la flota de vehículos. Notar que hay muchas posibles alternativas de inserción del nuevo requerimiento y será función del controlador determinar cual es la mejor resolviendo el problema de minimización de la función objetivo.

La función objetivo a dos pasos es distinta a la anterior ya que ahora no se conoce con certeza la ubicación de la llamada que ocurrirá a dos pasos. Se asume en la formulación que la decisión que se presente dependerá de los requerimientos potenciales futuros. Ocupando un enfoque probabilístico, se incorpora los requerimientos más probables a ocurrir a dos pasos en la función objetivo, según el periodo del día en que se realice la incorporación de la nueva llamada.

El controlador predictivo híbrido con criterio de optimización a dos pasos selecciona las secuencias que minimizan la función objetivo a dos pasos. Esta función objetivo representa el costo ocasionado al sistema por posibles inserciones a futuro cuando un

nuevo requerimiento aparezca en las cercanías de la trayectoria de algún vehículo en un intervalo de tiempo específico y dado por una probabilidad  $p_h^{\Delta T(k+2)}$ .

La expresión correspondiente a la función objetivo a dos pasos se muestra en (45):

$$\begin{aligned}
 J &= \sum_{t=1}^2 \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h(k+t) \cdot (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2),h} = \\
 &= \sum_{j=1}^F \left[ C_j(k+1) \Big|_{S_j(k-1),1} - C(k) + \sum_{h=1}^{h_{\max}(k+2)} p_h(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \sum_{h=1}^{h_{\max}(k+2)} p_h(k+2) \cdot C_j(k+1) \Big|_{S_j(k-1),1} \right] = \\
 &= \sum_{j=1}^F \left[ \sum_{h=1}^{h_{\max}(k+2)} p_h(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \overbrace{C(k)}^{\text{constante conocida}} \right] \\
 J &= \sum_{j=1}^F \left[ \sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \overbrace{C(k)}^{\text{constante conocida}} \right] \quad (45)
 \end{aligned}$$

Otro caso interesante es el de la función objetivo a tres pasos, nuevamente a partir de la expresión (40), es:

$$J = \sum_{j=1}^F \left[ \sum_{h_2=1}^{h_{\max}(k+2)} p_{h_2}(k+2) \cdot \left( \sum_{h_3=1}^{h_{\max}(k+3)} p_{h_3}(k+3) \cdot C_j(k+3) \Big|_{S_j(k),h_2,h_3} \right) - \overbrace{C(k)}^{\text{known constant}} \right] \quad (46)$$

Para efectos ilustrativos de la metodología propuesta como se muestra en la Figura 29, se analiza el caso de predicción a tres pasos (con *no swapping*) para un ejemplo de dos pares origen destino probables a dos pasos, y cuatro posibles a tres pasos, en los cuales la estrategia debe ser evaluada en la siguiente cadena de escenarios.

En el instante  $k-1$ , los vehículos tienen una cierta secuencia establecida  $S(k-1)$  asociada a un costo total  $C(k)$ . Cuando un nuevo requerimiento entra al sistema, hay muchas secuencias factibles  $S(k)$  que deben ser evaluadas por el controlador (cada alternativa inserta la nueva recogida y entrega en un segmento factible de un vehículo específico). A un paso, se considera el nuevo requerimiento (instante  $k$  con probabilidad igual a 1). A

dos pasos, se fijan dos requerimientos potenciales en el siguiente instante  $k+1$ , con probabilidades  $p_1(k+2)$  y  $p_2(k+2)$  respectivamente. A tres pasos, se considera que aparecen cuatro requerimientos potenciales en el instante  $k+2$ , con probabilidades  $p_1(k+3)$ ,  $p_2(k+3)$ ,  $p_3(k+3)$  y  $p_4(k+3)$  respectivamente, incorporando así la naturaleza dinámica del problema, y consecuentemente para tener mejores estimaciones tanto del tiempo de viaje como del de espera en la función objetivo. Finalmente, ocho secuencias potenciales de requerimientos son evaluadas, conteniendo tres nuevas inserciones en cada secuencia (la llamada conocida que llega a un paso, y las llamadas potenciales que aparecen a dos y tres pasos).

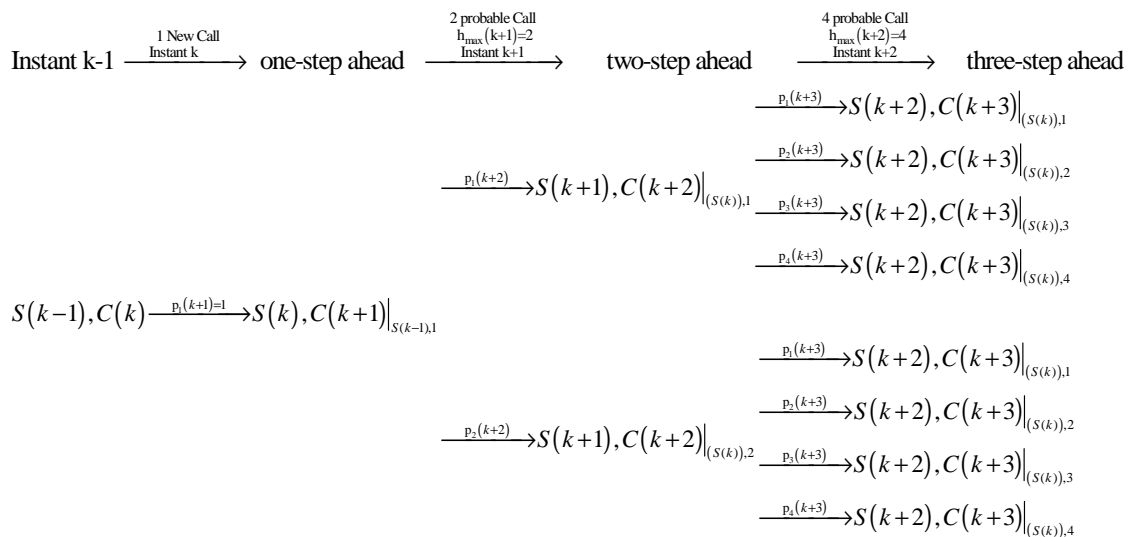


FIGURA 29. LLAMADAS POTENCIALES.

Para obtener buenas estimaciones de los posibles escenarios futuros que deben ser ocupados en la función objetivo, se analizan los datos históricos a través de una metodología sistemática que permite determinar los patrones de viaje y sus correspondientes probabilidades de ocurrencia. En la siguiente sección, un enfoque que ocupa clustering difuso es utilizado para tratar este problema.

### 4.3.3 Zonificación difusa

En esta sección se desarrolla una metodología sistemática para dividir el área urbana en regiones conceptuales para una mejor representación de los patrones de demanda histórica. Estos pueden ser obtenidos de la información de demanda asociada a un día de operación representativo. Esta propuesta resulta ser una alternativa al enfoque de zonificación propuesto por Cortés et al. (2005), donde el área total es dividida en zonas homogéneas y no sobrepuestas. Este enfoque puede perjudicar al funcionamiento del controlador en casos en donde los patrones de viaje típicos no coincidan con los pares de zonas predefinidas de acuerdo al método convencional. De hecho, una zonificación equivocada puede impactar negativamente sobre el cálculo de las probabilidades en la función objetivo para predicciones a más de dos pasos. La zonificación sistemática propuesta a continuación está basada en el método de fuzzy clustering el cual permite clasificar las típicas llamadas origen destino en clusters representativos de la data histórica y flexibles. Por simplicidad y considerando las características del problema, se adopta fuzzy C-Mean para hacer la clasificación espacial de la demanda.

#### Fuzzy DPDP para cálculo de las probabilidades

El método fuzzy C-means (FCM) propuesto por Bezdek (1973) es una técnica de clasificación de datos donde cada punto de la data pertenece a un cluster con un único grado de pertenencia. En otras palabras, FCM divide el espacio en un número específico de clusters representativos. FCM considera una partición difusa, tal que un punto de la data en el espacio pueda pertenecer a más de un cluster, pero con distintos grados de pertenencia (que varía de 0 a 1). FCM es un algoritmo iterativo que permite encontrar centros de cluster (centroides) que minimizan la siguiente función objetivo:

$$S(c) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^m \|x_k - v_i\|^2 \quad (47)$$

Donde  $n$  es el número del dato,  $c$  es el número de clusters,  $\mu_{ik}$  es la partición difusa entre 0 y 1,  $v_i$  representa al centro del cluster  $i$  y  $m \in [1, \infty]$  es un factor de peso. Los detalles del



algoritmo fuzzy C-means se encuentran por ejemplo en Babuska (1999). En esta aplicación, el método FCM es usado para determinar centros representativos asociados con patrones históricos de demanda origen-destino, los cuales permitirán calcular las correspondientes probabilidades de ocurrencia.

Se propone el cálculo de las probabilidades de cada cluster asociado a un par origen-destino, mediante el siguiente procedimiento:

- Paso 1: Se obtienen los fuzzy clusters a partir de la demanda histórica mediante el método FCM.
- Paso 2: Se calcula el grado de pertenencia asociada a cada llamada histórica mediante el procedimiento FCM del Paso 1.
- Paso 3: Se asocia cada llamada a un solo cluster, correspondiente a aquel en que la llamada posea el mayor grado de pertenencia.
- Paso 4: Llamadas con un grado de pertenencia inferior a un *“threshold”* predefinido no son consideradas en el proceso de cálculo de probabilidades.
- Paso 5: Se calcula una probabilidad de ocurrencia de un nuevo requerimiento cercano a un par origen-destino específico como el número de llamadas que pertenecen al cluster difuso dividido por el número total de llamadas (luego de remover la data que no usará en el Paso 4).
- Paso 6: Se realiza un nuevo cálculo de los centros de cluster con la demanda histórica sin los datos removidos en el paso 4.

En resumen, el método FCM permite obtener patrones origen-destino más realista a partir de la data histórica, y como consecuencia, se sistematiza y mejora del proceso de zonificación y cálculo de probabilidades. Este procedimiento podría mejorar el poder predictivo de la incerteza que resulta debido a la demanda futura desconocida, en estrategias de control mayores a dos pasos.

Por ejemplo, FCM logra reconocer muy bien patrones de viaje que se traslapan, en donde las zonas representativas se intersectan. A continuación, un ejemplo en una dimensión se muestra para ilustrar la aplicación del método en el contexto del DPDP.

### Ejemplo ilustrativo del método FCM

Un ejemplo simple para el problema dinámico de rutear un vehículo es mostrado en la Figura 30 para clarificar la aplicación de FCM en la zonificación de la demanda. Asuma requerimientos puerta a puerta ocurriendo una sola vía en la cual el vehículo puede ir en ambas direcciones. La vía es de 9 kilómetros y los requerimientos de recogida y entrega ocurren dentro de ese rango. En el ejemplo, suponga que ocurren las diez llamadas durante un cierto periodo de tiempo (Figura 30), y suponga que esa información es usada para determinar la zonificación óptima y las correspondientes probabilidades asociadas a dicha partición.

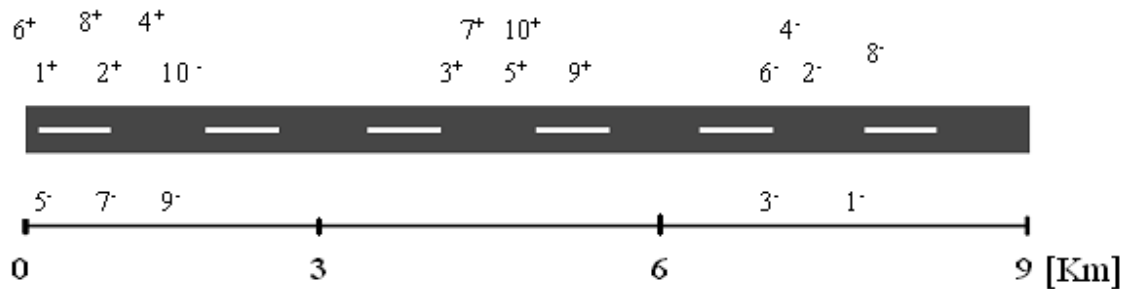


FIGURA 30. REQUERIMIENTO PARA UN VEHICULO EN UN CIERTO PERIODO DE TIEMPO.

La Figura 31 muestra una representación en dos dimensiones para la coordenada de la recogida y de la entrega, para los requerimientos mostrados en la Figura 30. Analizando la Figura 31, los patrones de viaje pueden ser identificados con tan solo observar los puntos que se encuentran más cercanos en ellos, debido a que el problema se definió como viajes en una dimensión. Sin embargo, cuando el problema se define con requerimientos en dos dimensiones, el análisis requiere una metodología automática como la FCM propuesta.

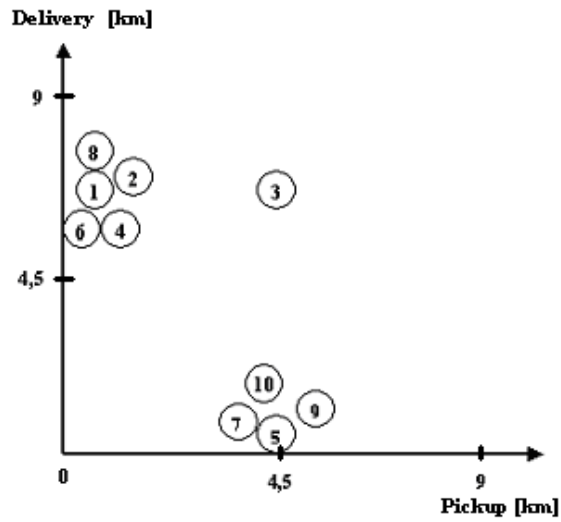


FIGURA 31. COORDENADAS PICKUP-DELIVERY DE LA DEMANDA HISTORICA EN UN CIERTO PERIODO

A partir de la data histórica de la Figura 30 (o Figura 31), se ocupa FCM para obtener la zonificación más representativa del problema asociada a dicha base de datos. Para hacer esto, se selecciona un número fijo de clusters tal y como muestra la Figura 32 en donde se seleccionan 2 y 3 clusters, respectivamente. Los centros de los clusters son calculados y se denotan con una marca "x" en la figura.

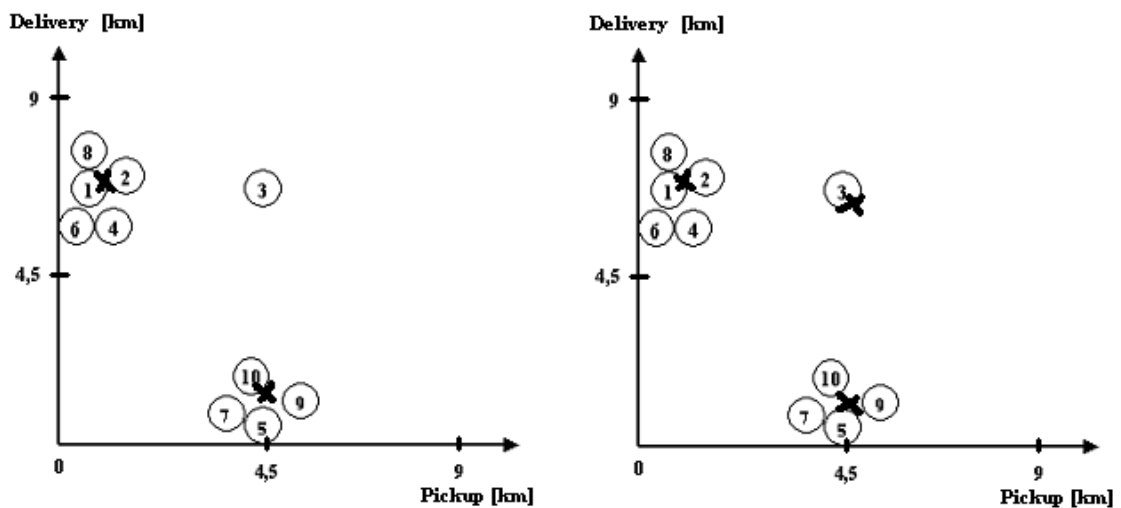


FIGURA 32. CENTROS DE CLUSTER PARA UN NÚMERO DE 2 Y 3 CLUSTERS.

Entonces, el centro de masa se obtiene luego de aplicar el método FCM correspondiente para obtener los patrones de viaje, para este ejemplo particular. De un análisis de la Figura 32, parece ser razonable ocupar 2 clusters en lugar de 4, debido a que la mayoría de los requerimientos están bien agrupados alrededor de los dos centros de masa. En general, definir el número adecuado de clusters no es tan fácil como en este ejemplo, y en algunos casos, el diseñador de la estrategia debería usar alguna otra metodología como por ejemplo, el método de clustering merging difuso (Babuska, 1999).

La Figura 33 muestra los grados de activación de cada requerimiento de los 10 usados, con FCM usando 2 clusters. Como se muestra en la Figura 33, la selección de un "threshold" determina que la llamada 3 no pertenece a ninguno de los dos clusters, y entonces ese dato no debe ser considerado en las estadísticas ya que se sale de la desviación.

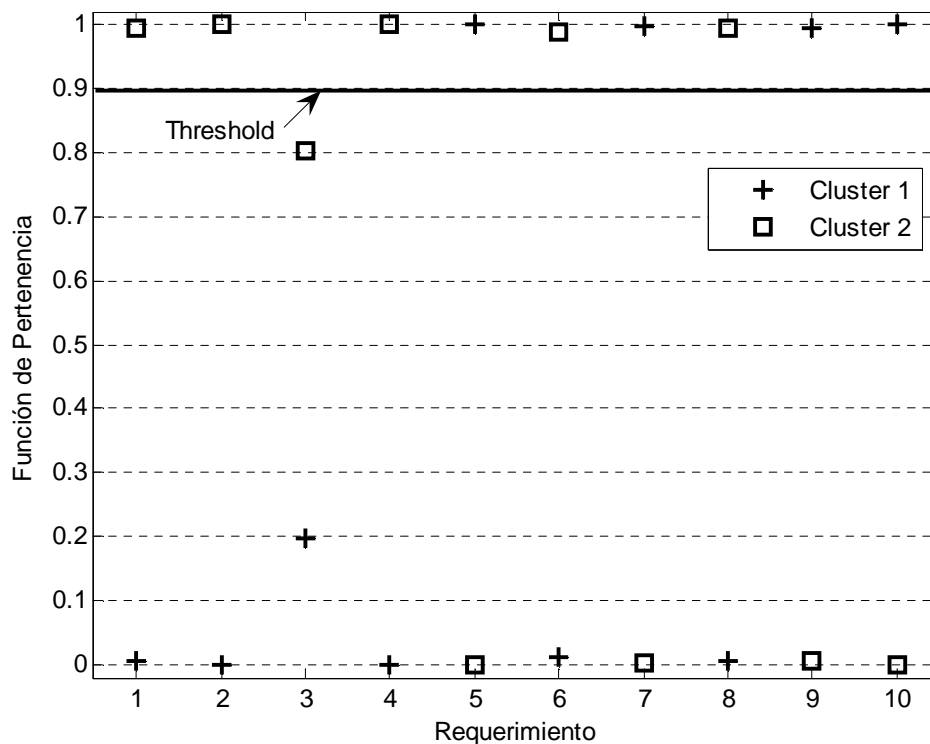


FIGURA 33. GRADO DE PERTENENCIA A CADA CLUSTER DE LA DATA HISTORICA.

Finalmente, y usando el procedimiento FCM, las probabilidades asociadas a cada patrón de viaje se determinan y son las mostradas en la Tabla 7, para ambos clusters.

TABLA 7. PROBABILIDADES DE CADA PATRON DE VIAJE USANDO DOS CLUSTERS DIFUSOS.

Patrón de viaje	Posición recogida	Posición entrega	Probabilidad
Fuzzy cluster 1	0.7194	6.9800	4/9
Fuzzy cluster 2	4.4748	0.2750	5/9

Esta metodología FCM propuesta se aplicó a un problema por simulación más complejo de DPDP, y se comparó con la zonificación clásica usada por Cortés et al (2005), como se muestra en el capítulo de resultados.

#### 4.3.4 Análisis del espacio de búsqueda de soluciones y restricciones

En esta aplicación, la optimización se realiza en un espacio de solución reducido que corresponde a las secuencias que satisfacen la restricción *no-swapping*. Este criterio provee secuencias que localizan la entrega y recogida de la última llamada entrante dentro de la secuencia anterior (el orden de las paradas ya ruteadas no cambia).

El óptimo global del problema de ruteo dinámico en términos de una variable de optimización, llamada matriz  $G$ , se obtiene escogiendo adecuadamente sus factores  $g_{ir}$ , para cada vehículo en la flota. Entonces,  $G^*$  determina la secuencia óptima de paradas  $P_j(k)$  de cada vehículo  $j$  que minimiza la función objetivo definida en la sección anterior, cuando un nuevo requerimiento en tiempo real es insertado en alguna de las secuencias previas. Explícitamente, el vector óptimo  $P_j(k)$  está dado por:

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11}^* & g_{12}^* & \cdots & \cdots & g_{1(n-1)}^* & g_{1n}^* \\ g_{21}^* & g_{22}^* & \cdots & \cdots & g_{2(n-1)}^* & g_{2n}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1}^* & g_{(n-1)2}^* & \cdots & \cdots & g_{(n-1)(n-1)}^* & g_{(n-1)n}^* \\ g_{n1}^* & g_{n2}^* & \cdots & \cdots & g_{n(n-1)}^* & g_{nn}^* \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G^* \cdot f \quad (48)$$

donde  $f$  es un vector que contiene la lista de las paradas que deben ser recorridas en el sistema en el instante  $k$ . En el caso *no-swapping*, las nuevas llamadas se insertan directamente en la secuencia anterior asignada, manteniendo el orden (sólo se permite inserciones dentro de los segmentos anteriores). Como la secuencia anterior se mantiene, por ejemplo  $(f_1, f_2, \dots, f_{n-2})$ , la nueva llamada es agregada al vector  $f$  como las últimas dos componentes (la recogida y la entrega), denotándolas por  $(f_{n-1}, f_n)$  y se imponen las siguientes condiciones (49) en relación a los factores de la matriz  $G$  en (48).

$$P_i(k) = \begin{cases} g_{11}f_1 + g_{1,n-1}f_{n-1} = (x_1, y_1) & \text{if } i=1 \\ g_{21}f_1 + g_{22}f_2 + g_{2,n-1}f_{n-1} + g_{2n}f_n = (x_2, y_2) & \text{if } i=2 \\ g_{i,i-2}f_{i-2} + g_{i,i-1}f_{i-1} + g_{i,i}f_i + g_{i,n-1}f_{n-1} + g_{i,n}f_n = (x_i, y_i) & \text{if } i=3, \dots, (n-2) \\ g_{n-1,n-3}f_{n-3} + g_{n-1,n-2}f_{n-2} + g_{n-1,n-1}f_{n-1} + g_{n-1,n}f_n = (x_{n-1}, y_{n-1}) & \text{if } i=n-1 \\ g_{n,n-2}f_{n-2} + g_{n,n}f_n = (x_n, y_n) & \text{if } i=n \end{cases} \quad (49)$$

donde  $(x_i, y_i)$  son las coordenadas espaciales de la parada  $i$ -ésima. Por ejemplo, el primer término de (49) ( $i=1$ ) representa a la primera componente de la secuencia de paradas que el vehículo debe realizar, y puede ser ya sea la recogida de la nueva parada o bien la primera parada de la secuencia previa. El segundo término ( $i=2$ ) representa la segunda componente de la secuencia de paradas que tiene mas opciones, ya sea la primera parada de la secuencia anterior, la segunda parada de la secuencia anterior, la recogida de la nueva llamada o bien el delivery de la nueva llamada.

Las ecuaciones (49) también se pueden escribir en la forma general (48), obteniendo la siguiente matriz  $G$  de incógnitas:

$$G = \begin{bmatrix} g_{11} & 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{1(n-1)} & 0 \\ g_{21} & g_{22} & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{2(n-1)} & g_{2n} \\ g_{31} & g_{32} & g_{33} & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{3(n-1)} & g_{3n} \\ 0 & g_{42} & g_{43} & g_{44} & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{4(n-1)} & g_{4n} \\ 0 & 0 & g_{53} & g_{54} & g_{55} & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{5(n-1)} & g_{5n} \\ 0 & 0 & 0 & g_{64} & g_{65} & g_{66} & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{6(n-1)} & g_{6n} \\ \vdots & \vdots & \vdots & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & g_{(n-4)(n-6)} & g_{(n-4)(n-5)} & g_{(n-4)(n-4)} & 0 & 0 & g_{(n-4)(n-1)} & g_{(n-4)n} \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & 0 & g_{(n-3)(n-5)} & g_{(n-3)(n-4)} & g_{(n-3)(n-3)} & 0 & g_{(n-3)(n-1)} & g_{(n-3)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & g_{(n-2)(n-4)} & g_{(n-2)(n-3)} & g_{(n-2)(n-2)} & g_{(n-2)(n-1)} & g_{(n-2)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & 0 & g_{(n-1)(n-3)} & g_{(n-1)(n-2)} & g_{(n-1)(n-1)} & g_{(n-1)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & 0 & 0 & g_{n(n-2)} & 0 & g_n \end{bmatrix}$$

FIGURA 34. MATRIZ G CASO NO SWAPPING.

Esta formulación analítica permite generalizar el criterio de optimización a N pasos en la siguiente sección y evaluar distintos métodos de optimización entera mixta no lineales, como GA. Si la restricción *no-swapping* es relajada, el espacio de búsqueda aumenta, resultado en una matriz G con menos ceros, y permitiría indagar en más soluciones dentro de las cuales se puede encontrar el óptimo global (problema menos restrictivo). Un caso de solución intermedia (*partial swapping*) puede ser también sugerido, como solución intermedia entre *swapping* y búsqueda completa.

Una vez que las variables de estado se definen analíticamente en la sección 4.3.1, y las condiciones en el espacio de búsqueda son planteadas (sección 4.3.4), se minimiza a cada instante la función objetivo (sección 4.3.2 con zonificación difusa explicada en 4.3.3) cumpliendo todas las restricciones impuestas.

#### 4.4 Algoritmos de Optimización

Actualmente, con el creciente interés de la comunidad de Control Automático en resolver problemas de control de Sistemas Híbridos, las limitantes en cuanto al tiempo computacional que se requiere para resolver dichos problemas han aparecido.

Los métodos de optimización más usados involucran dos algoritmos: Enumeración Explícita (EE) y Branch and Bound (BB). Ambos permiten resolver problemas de optimización entera mixta no lineal (Floudas, 1995), pero el elevado tiempo computacional requerido, especialmente en el caso de EE, resulta en que las soluciones son ineficientes para problemas en tiempo real. En ese sentido, los algoritmos genéticos (GA) aparecen como herramienta eficiente para resolver MIOP (Problemas de optimización entera mixta) (Man et al., 1998). Entonces, como VRP es un problema NP hard, se considera resolver el problema de control predictivo híbrido basado en optimización GA, para enfrentar el problema DPDP que se explicó en la sección anterior.

#### **4.4.1 Optimización basada en Algoritmos Genéticos (GA)**

El método que entrega GA se puede implementar fácilmente en el problema DPDP debido a que las variables de optimización son enteras y entonces la codificación binaria no es necesaria. En otras palabras, los genes de los individuos (soluciones factibles) están dados directamente por la optimización de las variables enteras. Además, debido a que GA no ocupa el gradiente de la función objetivo, se ahorra un elevado tiempo computacional en la resolución del problema no lineal.

El algoritmo de optimización GA, se resume en los siguientes pasos (Man et al. 1998):

1. Inicializar una población aleatoria de individuos.
2. Evaluar la función objetivo de cada individuo de la población actual.
3. Seleccionar padres aleatorios de la población actual.
4. Aplicar operadores genéticos como *crossover* y *mutation* a los padres, para producir una nueva generación.
5. Evaluar la función objetivo de todos los nuevos individuos de la generación.
6. Escoger a los mejores individuos según los mejores valores de la función objetivo.
7. Reemplazar los individuos más débiles de la generación anterior por los mejores obtenidos del paso 6.
8. Si el valor de la mejor función objetivo convergió o bien se alcanzó el número máximo de generaciones, entonces el algoritmo se detiene. En otro caso, ir al Paso 2.



En resumen, GA entrega soluciones cercanas al óptimo global. Los parámetros de sintonización son número de individuos, número de generaciones, probabilidad de crossover, probabilidad de mutación y el criterio de parada.

#### 4.4.2 Aplicación de GA para el problema de ruteo dinámico

Se propone GA como algoritmo eficiente de optimización para el problema DPDP, en donde las variables de optimización son el orden en que se satisfacen los requerimientos sujetos a las restricciones de operación. Los individuos corresponderán a secuencias factibles, que cumplen las restricciones de carga, precedencia y no swapping. Un individuo posee tres genes, el vehículo  $j$  en el cual se insertará la nueva llamada y la posición dentro de la secuencia anterior de la nueva llamada (tanto para la recogida como entrega), asumiendo la política *no-swapping*.

Para explicar la codificación, a continuación se presenta un ejemplo sencillo. Considere el siguiente vector de paradas para el vehículo  $j$  en el instante  $k-1$ ,  $P_j(k-1)$ , asociado a la secuencia en el instante previo  $k-1$  ( $S_j(k-1)$ ):

$$P_j(k-1) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \end{bmatrix}}_f \quad (50)$$

donde  $b(x)$  es la posición de la parada  $x$ . En este ejemplo, entra al sistema un nuevo requerimiento con etiqueta 3, y debe ser insertado en algún vehículo, dentro de su secuencia. La nueva variable de optimización puede ser representada en términos de  $P_j(k)$  como muestra el siguiente sistema, agregando el nuevo requerimiento en las últimas filas del vector  $f$ , aumentando la dimensión de la matriz  $G$ .

$$P_j(k) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \\ P_j^5 \\ P_j^6 \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11} & 0 & 0 & 0 & g_{15} & 0 \\ g_{21} & g_{22} & 0 & 0 & g_{25} & g_{26} \\ g_{31} & g_{23} & g_{33} & 0 & g_{35} & g_{36} \\ 0 & g_{24} & g_{34} & g_{36} & g_{45} & g_{46} \\ 0 & 0 & g_{35} & g_{37} & g_{55} & g_{56} \\ 0 & 0 & 0 & g_{38} & 0 & g_{66} \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \\ b(3^+) \\ b(3^-) \end{bmatrix}}_f \quad (51)$$

Debido a las restricciones de precedencia y de no swapping, la secuencia se mantiene, y las variables de decisión están dadas por la ubicación de las nuevas paradas dentro de la matriz G. Usando la codificación propuesta, se muestra a continuación una población factible de 7 individuos para el vehículo  $j$ , considerando la secuencia previa y la nueva llamada:

$$\text{Population} \Leftrightarrow \begin{pmatrix} \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \\ \text{Individual 4} \\ \text{Individual 5} \\ \text{Individual 6} \\ \text{Individual 7} \end{pmatrix} \Leftrightarrow \begin{pmatrix} (j,1,4) \\ (j,1,6) \\ (j,5,6) \\ (j,3,5) \\ (j,4,6) \\ (j,1,6) \\ (j,2,4) \end{pmatrix} \Leftrightarrow \begin{pmatrix} j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow \boxed{3^-} \rightarrow 1^- \rightarrow 2^- \\ j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^+} \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow 2^+ \rightarrow \boxed{3^+} \rightarrow 1^- \rightarrow \boxed{3^-} \rightarrow 2^- \\ j, 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow \boxed{3^+} \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow \boxed{3^+} \rightarrow 2^+ \rightarrow \boxed{3^-} \rightarrow 1^- \rightarrow 2^- \end{pmatrix} \quad (52)$$

Por ejemplo, el individuo  $(j,1,4)$  en términos de  $P_j(k)$  puede ser escrito como:

$$\text{Individual 1} \Leftrightarrow P_j(k) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \\ P_j^5 \\ P_j^6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \\ b(3^+) \\ b(3^-) \end{bmatrix}}_f$$

En resumen, las últimas dos columnas de la matriz  $G$  son las nuevas variables de optimización (indican posición de recogida y entrega de la nueva llamada) en el instante  $k$ . Como los individuos son generados al azar, un mismo individuo puede repetirse en la misma población. Por ejemplo, el individuo 2 y 6 es el mismo en (52),  $(j, 1, 6)$ .

Notar que como GA considera generaciones aleatorias de individuos, los operadores genéticos mutación y crossover podrían generar soluciones infactibles que deben ser penalizados de manera de que el algoritmo detecte su infactibilidad, o bien eliminado de la población, o bien reparado según alguna heurística. Esta infactibilidad típicamente aparece por la restricción de capacidad. En esta tesis, el mecanismo usado con individuos no factibles es el de eliminación, por lo que es necesario que el algoritmo comience con al menos un individuo factible, como por ejemplo el  $(j, w_j - 1, w_j)$  ( $w_j$  es el número de paradas incluyendo la última llamada). El número de individuos por el número de generaciones debe ser más pequeño que el número total de combinaciones factibles con el fin de que no se enumeren todas las soluciones factibles como lo hace el algoritmo EE. El operador crossover no será aplicado en este caso debido a que la restricción no swapping debe ser satisfecha.

La Figura 35 muestra el esquema de control predictivo híbrido propuesto. Al sistema real de flota y clientes se le asignan las secuencias usando el controlador HAPC basado en predicciones que usan el modelo en variable de estado, predictor de llamadas y los detalles del nuevo requerimiento.

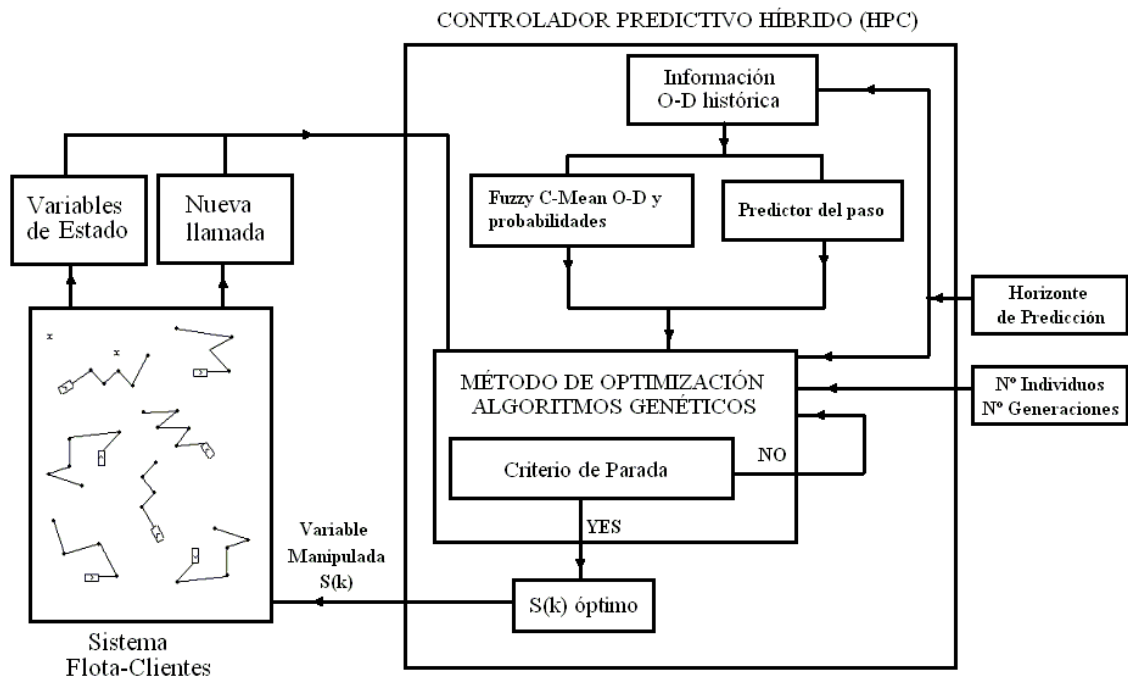


FIGURA 35. DIAGRAMA DE BLOQUES COMPLETO DE HAPC PARA EL DPDP.

A continuación se presenta un ejemplo de aplicación de GA en el contexto de DPDP, para visualizar las ventajas del método cuando es comparado con enumeración explícita, tanto en ahorro computacional como en calidad de solución.

#### 4.4.3 Ejemplo por simulación

En esta sección, se realizan pruebas ilustrativas usando los métodos enumeración explícita (EE) y GA para evaluar el desempeño de los controladores mediante la evaluación de la función objetivo y los correspondientes tiempos computacionales.

Un DPDP con 4 vehículos en la flota y función objetivo a dos pasos con 6 patrones de llamadas es considerado. Los vehículos cubren un área urbana de servicio de 81 km<sup>2</sup>, viajando a una velocidad promedio de 20 kilómetros por hora (Cortés *et al.* 2005).

Las pruebas por simulación consideran:

- i) Ruteo dinámico de vehículos bajo condiciones de alta demanda.
- ii) Ruteo dinámico de vehículos bajo condiciones de demanda normal y
- iii) Ruteo dinámico de vehículos bajo condiciones de alta demanda considerando como secuencia previa la entregada por EE (combinando GA y EE).

Como se mencionó antes, el método GA considera el número de individuos y generaciones, y probabilidad de mutación como parámetros de sintonía. Resultados para tres casos de sintonizaciones de parámetros son presentados. El primero considera la solución por GA usando 5 individuos y 5 generaciones, G2 usa 10 individuos y 10 generaciones, y finalmente G- considera 20 individuos y 20 generaciones.

Las pruebas por simulación se realizaron en Matlab versión 6.5.1 release 13, en un procesador Pentium IV.

### **Prueba 1: Ruteo dinámico de vehículos bajo condiciones de alta demanda.**

En este caso, muchos requerimientos entran al sistema en un periodo corto de tiempo, generando largas secuencias y tiempos computacionales debido al incremento en el espacio de búsqueda. La Figura 36 muestra el tiempo computacional y la función objetivo para un cierto periodo en el cual muchas llamadas entraron al sistema (recuerde que el paso en el modelo es variable, y depende de cuando el despachador recibió la nueva llamada).

De la Figura 36, se observa una congestión de requerimientos, y GA presenta un costo acumulativo en cada nueva llamada debido a que la decisión tomada en el instante previo (secuencia previa) no siempre correspondía a la óptima global. Pero por otro lado, el tiempo computacional crece exponencialmente usando EE mientras que el número de paradas aumenta, a diferencia de GA que muestra tiempos computacionales estables en relación a la intensidad de llamadas.

En la Tabla 8, el valor medio de la función objetivo y tiempo computacional son reportados usando la información de la Figura 36. Según la Figura 36 y la Tabla 8, cuando el número

de individuos y de generaciones aumenta, un mejor seguimiento del óptimo global de la función objetivo es observado (en especial G3) con un tiempo computacional reducido.

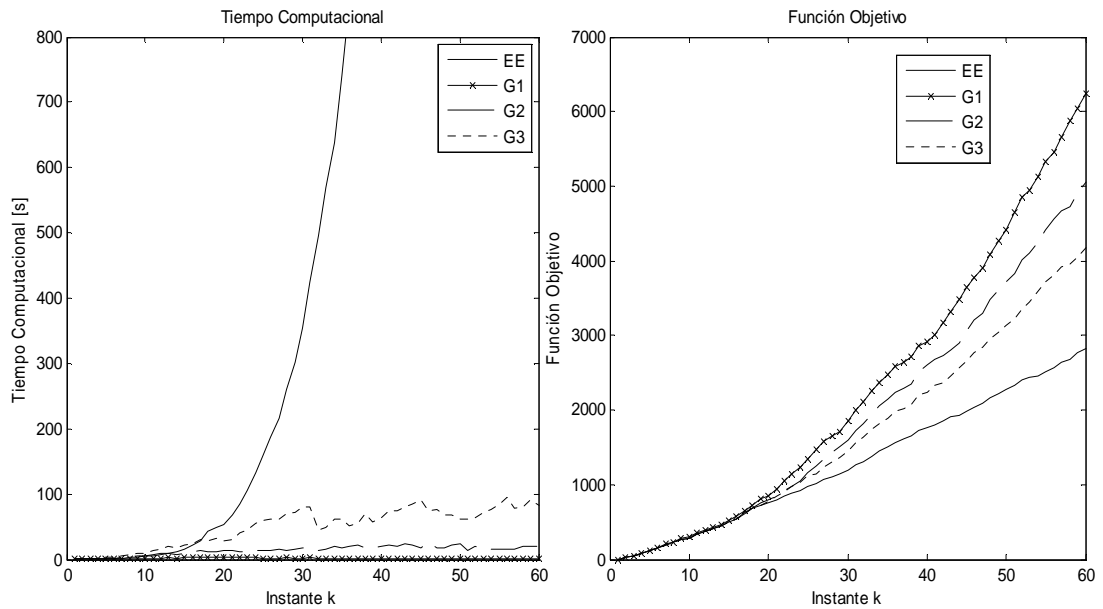


FIGURA 36. TIEMPO COMPUTACIONAL Y EVOLUCIÓN DE LA FUNCIÓN OBJETIVO.

TABLA 8. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.

	<b>Media Función Objetivo [min]</b>	<b>Media tiempo computacional [s]</b>
<b>Enumeración Explícita</b>	1297.4	1536.7
<b>Algoritmos Genéticos G1</b>	2288.2	1.4
<b>Algoritmos Genéticos G2</b>	1945.8	13.9
<b>Algoritmos Genéticos G3</b>	1694.6	49.7

**Prueba 2: Ruteo dinámico de vehículos bajo condiciones de demanda normal.**

En este caso, pocos requerimientos entran al sistema sobre el periodo de tiempo estudiado. La selección de soluciones sub-óptimas no es relevante debido a que las secuencias son cortas y los requerimientos mal ruteados alcanzan a ser satisfechos (el sistema alcanza a recuperarse).

La Figura 37 y la Tabla 9 muestran los valores del tiempo computacional y de la función objetivo. Mirando la evolución de la función objetivo en la Figura 37, se aprecia que GA se comporta similar a EE, mientras que un no significativo tiempo computacional es observado usando GA. La Tabla 9 muestra que cuando el número de individuos y generaciones aumenta, la solución converge al óptimo global (EE). Notar que la solución G3 es la misma que entrega EE, debido a que G3 calcula prácticamente todas las soluciones posibles, consumiendo además un elevado tiempo computacional.

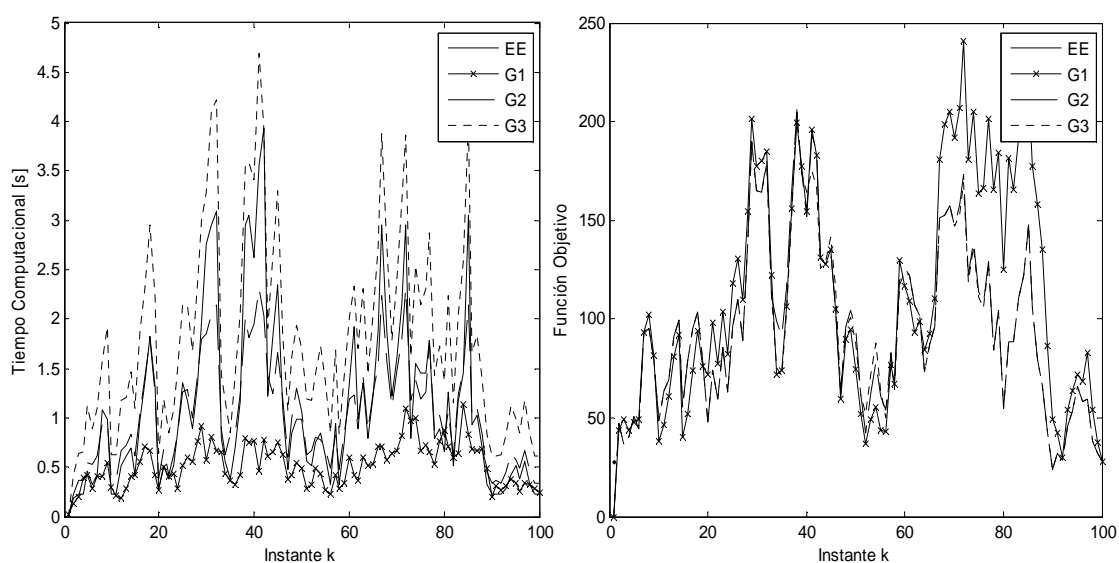


FIGURA 37. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL Y LA FUNCIÓN OBJETIVO.

TABLA 9. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.

	<b>Media Función Objetivo [min]</b>	<b>Media tiempo computacional [s]</b>
<b>Enumeración Explícita</b>	94.5	1.1
<b>Algoritmos Genéticos G1</b>	110.9	0.5
<b>Algoritmos Genéticos G2</b>	95.4	1.1
<b>Algoritmos Genéticos G3</b>	94.5	1.8

### Prueba 3: Ruteo dinámico de vehículos bajo condiciones de alta demanda.

Este caso es similar a la Prueba 1, pero aquí la secuencia previa que usa el algoritmo GA es la calculada por EE, es decir, en cada instante se usa una buena solución inicial. Figura 38 y Tabla 10 muestran la evolución de la función objetivo y su error con respecto a la solución óptima encontrada por el método EE. Aunque las secuencias son largas, el error en la función objetivo GA no crece significativamente.

De acuerdo a la Figura 38 y Tabla 10, las decisiones de despacho obtenidas por GA son similares a EE, sin importar el número de paradas planificadas.

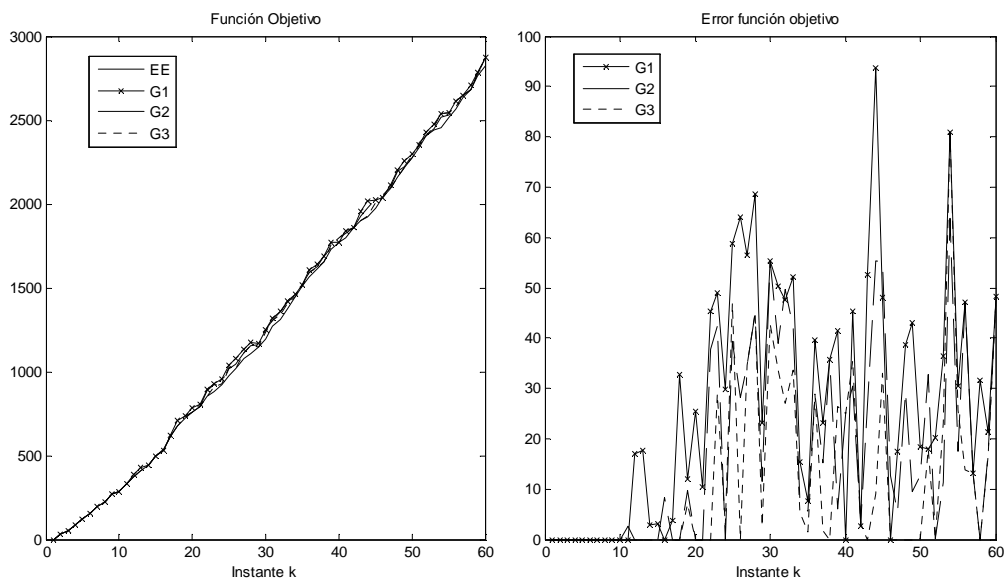


FIGURA 38. TIEMPO COMPUTACIONAL Y EVOLUCION DE LA FUNCIÓN OBJETIVO.

TABLA 10. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y ERROR.

	<b>Media Función Objetivo [min]</b>	<b>Media tiempo computacional [s]</b>
<b>Enumeración Explícita</b>	1297.4	-----
<b>Algoritmos Genéticos G1</b>	1324.0	26.6
<b>Algoritmos Genéticos G2</b>	1315.1	17.7
<b>Algoritmos Genéticos G3</b>	1309.3	11.9



En la siguiente sección, un ejemplo por simulación es mostrado, incluyendo todas los métodos propuestos (FCA con GA para uno, dos y tres pasos).

#### 4.5 Pruebas por simulación

Se utiliza un esquema de simulación de eventos discretos por un periodo de dos horas para evaluar el desempeño de la zonificación difusa y del método de algoritmos genéticos usando la política de *no swapping*. Se considera una flota de nueve vehículos, con capacidad de cuatro pasajeros cada uno.

Se considera un area urbana de servicio de aproximadamente 81 km<sup>2</sup>. Se asume que los vehículos viajan en línea recta entre las paradas a una velocidad comercial promedio de 20 Km./hr dentro de la región. Todas las simulaciones se realizan en dos horas representativas (14:00–14:59, 15:00–15:59) de un día laboral.

La data histórica generada vía simulación sigue los patrones de viaje mostrados en la Figura 39 con flechas.

Para la prueba por simulación, fueron generadas 120 llamadas sobre el periodo de simulación de dos horas según la distribución espacial y temporal de la misma característica que la data histórica. Respecto a la dimensión temporal, se asumió una distribución exponencial negativa para intervalos de tiempo entre llamadas con tasa 1[llamada/minuto] para ambas horas de simulación. Respecto a la distribución espacial, las coordenadas de pick-up y delivery fueron generadas aleatoriamente dentro de cada correspondiente zona. Las 10 llamadas del principio y del final de las pruebas fueron quitadas de las estadísticas para evitar distorsión de límites. 50 replicas de cada experimento fue realizado para obtener las estadísticas globales. Con respecto a la función de costos, un peso  $\alpha = 1$  fue usado, lo cual significa que tanto el tiempo de viaje como de espera tiene igual importancia.

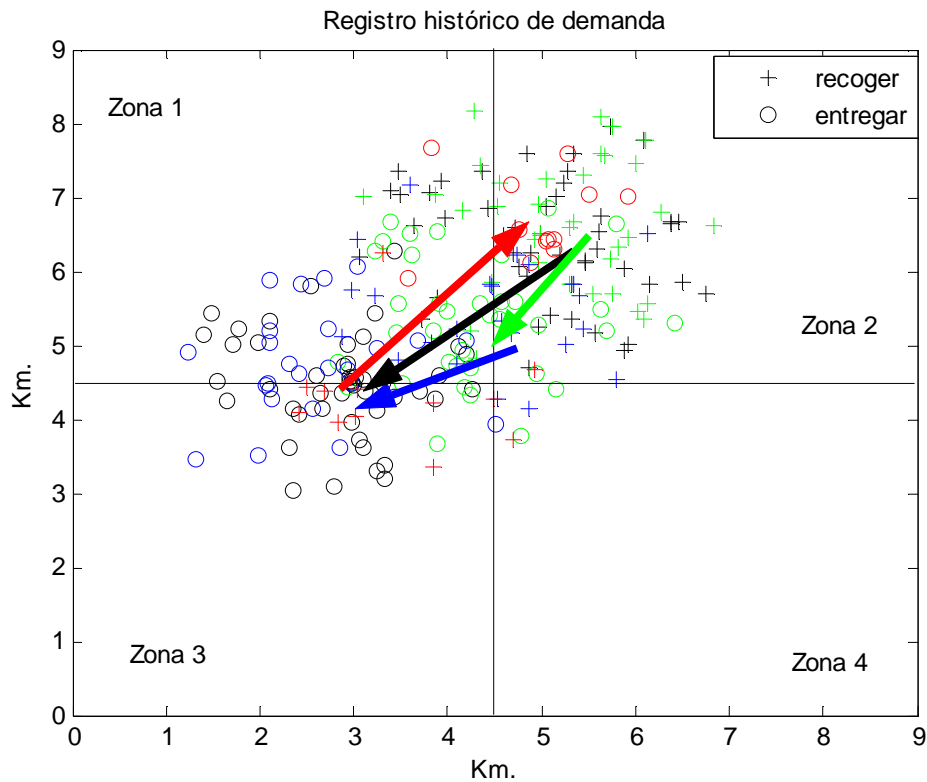


FIGURA 39. PATRONES ORIGEN-DESTINO.

#### 4.5.1 Zonificación difusa

Para comparar el desempeño de la zonificación difusa propuesta con la zonificación clásica (usando las cuatro áreas cuadradas mostradas en la Figura 39), se probó el algoritmo a dos pasos usando el algoritmo de enumeración explícita.

La Figura 40 muestra los grados de pertenencia de cada llamada a 4 clusters difusos obtenidos con el método de zonificación difusa. Tal y como la muestra la figura, el threshold es fijo e igual a 0.6 para considerar solo la data que ajusta de mejor forma a los patrones de viaje relevantes. A continuación, tal y como establece el procedimiento, fuzzy C-Means se aplica a los datos relevantes, los centros de los cluster son obtenidos y las correspondientes probabilidades calculadas.

La Tabla 11 muestra las coordenadas de los centros de clusters que corresponden a las coordenadas y pickup y delivery de los patrones de viaje y sus correspondientes

probabilidades. Por otro lado, la Tabla 12 muestra la zonificación clásica basada en 4 pares origen destino.

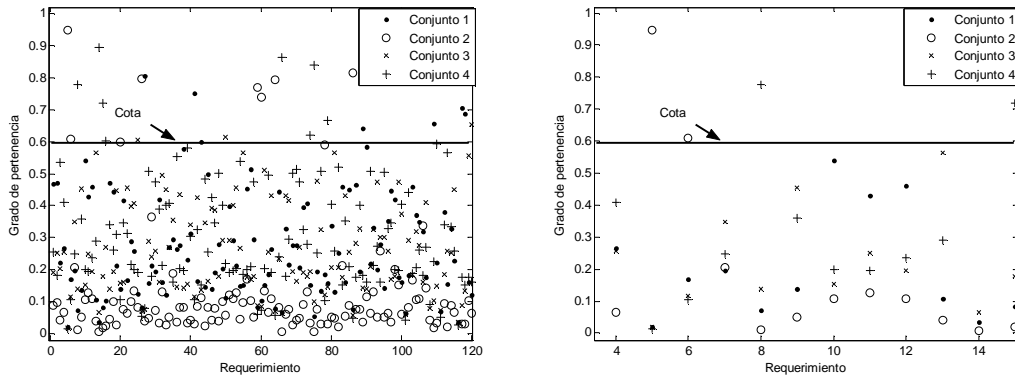


FIGURA 40. GRADO DE PERTENENCIA DE LOS REQUERIMIENTOS.

TABLA 11. COORDENADAS PATRONES DE VIAJE Y PROBABILIDADES: ZONIFICACIÓN DIFUSA.

X pickup	Y pickup	X delivery	Y delivery	Probabilidad
4.5540	5.7155	2.9218	4.7514	0.1282
3.7514	4.4812	5.2293	6.2232	0.2051
4.7989	6.6121	3.0751	4.4972	0.2564
5.2595	6.5057	4.3494	5.5161	0.4103

TABLA 12. COORDENADAS PATRONES DE VIAJE Y PROBABILIDADES: ZONIFICACIÓN CLÁSICA.

X pickup	Y pickup	X delivery	Y delivery	Probabilidad
6.75	6.75	6.75	6.75	0.0968
2.25	6.75	2.25	6.75	0.2151
6.75	6.75	2.25	2.25	0.3118
6.75	6.75	2.25	6.75	0.3763

Un parámetro relevante de sintonía para algoritmos de dos o pasos es el tiempo adecuado en el cual se considerará que ocurre las llamadas predichas ( $\tau$ ). Pese a que este parámetro tiene una interpretación física ya que corresponde al tiempo en que ocurre el siguiente requerimiento, se verá que su valor óptimo no corresponde al valor que se

obtiene de la tasa de llegadas de llamadas en el horario y por ende un análisis por sensibilidad será fundamental para obtener su valor óptimo. La Figura 41 muestra la función objetivo efectiva (considerando costos de usuario y operación) usando diferentes valores de  $\tau$  para tanto la zonificación clásica como la difusa. Diez replicas para cada valor de  $\tau$  fueron realizadas para obtener su valor óptimo. Para ambos métodos, el óptimo se ubicó en  $\tau = 5$ .

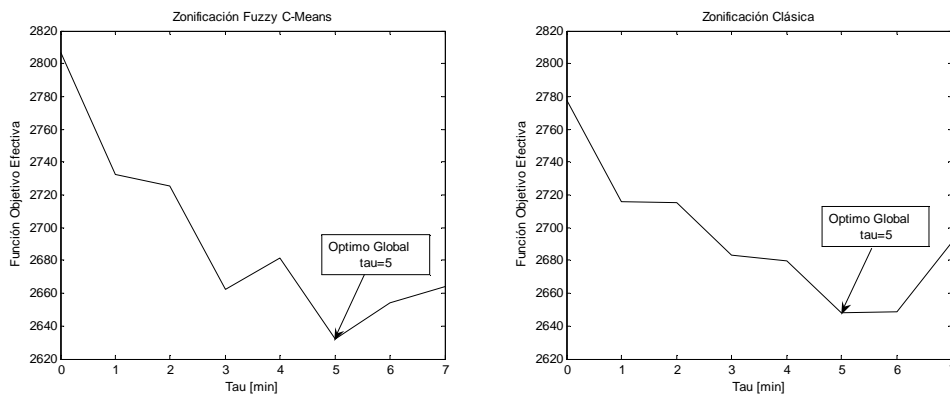


FIGURA 41. ANÁLISIS DE SENSIBILIDAD PARA  $\tau$  (ZONIFICACIÓN CLASICA Y DIFUSA).

Usando los valores óptimos para  $\tau$ , se realizaron 50 replicas del algoritmos a dos pasos basados en enumeración explícita para comparar los desempeños de los controladores basándose en ambos métodos de zonificación. La Tabla 13 muestra el valor medio y la desviación standard de los tiempos de espera, viaje y tiempo total de los usuarios. La comparación entre ambos métodos de zonificación se muestra en la misma tabla. Se observa que el tiempo de espera se reduce en un 3.36%, mientras que el tiempo de viaje se mantiene prácticamente constante y consecuentemente, el tiempo total se reduce en un 1.71%.

TABLA 13. COSTOS DE USUARIOS.

Control a dos pasos	Tiempo de espera (min)		Tiempo de viaje (min)		Tiempo total (min)	
	Media	Std	Media	Std	Media	Std
<b>Classic - zoning</b>	6.1437	0.87	10.2358	0.71	16.3795	1.44
<b>Fuzzy - zoning</b>	5.9370	0.72	10.1629	0.76	16.0999	1.36
<b>Ahorros</b>	0.2067		0.0729		0.2796	
<b>Mejora. (%)</b>	<b>3.36%</b>		<b>0.71%</b>		<b>1.71%</b>	

Los costos operacionales (media y desviación standart) para la flota completa son presentados en la Tabla 14. Además el costo total incluyendo costos de usuario y operacionales (como lo hace la función objetivo) se muestran también en la Tabla 14. Una mejora moderada es observada en ambas componentes. Sin embargo, la zonificación difusa es una alternativa sistemática que permite determinar patrones de viaje y sus correspondientes probabilidades sobre un sistema dinámico de llamadas on-line, guardando una relación directa con la forma que tiene la demanda y no ajustando la demanda a una cierta forma o distribución.

TABLA 14. COSTOS OPERACIONALES Y TOTALES.

<b>Control a dos pasos</b>	<b>Costos operacionales (min)</b>		<b>Costos totales efectivos (min)</b>	
	<b>Media</b>	<b>Std</b>	<b>Media</b>	<b>Std</b>
<b>Classic - zoning</b>	117.9	8.81	2699.4	122.84
<b>Fuzzy - zoning</b>	115.7	8.12	2651.1	112.86
<b>Ahorros</b>	2.2618		48.3163	
<b>Mejora. (%)</b>	<b>1.92%</b>		<b>1.79%</b>	

#### 4.5.2 HAPC basado en GA y zonificación difusa

Para analizar y evaluar el desempeño de la zonificación difusa en conjunto con HAPC usando GA, se realizaron pruebas por simulación para uno, dos y tres pasos. A continuación se presentan resultados de 50 replicas con GA ocupando 20 individuos y 20 generaciones. Se asume además que los patrones de viaje y de probabilidades se mantienen y son las mostradas en la Figura 40 para los escenarios a dos y tres pasos.

La Tabla 15 muestra los tiempos medios efectivos de espera, viaje y totales por pasajero usando HAPC basado en GA con la zonificación difusa, para diferentes horizontes de predicción.

Se observa que el tiempo de espera es significativamente reducido usando la estrategia a dos pasos (15.04%) e incluso mas para la estrategia a tres pasos (22.30%), cuando se comparan la estrategia miope a un paso. Además, se observa una mejora moderada en tiempos de viaje. Un caso interesante es la comparación entre dos pasos con tres pasos en términos de tiempo de viaje. De hecho, mejores en tiempo de viaje son mejores para dos pasos, principalmente debido a la incerteza que se tiene al aumentar cada vez el paso de predicción. Debido a este hecho, el ahorro en tiempo total del usuario obtenido con tres pasos es prácticamente igual al obtenido con dos pasos (9.78% y 9.45% respectivamente).

TABLA 15. COMPARACIÓN DE CONTROLADORES CON PREDICCIÓN DE UNO, DOS Y TRES PASOS.

	Tiempo de espera (min)		Tiempo de viaje (min)		Tiempo total (min)	
	Media	Std	Media	Std	Media	Std
<b>Un paso</b>	6.969	0.82	10.877	0.89	17.847	1.46
<b>Dos pasos</b>	5.921	0.67	10.238	0.79	16.159	1.42
<b>Tres pasos</b>	5.415	0.53	10.687	0.65	16.102	1.35
<b>Ahorros 2 pasos</b>	1.048		0.639		1.688	
<b>Mejora (%)</b>	<b>15.04%</b>		<b>5.87%</b>		<b>9.45%</b>	
<b>Ahorros 3 pasos</b>	1.554		0.190		1.745	
<b>Mejora (%)</b>	<b>22.30%</b>		<b>1.75%</b>		<b>9.78%</b>	

La Tabla 16 describe los costos operacionales (media y desviación standart) para toda la flota de vehículos. Se reporta en la tabla además el costo total efectivo. Se observa que los costos operacionales de la flota aumentan con los métodos de dos y tres pasos, sin embargo, el costo total efectivo es reducido por dos pasos (5.9%) y para tres pasos (4.47%). De los resultados, se puede decir que la estrategia a dos pasos resulta mejor que la estrategia a tres pasos y en parte se debe a que mientras mas largo el horizonte de predicción, menos confiables las predicciones.

TABLA 16. COSTOS OPERACIONALES Y TOTALES.

	<b>Costos operacionales (min)</b>		<b>Costos totales efectivos (min)</b>	
	<b>Media</b>	<b>Std</b>	<b>Media</b>	<b>Std</b>
<b>Un paso</b>	105.04	9.76	2730.0	127.832
<b>Dos pasos</b>	105.87	11.68	2568.7	114.516
<b>Tres pasos</b>	110.86	11.18	2608.0	112.444
<b>Ahorros 2 pasos</b>	-0.84		161.27	
<b>Mejora (%)</b>	<b>-0.79%</b>		<b>5.90%</b>	
<b>Ahorros 3 pasos</b>	-5.82		122.05	
<b>Mejora (%)</b>	<b>-5.54%</b>		<b>4.47%</b>	

#### 4.5.3 Disminución de áreas en zonificación clásica

En la misma área urbana, pero en distinto horario, se realiza un experimento para evaluar el funcionamiento del sistema en un periodo de tres horas. Se mantiene la flota de 9 vehículos y capacidad 4 pasajeros.

La simulación se hizo en el horario punta mañana de un día laboral, i.e.  $T_p = (7:00 - 7:59, 8:00 - 8:59, 9:00 - 9:59)$ .

El objetivo del experimento es probar los algoritmos a uno y dos pasos bajo distintas consideraciones en las condiciones y en la modelación. Uno de los factores más importantes es la distribución espacial y probabilidades de ocurrencia de los patrones de viaje, así como la disgregación del área total.

Siguiendo la metodología, se generó datos históricos asumiendo que el 90% de las llamadas tienen como requerimiento la visita de 6 principales pares origen destino (o patrón de llamada  $H=6$ ). Por simplicidad se asume que la probabilidad de ocurrencia de una llamada no cambia dentro del periodo de tiempo, pero que lo hacen dentro de cada intervalo. Para efectos de claridad y simplicidad, no se considera viajes dentro de la zona.

El primer ejemplo fue probado en 9 zonas homogéneas, donde la demanda histórica muestra que hay 6 patrones principales de viaje entre esas 9 zonas, como se muestra en la Figura 42. La probabilidad asociada a cada patrón de viaje se encuentra en la tabla justo debajo de la cada figura. Por ejemplo, respecto al horario entre 7:00 y 7:59 AM, la probabilidad de que un requerimiento pida ir desde la zona 1 a la 5 es de 0.2 entre todos los viaje que ocurren dentro de ese intervalo. La probabilidad de otros viajes entre zonas no consideradas dentro de las seis más relevantes, se despreció.

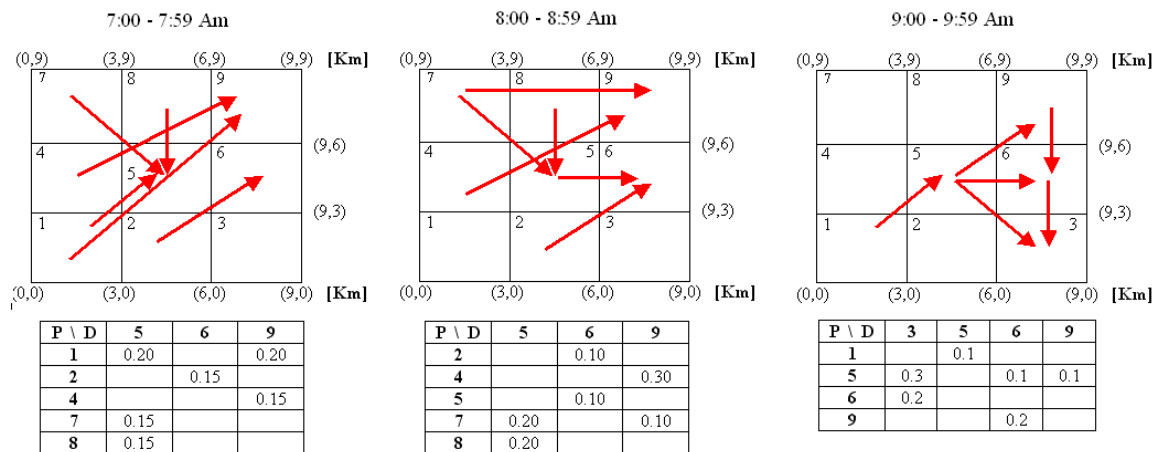


FIGURA 42. PATRONES ORIGEN-DESTINO (9 ZONAS) P \ D: ZONAS PICKUP Y DELIVERY.

Un segundo experimento realizado, consistió en dividir el area total en 4 zonas en lugar de nueve. La Figura 43 muestra un esquema similar para este caso similar a la Figura 42.



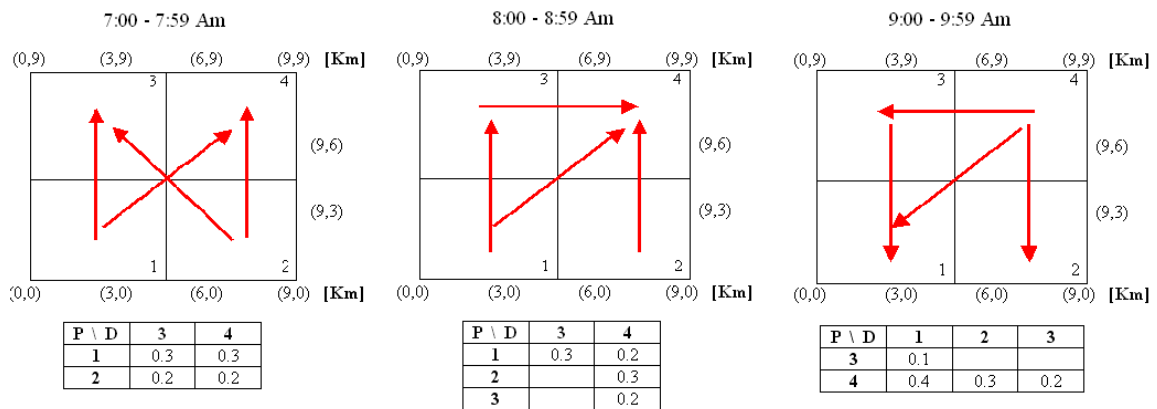


FIGURA 43. PATRONES ORIGEN-DESTINO (4 ZONAS) P \ D: ZONAS PICKUP Y DELIVERY.

En términos de demanda, 90 llamadas fueron generadas sobre el periodo completo de simulación de tres horas según la distribución temporal y espacial de la demanda histórica. Para el patrón temporal, se asumieron distribuciones exponencial negativa para intervalos entre llamadas, con tasas de 0.5 [llamadas/minuto], 1 [llamada/minuto] y 0.5 [llamadas/minuto] para la primera, segunda y tercera hora de simulación. En cuanto a la distribución espacial, los puntos de recogida y entrega se generaron aleatoriamente dentro de cada zona para replicar el patrón de viaje y las probabilidades mostradas en las figuras 42 y 43, dependiendo del experimento. Arbitrariamente, los vehículos fueron inicialmente localizados en los centros de masa de las zonas. Este hecho no afecta a las estadísticas finales ya que un periodo razonable de estabilización fue considerado, para evitar distorsiones de borde. 30 réplicas para cada experimento fueron realizadas para obtener los estadísticos globales. Con respecto a la función de costo, se consideró  $\alpha = 1$  con lo cual el tiempo de viaje tiene igual importancia al tiempo de espera. Los algoritmos a un paso y a dos serán evaluados y comparados en ambos casos.

Como ya se ha indicado, un parámetro de sintonización importante es el paso entre llamadas para la estrategia a dos pasos ( $\tau$ ). Para encontrar su valor óptimo se realizó análisis de sensibilidad en cada caso (9 zonas y 4 zonas). En la Figura 44 se muestra este análisis, a partir del cual se obtuvo el valor óptimo  $\tau = 5$  para 9 zonas y  $\tau = 2$  para 4 zonas.

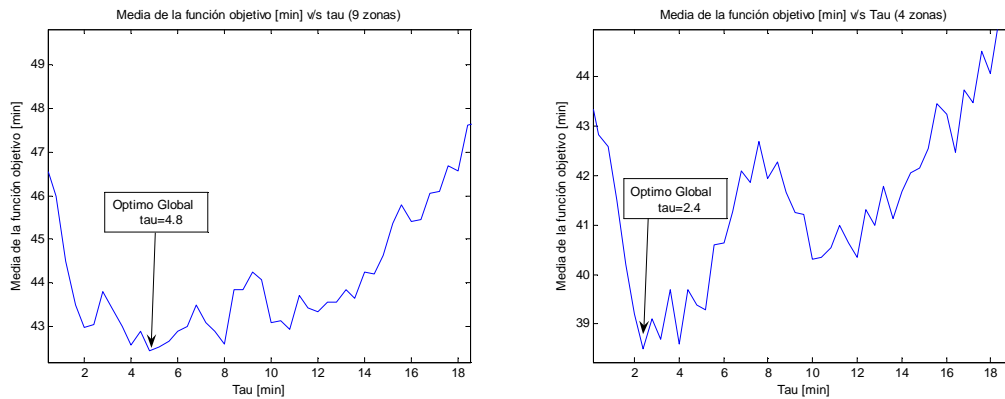


FIGURA 44. ANÁLISIS DE SENSIBILIDAD PARA  $\tau$  (PARTICIONES DE 9 Y 4 ZONAS).

En las tablas 17 y 18 se muestran la media y desviación estándar de los valores de tiempo de espera, tiempo de viaje y tiempo total para 30 replicas. Esos valores capturan los índices del sistema en términos de nivel de usuario, cuando se analizan los algoritmo a uno y dos pasos para ambos experimentos con 9 zonas (6 patrones de viaje) y 4 zonas (4 patrones de viaje). Adicionalmente, la Tabla 19 muestra el costo operacional. Los periodos de 20 minutos al inicio y 20 minutos al término de la experiencia no son considerados en las estadísticas.

Los estadísticos de las tablas 17, 18 y 19 muestran el desempeño de los dos algoritmos sobre el periodo completo de tres horas excluyendo los intervalos iniciales y finales de 20 minutos. De los resultados mostrados en las tablas anteriores, se aprecia que el algoritmo predictivo a dos pasos, sistemáticamente se desempeña mejor que el algoritmo miope de un paso, apoyando a la tesis de que es necesario incorporar poder predictivo en el problema dinámico de ruteo en tiempo real para lograr mejores resultados.

TABLA 17. COMPARACIÓN UN PASO Y DOS PASOS. 9 ZONAS, 6 PATRONES DE VIAJE.

	Tiempo de espera (min)		Tiempo de viaje (min)		Tiempo total (min)	
	Media	Std	Media	Std	Media	Std
<b>Un paso</b>	20.6479	2.27	25.4290	1.02	46.0769	2.98
<b>Dos pasos</b>	18.3375	1.75	24.9952	1.16	43.3327	2.61
<b>Ahorro</b>	2.3104		0.0171		0.0593	
<b>Mejora (%)</b>	<b>11.12%</b>		<b>1.71%</b>		<b>5.93%</b>	

TABLA 18. COMPARACIÓN UN PASO Y DOS PASOS. 4 ZONAS, 4 PATRONES DE VIAJE..

	Tiempo de espera (min)		Tiempo de viaje (min)		Tiempo total (min)	
	Media	Std	Media	Std	Media	Std
<b>Un paso</b>	20.2571	2.32	25.3887	0.98	45.6458	2.96
<b>Dos pasos</b>	17.1415	5.03	23.3106	6.54	40.4520	11.47
<b>Ahorro</b>	3.1156		2.0781		5.1937	
<b>Mejora (%)</b>	<b>15.38%</b>		<b>8.19%</b>		<b>11.38%</b>	

TABLA 19. COMPARACIÓN UN PASO Y DOS PASOS, COSTO OPERACIONAL POR VEHÍCULO.

Tiempo de viaje vehículo (min/veh)	4 zonas, 4 patrones		9 zonas, 6 patrones	
	Media	Std	Media	Std
<b>Un paso</b>	141.5668	13.5577	152.7011	11.2001
<b>Dos pasos</b>	141.4056	9.6822	153.4361	6.7921

Los ahorros más importantes con el enfoque predictivo se aprecian en los tiempos de espera (15.38% en el mejor caso). De este resultado, se puede inferir que los beneficios en la predicción se deben a que se evita las esperas extras de los futuros puntos de recogida, lo cual es en algo manejable por el despachador debido a que conoce data histórica con los patrones comunes de viaje.

A partir del ejemplo, con la estrategia predictiva se obtienen mejoras en los tiempos de viaje mejores que las obtenidas en tiempo de espera (8.19% en el mejor caso), y tampoco hay un incremento sustancial en el costo operacional, el cual prácticamente se mantiene constante.

A partir de los resultados de la zonificación mas agregada (Tabla 18), el resultado ha sido un esquema de ruteo mas eficiente a dos pasos que la solución obtenida a un paso, cuando se compara con los resultados de 9 zonas (Tabla 17). Una razón que explica la diferencia es que mientras mas agregado el area es, mas precisa es la predicción de los patrones y probabilidades. Entonces, es más fácil encontrar rutas de vehículos que siguen a los conocidos patrones de viaje, resultando en mejores predicciones de los potenciales reruteos.

Para una representación mas detallada de como el sistema dinámico de ruteo funciona, la Figura 46 muestra el tiempo efectivo de espera, de viaje y total de cada pasajero durante su viaje, para una replica específica (tomada entre las 30 repeticiones resumidas anteriormente), para el experimento con mayor agregación (4 zonas).

En la Figura 45 se muestra los patrones de desempeño (en términos de tiempo de espera y viaje) para cada pasajero. Esta es una réplica razonable con 26.45% y 6.64% de ahorros en tiempos de espera y viaje respectivamente en comparación con la estrategia miope a un paso. Mirando la Figura 45a, es claro que, con excepción de uno o dos casos, muchos pasajeros considerablemente ven disminuido su tiempo de espera efectivo. La Figura 46b por otro lado, muestra un buen desempeño del algoritmo a un paso en términos de tiempo de viaje, resultando en bajas mejoras para el algoritmo a dos pasos.

Más aun, nótese que cuando la intensidad de llamadas crece (aproximadamente cuando se sirve al requerimiento 30), el algoritmo a dos pasos principalmente en cuanto a tiempo de espera consistentemente se desempeña mejor que el método a un paso para todos los requerimientos que son servidos. El mismo fenómeno ocurre en tiempo de espera pero con menor relevancia (Figura 45b).

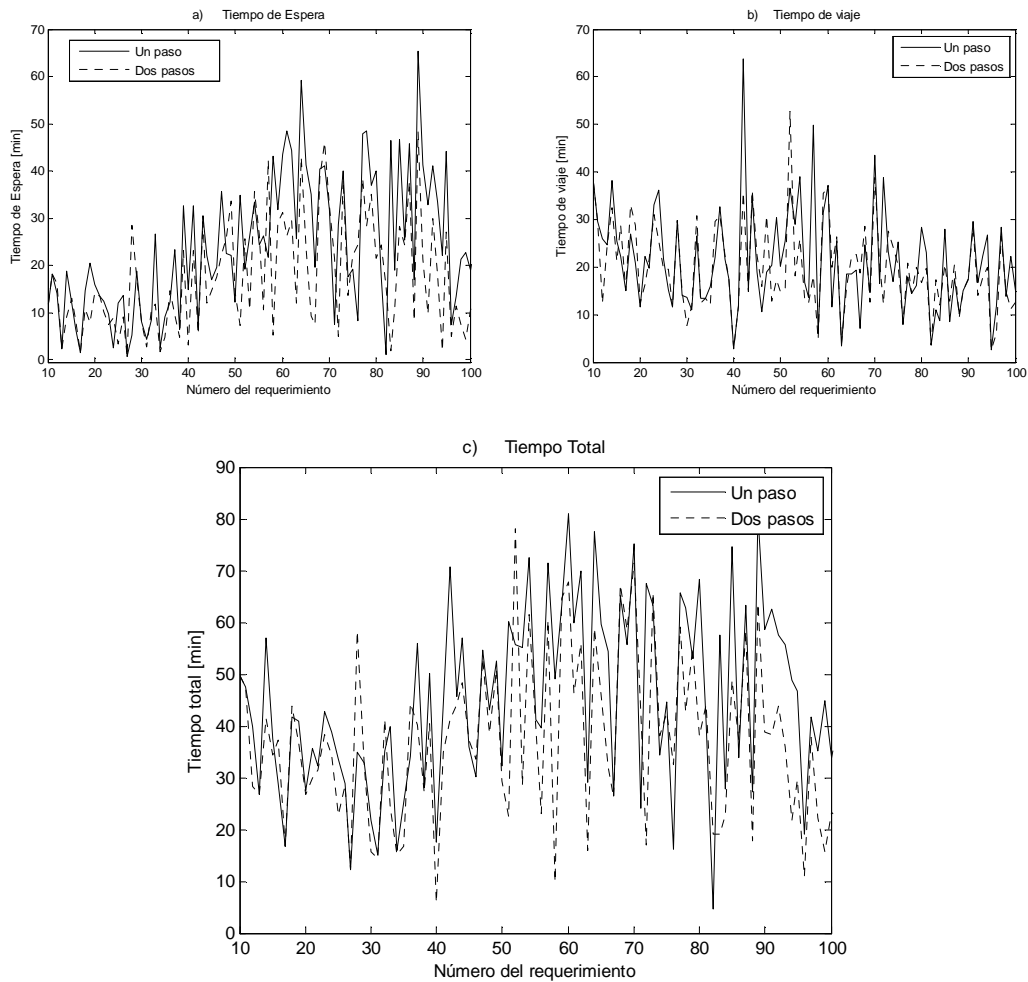


FIGURA 45. TIEMPO DE ESPERA, TIEMPO DE VIAJE Y TIEMPO TOTAL POR PASAJERO PARA UNA REALIZACIÓN (4 ZONAS).

Una conclusión que se puede sacar de esta tendencia es que la información futura es más importante al tomar decisiones de ruteo en tiempo real a medida que el tamaño del problema aumenta. En ese sentido el análisis de algoritmos eficientes para resolver el problema en poco tiempo es fundamental.

#### **4.5.4 Síntesis y conclusiones**

En este trabajo se formuló analíticamente el problema DPDP basado en un enfoque HAPC en el cual se incorpora dentro de la formulación del controlador la información de demanda histórica, mediante la cual se realizan predicciones y se mejora las decisiones de despacho. Las mayores contribuciones son tres. La primera es el modelo analítico en espacio de estado. La segunda, la zonificación difusa que calcula probabilidades y patrones de viaje basados en data histórica, y siguiente la geometría espacial de la demanda. Y la tercera, y basado en un enfoque analítico, los algoritmos genéticos son propuestos y probados en un ejemplo por simulación.

Otra contribución de esta formulación es el uso de métodos de inteligencia computacional para encontrar mejores decisiones de despacho bajo escenarios no miopes (predicción a más de dos pasos). Particularmente, GA es mostrado como un algoritmo eficiente de optimización para el problema DPDP. Se probó que bajo ciertas condiciones, una estrategia a más de dos pasos puede ser resuelta usando GA en un tiempo computacional razonable.

Por otro lado, se propuso zonificación difusa basada en clustering difuso para sistemáticamente estimar los patrones de origen destino a partir de data histórica y obtener consecuentemente resultados más confiables para las probabilidades de esos patrones. El método de zonificación difusa mejora el desempeño de los algoritmos predictivos en la medida que sea bien ajustado en cuanto a número óptimo de clusters y de eliminación de datos que no aportan en la clasificación. Por otro lado, parece ser más razonable usar esta estrategia en casos en que las zonas de demandas se traslapen y en demandas que no parecieran seguir patrón alguno.

## 5. CONCLUSIONES

Se presentaron resultados y metodología para aplicar el enfoque de Control Predictivo Híbrido (HPC) al problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta.

En el Capítulo 2 se presentó el problema de Control Predictivo General y a partir de esta formulación se mostraron algunos casos importantes como lo son HPC basado en modelos lineales y HPC para procesos no lineales basado en modelos difusos. La principal ventaja del control predictivo híbrido está relacionada a la inclusión de las variables enteras dentro de la formulación del problema de optimización lo que representa de mejor forma al proceso real. Entonces, se evitan las simplificaciones y la solución óptima global en cada instante se calcula y se aplica al proceso.

En el Capítulo 3, se aplicó la metodología sistemática de resolución de HPC a los problemas de control de temperatura de un estanque con tres calefactores on/off (variable manipulada entera) y al problema de control de nivel de un sistema de estanques no lineal híbrido (variable manipulada y variable de estado discreta).

Algunas de las alternativas de solución del problema de control predictivo híbrido son enumeración explícita (HPC&EE) y Branch and Bound (HPC&BB) entre otras. HPC&EE requiere de un alto esfuerzo computacional y por ejemplo, no fue capaz de resolver el problema de control de temperatura real en un tiempo razonable por lo que no se pudo implementar experimentalmente. En HPC&EE el tiempo computacional aumenta exponencialmente con el número de predicciones. HPC&BB es una mejor alternativa en comparación a HPC&EE pero si el horizonte de control y de predicción aumentan mucho, el tiempo computacional se verá incrementado también. Por otro lado, HPC&GA ha mostrado ser eficiente y permite alcanzar soluciones cercanas a la óptima. Además, muestran un buen desempeño en cuanto a tiempo computacional, y entonces permite encontrar buenas soluciones en tiempo reducido. Esta última característica es una consecuencia de evitar cálculo de gradiente.

Se puede concluir entonces que los Algoritmos Genéticos corresponden a una herramienta poderosa para resolver problemas de optimización y muy flexibles ya que se adecuan al problema, y no el problema a los algoritmos.

Los resultados experimentales obtenidos (escasos en la literatura de Control Predictivo Híbrido) se lograron utilizando HPC&BB y HPC&GA en el control de temperatura del estanque. Los resultados mostraron ser muy similares, lo cual demuestra que los algoritmos genéticos son una buena herramienta ya que se puede ajustar su precisión en función del tiempo computacional disponible.

También en el capítulo 2 y 3 se presentó un nuevo enfoque para controlar sistemas no lineales con variables de estado y manipuladas tanto continuas como enteras. Usando una estrategia de control predictiva ocupando modelación difusa, el problema de los estados discretos y variables continuas puede ser resuelto.

El elemento clave de la modelación difusa de sistemas híbridos está en la identificación y detección de las zonas de switch, el cual es realizado mediante el uso de clustering difuso y análisis de componentes principales. La optimización de la función objetivo es en este caso un problema del tipo NP-Hard, el cual pudo ser resuelto eficientemente mediante algoritmos genéticos. El algoritmo de control propuesto HFPC-GA fue exitosamente probado en el control de sistema de estanques híbrido tanto en tiempo computacional como en precisión.

En el Capítulo 4 se mostró en detalle la solución basada en HPC para el problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta.

La formulación analítica del problema de control incorporó la información de demanda histórica, con la cual se realizaron predicciones y le entregaron poder predictivo al controlador. Como ya se mencionó anteriormente, las mayores contribuciones de este capítulo son el modelo analítico en espacio de estado, la zonificación difusa y la favorable aplicación de los algoritmos genéticos. No se encontraron trabajos en la literatura con el mismo enfoque por lo que las tres contribuciones son novedosas y valiosas para el área de Ingeniería Civil en Transporte.



Se concluye además que EE trabaja bien para problemas pequeños (pocas paradas planificadas por vehículo). Sin embargo a medida que el problema aumenta de tamaño (por ejemplo, bajo escenarios más reales), GA se convierte en una alternativa atractiva para resolver el problema en tiempo computacional reducido. Además se cree que GA puede resolver problemas mucho más complejos, por ejemplo con muchas secuencias, funciones objetivo mucho más sofisticadas, problemas con restricciones, etc. Debe notarse que el usar como criterio de parada el número de generaciones permite acotar el tiempo computacional, pero otro lado la selección del número adecuado de individuos por generaciones es crítico cuando se trata de lograr mejor calidad de soluciones.

Por otro lado, la zonificación difusa propuesta, basada en clustering difuso, permitió sistematizar las estimaciones de los patrones de origen destino a partir de data histórica y obtener mejores probabilidades de esos patrones. El método de zonificación difusa mejoró el desempeño de los algoritmos predictivos cuando fue bien ajustado.

En resumen, la metodología integrada (HAPC difuso basado en GA) permite resolver a más de dos pasos de predicción el problema DPDP y manejar la incerteza en demanda con patrones heterogéneos.

En trabajo futuro, se propone combinar los datos históricos (offline) con información online en modelo mas elaborados capaces de capturas eventos inminentes que podrían afectar el desempeño del sistema. Segundo, generar funciones objetivo más completas y rigurosas, y restricciones como ventanas de tiempos (hard y soft), y una mejor incorporación de los costos operacionales. También se cree posible mejorar la estimación de las variables de sintonía, como número de llamadas probables, predicción del paso óptimo futuro  $\tau$  el cual es desconocido, horizonte de predicción, políticas de servicio y búsqueda sobre diferentes espacios de soluciones factibles. Respecto a lo último, relajar la política operacional de *no-swapping* para probar decisiones de despacho menos restrictivas, para lo cual la formulación analítica parece ser útil. La incorporación de velocidad en el modelo de estado es también algo importante por hacer para obtener una mejor representación del sistema de tráfico real y estocasticidad en tiempos de viaje.

## REFERENCIAS

- BABUSKA, R. 1999. Fuzzy Modeling for Control. Kluwer Academic Publishers.
- BEMPORAD, A., MORARI, M. 1999. "Control of systems integrating logic, dynamics and constraints," *Automatica*, vol. 35, pp. 407–427.
- BEMPORAD, A., MORARI, M. 2000. "Predictive control of constrained hybrid systems," *Automatica*, pp. 71–78.
- BEMPORAD, A., HEEMELS, W., DE SCHUTTER, B. 2002. "On hybrid systems and closed-loop mpc systems," *IEEE Transaction on Automatic Control*, vol. 47, pp. 863–869.
- BERTSIMAS, D. 1992. A vehicle routing problem with stochastic demand. *Operations Research*, 40, 574-585.
- BERTSIMAS, D., JAILLET, P., ODoni, A. 1990. A priori optimization. *Operations Research*, 38 (6), 1019-1033,.
- BEZDEK, J. 1973. Fuzzy Mathematics in Pattern Classification. PhD Thesis, Applied Math. Center, Cornell University, Ithaca.
- BIANCHESSI, N., RIGHINI, G. 2006. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research* (Article in Press).
- CEYLAN H., BELL, M.G.H. 2004. Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing, *Transportation Research B*, 38, 329-342.
- CLARKE, D., MOHTADI, C., TUFFS, P. 1987. Generalized predictive control (Parts I and II) . *Automatica*, Vol. 23, N°2, pp. 137-160.
- CORTÉS C.E., JAYAKRISHNAN, R. 2004. Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems, *Proceeding of the Triennial Symposium on Transportation Analysis (TRISTAN) V*, Guadeloupe, French West Indies.
- CORTÉS C. E., NÚÑEZ A., SÁEZ D., MUÑOZ D. 2006, "Hybrid predictive control for a real-time routed transit system". En tercera revisión para la revista *ISI Transportation Science*.
- DIAL, R. 1995. Autonomous Dial a Ride Transit – Introductory Overview. *Transportation Research - Part C* 3, 261-275.
- DORIGO, M., STÜTZLE, T. 2004. *Ant Colony Optimization*. The MIT Press.
- DRÉO, J., PÉROWSKI, A., SIARRY, P., TAILLARD, E. 2006. *Metaheuristics for Hard Optimization Methods and Case Studies*, Springer-Verlag.

FILIPPEC, M., SKRLEC, D., SLAVKO, K. 1998. An efficient implementation of genetic algorithms for constrained vehicle routing problem. Proceedings of IEEE International Conference on System, Man and Cybernetics, SMC'98.

FLOUDOS, C. 1995. Non-linear and mixed integer optimization, Oxford University Press.

GENDREAU, M., GUERTIN, F., POTVIN, J., TAILLARD, E. 1999. Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science* 33, 381-390.

GENDREAU, M., LAPORTE, G., SÉGUIN, R. 1995. An Exact algorithm for the Vehicle Routing Problem with stochastic Customers and Demands. *Transportation Science* 29, 143-155.

GENDREAU, M., LAPORTE, G., SEGUIN, R. 1996. Stochastic Vehicle Routing. *European Journal of Operational Research*, 88, 3-12.

GIRIMONTE, D., BABUSKA, R. 2004. "Structure for nonlinear models with mixed discrete and continuous inputs: a comparative study," Proc. of IEEE International Conf. on Systems, Man and Cybernetics, pp. 2392-2397.

GODFREY, G., POWELL, W.B. 2002. An Adaptive Dynamic Programming Algorithm for Stochastic Resource Allocation Problems I: Single Period Travel Times. *Transportation Science* 36, 21-39.

HAGHANI, A., JUNG, S. 2005. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research* 32, 2959-2986.

ICHOUA, S., GENDREAU, M., POTVIN, J.Y. 2005. Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching. Accepted for publication in *Transportation Science*.

JIH, W., YUN-JEN, J. 1999. Dynamic Vehicle routing using hybrid genetic algorithms. Proceeding of the IEEE International Conference on Robotics & Automation, Detroit, Michigan, May 1999, 453-458.

KENNEDY J., EBERHART R. 1998. "Swarm Intelligence". Morgan Kaufmann Publishers.

KLEYWEGT, A.J., PAPASTAVROU, J.D. 1998. The Dynamic and Stochastic Knapsack Problem. *Operations Research*, 46, 17-35.

LE BOUTHILLIER, A., CRAINIC, T.G. 2005. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research*, 32, 1685-1708.

LI, F., GOLDEN, B., WASIL, E. 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32, 1165-1179.

MADSEN, O., RAVEN, H., RYGAARD, J. 1995. A heuristics algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60, pp. 193-208.

MAN, K., TANG, K., KWONG, S. 1998. Genetic Algorithms, Concepts and Designs. Springer.

MONTEMANNI, R., GAMBARDELLA, L.M., RIZZOLI, A.E., DONATI, A. 2005. Ant colony system for a dynamic vehicle routing problem. Journal of Combinatorial Optimization, Issue: Volume 10, Number 4, December, Pages: 327 – 343.

NÚÑEZ A., SÁEZ D., CORTÉS C. E., “Problema de ruteo dinámico de una flota de vehículos con un enfoque de control predictivo híbrido basado en algoritmos genéticos”, Actas del XII Congreso Chileno de transporte, Valparaíso, Chile, 17-21 Octubre 2005, ISSN: 0717-3482, páginas 361-373.

NÚÑEZ A., SÁEZ D., OBLAK S., SKRJANC I., “Hybrid Predictive Control based on Fuzzy Model”. IEEE World Congress on Computational Intelligence, FUZZ IEEE 2006, Vancouver, BC, Canadá. 16-21 Julio.

OSMAN, M.S., ABO-SINNA, M.A., MOUSA, A.A. 2005. An effective genetic algorithm approach to multiobjective routing problems (MORPs), Applied Mathematics and Computation, 163, 769-781.

PALM, R., DRIANKOV, D. 1998. “Fuzzy switched hybrid systems –Modeling and identification,” in Proc. of the 1998 IEEE ISCI/CIRA/SAS Joint Conf., Gaithersburg MD, Sept.14-17, pp. 130-135.

POTOCNIK, B., MUSIC, G., ZUPANCIC, B. 2004. “Model predictive control systems with discrete inputs,” in Proc.12th IEEE Mediterranean Electrotechnical Conference, Dubrovnik, Croatia, 2004, pp:383–386.

POWELL, W.B. 1988. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. B.L. Golden and A.A. Assad editors, Vehicle routing methods and studies.

PSARAFTIS, H. 1998. Dynamic vehicle routing problems, B.L. Golden and A.A. Assad editors, en Vehicle routing methods and studies, 223-248.

PSARAFTIS, H. 1980. A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem. Transportation Science 14(2), pp. 130-154.

SÁEZ D. 1995. “Diseño y Evaluación en Tiempo Real de Controladores Difusos y su Aplicación al Control de un Péndulo Invertido”. Tesis para obtener el grado de Magíster en Ciencias de la Ingeniería, mención Ingeniería Eléctrica, Pontificia Universidad Católica de Chile (PUC).

SÁEZ D., CORTÉS C. E., NÚÑEZ A. 2007. “Hybrid predictive control based on Genetic Algorithm and Fuzzy Clustering for a real-time routed transit system”. Volumen especial de la revista ISI Computer & Operations Research (Aceptada).

SÁEZ D., NÚÑEZ A., CHONG A., ZÚÑIGA R. y ALBORNOZ V. 2004. “Control híbrido predictivo aplicado al control de temperatura de un estanque”, Actas del XVI Congreso de

la Asociación Chilena de Control Automático 2004, Santiago, Chile, 2-4 Noviembre 2004, páginas 61-67.

SARIMVEIS, G. 2003. "Fuzzy model predictive control of non-linear processes using genetic algorithms," *Fuzzy Sets and Systems*, N° 139. pp. 59-80.

SAVELSBERGH M.W.P., SOL, M. 1995. The general pickup and delivery problem. *Transportation Science*, 29(1), 17-29.

SECOMANDI, N. 1998. Exact and Heuristics Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. Ph.D. Dissertation, College of Business Administration, University of Houston.

SECOMANDI, N, 2000. Comparing Neuro-Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. *Computers and Operations Research* 27, 1201-1225.

SKRLEC, D., FILIPEC, M., KRAJCAR, S. 1997. A heuristic modification of genetic algorithm used for solving the single depot capacited vehicle routing problem. *Proceedings of Intelligent Information Systems, IIS '97*, 184-188.

SLUPPHAUG, O., VADA, J., FOSS, B. 1997. "MPC in systems with continuous and discrete control inputs," in *Proc. of American Control Conference*, Albuquerque, NM, USA.

SLUPPHAUG, O., FOSS, B. 1997. "Model predictive control for a class of hybrid systems," in *Proc. of European Control Conference*, Brussels, Belgium.

SPIVEY, M., POWELL, W.B. 2004. The Dynamic Assignment Problem. *Transportation Science* 38(4), 399-419.

TANAKA, K., IWASAKI, M., WANG, H.O. 2001. "Switching control of an R/C Hovercraft: stabilization and smooth switching," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 853-863, Dec. 2001.

TARANTILIS, C., IOANNOU, G., PRASTACOS, G. 2005. Advanced vehicle routing algorithms for complex operations management problems. *Journal of Food Engineering* 70, 455-471.

TARANTILIS, C. 2005. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research* 32, 2309-2327.

THOMAS J., DUMUR, D., BUISSON, J. 2004. Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition, *Proceedings of American Control Conference*, Boston, Massachusetts, USA.

TIGHE A., SMITH F., LYONS G. 2004. Priority based solver for a Real-Time Dynamic Vehicle Routing. *IEEE International Conference on Systems, Man and Cybernetics*, pgs: 6237-6242.

TOHT P., VIGO D. 2003. The granular Tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, Volume: 15. Fall, Number: 4. Pgs: 0333-0346.

TONG Z., NING L., DEBAO, S. 2004. Genetic Algorithm for Vehicle Routing Problem with Time Window with Uncertain Vehicle Number. *Proceeding of the 5<sup>th</sup> World Congress on Intelligent Control and Automation*, June 15-19, Hangzhou, P.R. China. Pgs: 2846-2849.

TOPALOGLU H., POWELL W.B. 2005. A Distributed Decision-Making Structure for Dynamic Resource Allocation Using Non Linear Functional Approximations. *Operations research* 53(2), 281-297.

TSANG, T., CLARKE, D. 1988. Generalised predictive control with input constraints. *IEE Proceedings*, Vol. 135-D, N<sup>o</sup> 6, pp. 451-460.

YANG W.H., MATHUR K., BALLOU R.H. 2000. Stochastic Vehicle Routing Problem with Restocking. *Transportation Science* 34(1), 99-112.

WATERS, C.D.J. 1989. Vehicle Routing Problems with Uncertainty and Omitted Customers. *Journal of the Operational Research Society* 40, 1099-1108.

WITSENHAUSEN, H. S. 1966. "A class of hybrid-state continuous time dynamic systems," *IEEE Trans. On Automatic Control*, vol. 11, No. 2, pp. 161-167.

ZHU K.Q., 2003. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. *15th IEEE International Conference on Tools with Artificial Intelligence*, November 3-5, 2003, 176-183.

## INDICE DE TABLAS

TABLA 1. VALORES MEDIOS DE LA FUNCIÓN OBJETIVO	42
TABLA 2. TIEMPO COMPUTACIONAL.	42
TABLA 3. VALORES MEDIOS FUNCIÓN OBJETIVO (9), RESULTADO EXPERIMENTAL	45
TABLA 4. PARÁMETROS MODELO DIFUSO	49
TABLA 5. VALORES MEDIOS DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.	60
TABLA 6. DESVIACIÓN STANDARD DE (Y-R) Y $\Delta U$ .	60
TABLA 7. PROBABILIDADES DE CADA PATRON DE VIAJE USANDO DOS CLUSTERS DIFUSOS.	93
TABLA 8. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.	102
TABLA 9. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y TIEMPO COMPUTACIONAL.	103
TABLA 10. VALOR MEDIO DE LA FUNCIÓN OBJETIVO Y ERROR.	104
TABLA 11. COORDENADAS PATRONES DE VIAJE Y PROBABILIDADES: ZONIFICACIÓN DIFUSA.	107
TABLA 12. COORDENADAS PATRONES DE VIAJE Y PROBABILIDADES: ZONIFICACIÓN CLÁSICA.	107
TABLA 13. COSTOS DE USUARIOS.	108
TABLA 14. COSTOS OPERACIONALES Y TOTALES.	109
TABLA 15. COMPARACIÓN DE CONTROLADORES CON PREDICCIÓN DE UNO, DOS Y TRES PASOS.	110
TABLA 16. COSTOS OPERACIONALES Y TOTALES.	111
TABLA 17.COMPARACIÓN UN PASO Y DOS PASOS. 9 ZONAS, 6 PATRONES DE VIAJE.	115
TABLA 18. COMPARACIÓN UN PASO Y DOS PASOS. 4 ZONAS, 4 PATRONES DE VIAJE..	115
TABLA 19. COMPARACIÓN UN PASO Y DOS PASOS, COSTO OPERACIONAL POR VEHÍCULO.	115

## INDICE DE FIGURAS

FIGURA 1. ESQUEMA LAZO CERRADO CON CONTROL PREDICTIVO HÍBRIDO. ....	20
FIGURA 2. ESQUEMA ÁRBOL COMPLETO PARA ALGORITMOS BRANCH AND BOUND. ....	28
FIGURA 3. ESQUEMA SOLUCIÓN PROBLEMA MINLP. ....	31
FIGURA 4. ESQUEMA SOLUCIÓN PROBLEMA MINLP USANDO ALGORITMOS EVOLUTIVOS. ....	33
FIGURA 5. DIAGRAMA DEL ALGORITMO GENÉTICO SIMPLE. ....	34
FIGURA 6. ESTANQUE DE AGUA CON CALEFACTORES ON/OFF. LABORATORIO DE ELECTROTECNOLOGÍAS UNIVERSIDAD DE CHILE. ....	35
FIGURA 7. EVOLUCIÓN DE LA FUNCIÓN OBJETIVO. ....	38
FIGURA 8. DETALLE DE LA EVOLUCIÓN DE LA FUNCIÓN OBJETIVO. ....	39
FIGURA 9. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL. ....	39
FIGURA 10. PRUEBAS POR SIMULACIÓN. RESPUESTA DE LA TEMPERATURA EN EL ESTANQUE. ....	40
FIGURA 11. PRUEBAS POR SIMULACIÓN. DETALLE DE LA RESPUESTA DE LA TEMPERATURA EN EL ESTANQUE. ....	41
FIGURA 12. PRUEBAS POR SIMULACIÓN. POTENCIA TOTAL ENTREGADA POR LOS CALEFACTORES. ....	41
FIGURA 13. RESULTADOS EXPERIMENTALES CON DISTINTAS ESTRATEGIAS HPC: VARIABLE MANIPULADA. ....	44
FIGURA 14. RESULTADOS EXPERIMENTALES CON DISTINTAS ESTRATEGIAS HPC: VARIABLE CONTROLADA. ....	44
FIGURA 15. SISTEMA DE ESTANQUES HÍBRIDO. ....	47
FIGURA 16. SEÑAL DE EXCITACIÓN DE ENTRADA. ....	47
FIGURA 17. SEÑAL DE SALIDA $H_j$ . ....	48
FIGURA 18. ANALISIS DE VECTORES PROPIOS Y FUNCIONES DE PERTENENCIA. ....	50
FIGURA 19. VALIDACIÓN DEL MODELO DIFUSO. ....	50
FIGURA 20. EVOLUCIÓN DE LA FUNCIÓN OBJETIVO. ....	56
FIGURA 21. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL. ....	56
FIGURA 22. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE CONTROLADA. ....	58
FIGURA 23. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE MANIPULADA. ....	58
FIGURA 24. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE CONTROLADA. DETALLE DE FIG. 22. ....	59
FIGURA 25. PRUEBA POR SIMULACIÓN. RESPUESTA DE LA VARIABLE MANIPULADA. DETALLE DE FIG. 23. ....	59
FIGURA 26. RUTA TÍPICA DE UN VEHÍCULO EN EL INSTANTE K. ....	69
FIGURA 27. DIAGRAMA DE BLOQUES COMPLETO PARA EL ENFOQUE HÍBRIDO PREDICTIVO PARA EL CALCULO DE RETARDOS OCASIONADOS POR RE-RUTEOS. ....	71
FIGURA 28. EJEMPLO DE SECUENCIA DEL VEHÍCULO 1 Y VARIABLES DE ESTADO ASOCIADAS. ....	75
FIGURA 29. LLAMADAS POTENCIALES. ....	87
FIGURA 30. REQUERIMIENTO PARA UN VEHICULO EN UN CIERTO PERIODO DE TIEMPO. ....	90
FIGURA 31. COORDENADAS PICKUP-DELIVERY DE LA DEMANDA HISTORICA EN UN CIERTO PERIODO. ....	91
FIGURA 32. CENTROS DE CLUSTER PARA UN NÚMERO DE 2 Y 3 CLUSTERS. ....	91
FIGURA 33. GRADO DE PERTENENCIA A CADA CLUSTER DE LA DATA HISTORICA. ....	92
FIGURA 34. MATRIZ G CASO NO SWAPPING. ....	95
FIGURA 35. DIAGRAMA DE BLOQUES COMPLETO DE HAPC PARA EL DPDP. ....	100
FIGURA 36. TIEMPO COMPUTACIONAL Y EVOLUCIÓN DE LA FUNCIÓN OBJETIVO. ....	102
FIGURA 37. EVOLUCIÓN DEL TIEMPO COMPUTACIONAL Y LA FUNCIÓN OBJETIVO. ....	103
FIGURA 38. TIEMPO COMPUTACIONAL Y EVOLUCION DE LA FUNCIÓN OBJETIVO. ....	104
FIGURA 39. PATRONES ORIGEN-DESTINO. ....	106
FIGURA 40. GRADO DE PERTENENCIA DE LOS REQUERIMIENTOS. ....	107



FIGURA 41. ANÁLISIS DE SENSIBILIDAD PARA $\tau$ (ZONIFICACIÓN CLASICA Y DIFUSA).....	108
FIGURA 42. PATRONES ORIGEN-DESTINO (9 ZONAS) P \ D: ZONAS PICKUP Y DELIVERY. ....	112
FIGURA 43. PATRONES ORIGEN-DESTINO (4 ZONAS) P \ D: ZONAS PICKUP Y DELIVERY. ....	113
FIGURA 44. ANÁLISIS DE SENSIBILIDAD PARA $\tau$ (PARTICIONES DE 9 Y 4 ZONAS).....	114
FIGURA 45. TIEMPO DE ESPERA, TIEMPO DE VIAJE Y TIEMPO TOTAL POR PASAJERO PARA UNA REALIZACIÓN (4 ZONAS).....	117

## **ANEXO: PUBLICACIONES GENERADAS**

D. Sáez, A. Núñez, A. Chong, R. Zúñiga, V. Albornoz, “Control híbrido predictivo aplicado al control de temperatura de un estanque”, Actas del XVI Congreso de la Asociación Chilena de Control Automático 2004, Santiago, Chile, 2-4 Noviembre 2004, páginas 61-67.

A. Núñez, D. Sáez, C. E. Cortés, “Problema de ruteo dinámico de una flota de vehículos con un enfoque de control predictivo híbrido basado en algoritmos genéticos”, Actas del XII Congreso Chileno de transporte, Valparaíso, Chile, 17-21 Octubre 2005, ISSN: 0717-3482, páginas 361-373.

A. Núñez, D. Sáez, S. Oblak and I. Skrjanc, “Hybrid Predictive Control based on Fuzzy Model”. IEEE World Congress on Computational Intelligence, FUZZ IEEE 2006, Vancouver, BC, Canadá. 16-21 Julio.

D. Sáez, C. E. Cortés, A. Núñez, “Hybrid predictive control based on Genetic Algorithm and Fuzzy Clustering for a real-time routed transit system”. Volumen especial de la revista ISI Computer & Operations Research (Aceptada).

C. E. Cortés, D. Sáez, A. Núñez, F. Murcia, “Hybrid predictive control for a real-time routed transit system”. En segunda revisión para la revista ISI Transportation Science.

**ANEXO 1: DISEÑO DE ESTRATEGIAS DE CONTROL PREDICTIVO HIBRIDO Y SU APLICACIÓN AL CONTROL DE TEMPERATURA DE UN ESTANQUE DE NIVEL**

D. Sáez, A. Nuñez, A. Chong, R. Zúñiga y V. Albornoz\*

Departamento de Ingeniería Eléctrica, Universidad de Chile  
Casilla 412-3, Santiago, Chile. Fax: 56-2-6720162 Email: [dsaez@ing.uchile.cl](mailto:dsaez@ing.uchile.cl)

\*Departamento de Industrias, Universidad Técnica Federico Santa María  
Av. Santa Maria 6400. Santiago. Chile.

Fax: 56-2-3531328 E-mail: [victor.albornoz@usm.cl](mailto:victor.albornoz@usm.cl)

Resumen: En este trabajo, se presenta la formulación del problema de control predictivo con variables manipuladas enteras. Para este problema, se analizan e implementan tres alternativas de solución: modulación por ancho de pulso, optimización basada en enumeración explícita y optimización basada en el método de Branch & Bound. Los algoritmos de solución son aplicados al control de temperatura de un estanque de nivel.

## Introducción

En la actualidad, la mayoría de los diseños de control se basan en la caracterización de los procesos por variables continuas. Sin embargo, los procesos reales se caracterizan por tener tanto variables continuas como variables enteras. Tradicionalmente para resolver el problema se han planteado soluciones como modulación por ancho de pulso o control on-off.

En esencia todos los procesos contienen componentes continuas y discretas dados por ejemplo por válvulas on/off, switches, overrides lógicos, etc. Estos sistemas híbridos pueden ser definidos como sistemas jerárquicos constituidos por componentes continuas y lógicas y/o discretas (Bemporad & Morari, 1999). La tendencia actual en control de procesos industriales es generar nuevas herramientas para el diseño de control para sistemas híbridos.

Recientemente, los sistemas híbridos han recibido mucha atención desde las ciencias de la computación y la comunidad de control, sin embargo aun no existe una metodología de diseño general para sistemas híbridos (Bemporad *et al.*, 1999, Bemporad *et al.*, 2002a).

Slupphaung *et al.* (1997) y Slupphaung & Foss (1997) presentan el diseño de control predictivo con variables de entrada continuas y enteras. En este caso, se resuelve el problema utilizando programación no lineal entera mixta y se evalúa favorablemente frente a una estrategia de control predictivo con separación de las variables continuas de las variables enteras. En este caso, se aplica por simulación al control de nivel y temperatura de un sistema de estanques.

Bemporad & Morari (1999, 2000) presentan un esquema de control predictivo para sistemas híbridos incluyendo restricciones operacionales. En este caso, el problema es resuelto utilizando programación cuadrática entera mixta. El algoritmo propuesto es aplicado por simulación a un sistema de suministro de gas, que considera variables manipuladas enteras.

Bemporad *et al.* (2002) establecen que un sistema híbrido con un controlador predictivo basado en una función objetivo cuadrática y restricciones lineales es una subclase de sistema híbrido MLD ("Mixed Logic Dynamical"). En otras palabras, el sistema en lazo cerrado está caracterizado como un sistema híbrido.

Todo el trabajo citado sobre control predictivo híbrido utiliza una función objetivo que no incluye la minimización del incremento de las variables manipuladas, es decir, el controlador no considera acción integral. Posiblemente esto se produce pues esta inclusión implica un mayor esfuerzo computacional dado por el aumento de variables de optimización. Por otra parte, en general, los algoritmos propuestos son sólo ejemplificados para sistemas de orden bajo y no se presentan comparaciones ni cualitativas ni cuantitativas con otras soluciones convencionales para sistemas híbridos.

Por lo tanto, en este trabajo se propone el análisis y desarrollo de algoritmos de optimización para el problema de control predictivo híbrido con variables manipuladas enteras. Además, se considera su evaluación cuantitativa y cualitativa mediante simulaciones para el control de temperatura a través de tres calefactores on-off para un estanque de nivel experimental del Laboratorio de Electrotecnologías de la Universidad de Chile.

## DISEÑO DE CONTROL PREDICTIVO HÍBRIDO

### 2.1 Formulación del problema

Los sistemas híbridos se pueden denominar como sistemas dinámicos lógicos mixtos (MLD) que describen estas componentes híbridas como un sistema de ecuaciones lineales sujeto a inecuaciones que involucran variables binarias y continuas. Estas variables pueden ser estados y/o variables de entrada del sistema.

Los sistemas híbridos MLD se pueden definir en general como (Bemporad & Morari, 1999):

$$\begin{aligned}x(t+1) &= A_{1t}x(t) + B_{1t}u(t) + B_{2t}\delta(t) + B_{3t}z(t) \\y(t) &= C_1x(t) + D_{1t}u(t) + D_{2t}\delta(t) + D_{3t}z(t) \\E_{2t}\delta(t) + E_{3t}z(t) &\leq E_{1t}u(t) + E_{4t}x(t) + E_{5t}\end{aligned}\quad (1)$$

donde  $x = [x_c, x_i]$  son las variables estado continuas ( $x_c$ ) y binarias ( $x_i$ : 0-1);  $y = [y_c, y_i]$  es vector de salida mixto y  $u = [u_c, u_i]$  es el vector de variables manipuladas mixta. Además,  $\delta$  y  $z$  son variables auxiliares binarias y continuas respectivamente. Las variables binarias permiten representar cualquier valor discreto.

En este trabajo, se considerarán sólo las variables manipuladas híbridas o enteras.

## 2.2 Control predictivo con restricciones

El controlador predictivo generalizado (GPC: “Generalized Predictive Controller”) propuesto por Clarke (1987) se diseña considerando un modelo ARIX del proceso (“Autoregressive Integrated with exogenous variable”), dado por:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{e(t)}{\Delta} \quad (2)$$

donde  $A(q^{-1})$  y  $B(q^{-1})$  son polinomios,  $\Delta = 1 - q^{-1}$ ,  $y(t)$  es la variable controlada,  $u(t)$  es la variable manipulada y  $e(t)$  es ruido blanco con esperanza igual a cero.

La salida predicha a  $j$  pasos, considerando las señales de salida hasta  $t$  y las acciones de control futuras, está dada por:

$$y_p(t+j) = G_j \Delta u(t+j-1) + F_j y(t) \quad (3)$$

con  $G_j = E_j(q^{-1})B(q^{-1})$ ,  $E_j$  y  $F_j$  polinomios únicos definidos por el polinomio  $A$  y el intervalo de predicción  $j$ .

El algoritmo de Control Predictivo Generalizado (GPC) calcula las acciones de control futuras, minimizando la siguiente función de costos:

$$J = J_1 + \lambda J_2 \quad (4)$$

con

$$J_1 = \sum_{j=N_1}^{N_2} (\hat{y}(t+j) - r(t+j))^2$$

$$J_2 = \sum_{j=N_1}^{N_u} \Delta u(t+j-1)^2$$

donde  $N_1$  y  $N_2$  corresponden a los horizontes de predicción,  $N_u$  es el horizonte de control,  $\hat{y}(t+j)$  es la predicción a  $j$  pasos para la variable controlada,  $r(t+j)$  es la referencia futura y  $\lambda$  es un factor de peso.

En la práctica, se requiere mantener las variables del proceso en rangos que aseguren el buen comportamiento de los equipos y evitar situaciones críticas. Por ejemplo, los actuadores tienen restricciones de límite y velocidad. Además, los puntos de operación del proceso

están determinados por objetivos económicos, que usualmente llevan al proceso a operar cerca de las restricciones. Los sistemas de control, en especial, los sistemas de control predictivo se anticipan a estas restricciones y corrigen las acciones de control.

Las restricciones típicas son de rango en la variable manipulada, en la variación de la variable manipulada o en el rango de las variables controladas. Es decir:

$$u_{\min} \leq u(t+i-1) \leq u_{\max}$$

$$\Delta u_{\min} \leq \Delta u(t+i-1) \leq \Delta u_{\max} \quad (5)$$

$$y_{\min} \leq \hat{y}(t+j) \leq y_{\max}$$

con  $j = N_1, \dots, N_2$  e  $i = N_1, \dots, N_u$ .

Finalmente, utilizando algoritmos de programación cuadrática se minimiza la función objetivo (4), sujeta a las restricciones dadas por las predicciones (3) y las restricciones definidas en (5) (Tsang & Clarke, 1988).

## 2.3 Control predictivo híbrido

En este caso, se considera la minimización de la misma función de costos definida en la ecuación (4), considerando que las variables manipuladas y/o variables de estado tienen que ser enteras. Este problema de optimización es difícil de resolver analíticamente y por lo tanto, se consideran diversas alternativas de solución numérica, que a continuación se detallan.

### ALGORITMOS DE SOLUCIÓN PARA CONTROL PREDICTIVO HÍBRIDO

#### 3.1 Control predictivo con modulación por ancho de pulso

En este caso, se considera resolver el problema de control predictivo con variables manipuladas enteras, resolviendo el problema de control predictivo con restricciones y a continuación aplicar una señal con modulación de ancho de pulso (PWM: “Pulse Wide Modulation”) a la planta a controlar.

En este caso, el algoritmo de modulación por ancho de pulso consiste en que la señal continua entregada por el controlador predictivo con restricciones descrito en la sección 2.2, modulará la señal de on-off durante cada período de muestreo hasta que el promedio de la señal modulada alcance ese valor de la acción de control

óptimo dado por el algoritmo de control predictivo con restricciones.

Según el proceso que se module se retiene la señal de entrada durante un cierto período de tiempo y según la capacidad de respuesta de los actuadores se tendrá un periodo de modulación que se repetirá durante todo el período de tiempo en que se retuvo la señal de entrada.

El problema de la modulación por ancho de pulso convencional es el desgaste de los actuadores debido a posibles altas frecuencias de las variables manipuladas a modular.

### 3.2 Control predictivo híbrido con enumeración explícita

Para solucionar el problema de control predictivo con variables manipuladas enteras y en un espacio finito de posibilidades, se considera el método de optimización basada en enumeración explícita.

En este caso, se generan todas las combinaciones de entradas posibles a aplicar en el horizonte de predicción y luego por simple evaluación de la función objetivo se encuentra la acción de control que minimiza el problema. Este método es factible siempre y cuando el horizonte de predicción sea reducido; pues a medida que este aumenta, se incrementa exponencialmente el número de combinaciones de entradas posibles a realizar y por ende aumenta el número de soluciones factibles de los modelos resultantes.

En un problema complejo con varias variables de entrada y/o un alto horizonte de predicción, un procesador no sería capaz de enumerar todas las soluciones posibles o simplemente encontraría el óptimo en un tiempo no razonable. Esto se debe al crecimiento exponencial del número de operaciones que se deben realizar en la medida que se aumentan los estados o el horizonte de predicción.

### 3.3 Control predictivo híbrido con Branch & Bound.

El problema de control predictivo híbrido se resuelve utilizando el método de optimización Branch & Bound para problemas de programación mixta entera (Floudos, 1995, Wolsey, 1998).

En este método, se aplican reglas para evitar enumerar todas las combinaciones posibles y se basa en resolver problemas continuos que buscan

encontrar soluciones enteras mejores a la mejor solución factible que se haya localizado hasta el momento, explorando todo el dominio del problema original, si es necesario. Esto permite ahorrar un apreciado esfuerzo computacional.

El método Branch & Bound se puede resumir como:

1. Resuelva el problema de optimización relajado con todas las variables continuas, es decir, sin considerar que las variables manipuladas son enteras (Problema P0). Si el resultado es que todas las variables tienen valores enteros, ésta es la solución óptima. En caso contrario se crean dos subproblemas. Se escoje la variable continua más fraccionaria ( $U(t+i1-1)$ ) y se construye el subproblema 1 (P1) agregando la restricción  $U(t+i1-1) \leq [U(t+i1-1)]$  y el subproblema 2 (P2) agregando la restricción  $U(t+i1-1) \geq [U(t+i1-1)]+1$ , donde el operador  $[ ]$  es la parte entera de la solución más fraccionaria elegida.
2. Seleccione el siguiente subproblema a resolver. Se resuelve el problema de optimización con restricciones. Si es que la función objetivo evaluada en el óptimo es mayor que el función objetivo evaluada en la mejor solución entera encontrada entonces se obtiene como salida del problema la mejor solución factible encontrada y se regresa al problema anterior. Al hacer esto se eliminan las posibles ramas que se generarían a partir de este problema. Si no se cumple lo anterior se escoge la variable más fraccionaria ( $U(t+ij-1)$ ) y se crean dos nuevos subproblemas agregando las restricciones  $U(t+ij-1) \leq [U(t+ij-1)]$  para el primer subproblema y agregando la restricción  $U(t+ij-1) \geq [U(t+ij-1)]+1$  para el segundo subproblema.
3. Si es que hay subproblemas por resolver se regresa al paso 2. Si no hay subproblemas por resolver entonces la solución óptima es la mejor entera que se haya encontrado hasta el momento. Si no hay soluciones el problema es infactible.

## CONTROL DE TEMPERATURA DE UN ESTANQUE DE NIVEL

### 4.1 Descripción del proceso

En la figura 2, se presentan un estanque de agua experimental con tres calefactores on-off y un mezclador situado en el Laboratorio de Electrotecnologías, Universidad de Chile.



Figura 2. Estanque con calefactores, Laboratorio Electrotecnologías.

Los tres calefactores on-off entregan una potencia máxima del orden de 2000 W cada uno. Por simplicidad se supondrá que los tres calefactores son iguales.

El objetivo de control es llevar la temperatura del agua en el estanque a una referencia dada, mediante el encendido y apagado de los tres calefactores, además de minimizar la energía de control o lo que es lo mismo minimizar el encendido y apagado de los calefactores. Por lo tanto, se considera las variables manipuladas binarias (encendido o apagado de los calefactores) y la variable controlada continua (temperatura).

En este trabajo, se considera el siguiente modelo de primer orden con retardo del estanque, cuyos parámetros han sido obtenidos en forma experimental:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{K}{\tau s + 1} e^{-\tau_d s} \quad (6)$$

donde  $K = 0.0067$  es la ganancia estática,  $\tau = 598.75$  seg. es la constante de tiempo y  $\tau_d = 2$  es el retardo del proceso.

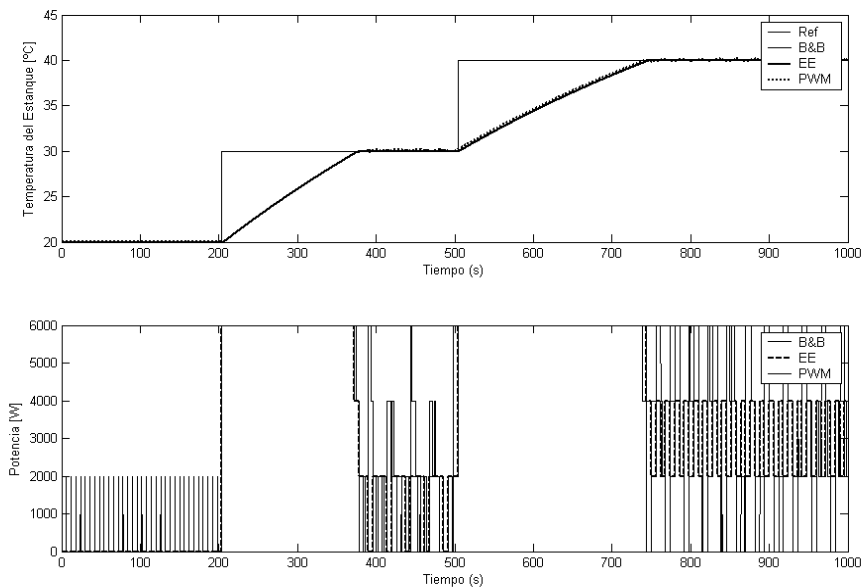


Figura 3. Respuesta del sistema.

#### 4.2 Bases de evaluación

En este trabajo, se considera el siguiente modelo ARIX del proceso, obtenido de la discretización de la ecuación (6) con un periodo de muestreo igual a 6 seg.:

$$\begin{aligned} & (1 - 0.99q^{-1})y(t) \\ & = (4.448q^{-1} + 2.213q^{-2}) \times 10^{-5} u(t) + \frac{e(t)}{\Delta} \quad (7) \end{aligned}$$

donde  $e(t)$  es una señal de error de tipo ruido blanco con esperanza igual a cero.

Para el diseño de las estrategias de control descritas en el capítulo III, se consideran los siguientes parámetros de la función objetivo definida en la ecuación (4):  $N_1 = 1$ ,  $N_2 = 5$  y  $N_u = 5$  y  $\lambda = 0.001$ . Para las pruebas por simulación, se evaluará el promedio de esta función objetivo.

$$0 \leq u(t+i-1) \leq 3 \quad (8)$$

con  $i = 1 \dots N_u$

Estas restricciones se deben a que la cantidad máxima de calefactores a encender es 3.

#### 4.3 Control predictivo con PWM

En primer lugar, se diseña un controlador predictivo con restricciones según lo descrito en la sección 2.2. En este caso, se minimiza la función objetivo definida en (4) con el modelo ARIX descrito en la ecuación (7), sujeto a las restricciones de cotas:

Como la planta tiene como entradas la señal de encendido o apagado de los tres calefactores, es necesario utilizar un PWM después del controlador predictivo GPC con restricciones para transformar la señal continua en una señal entera.

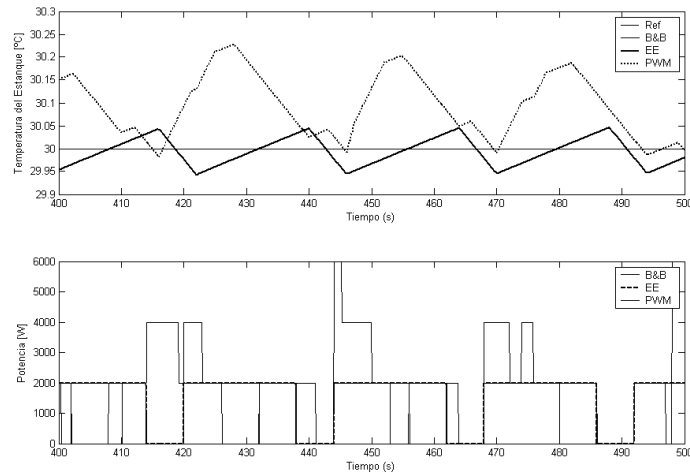


Figura 4. Detalle de la respuesta del sistema entre 400 y 500 segundos.

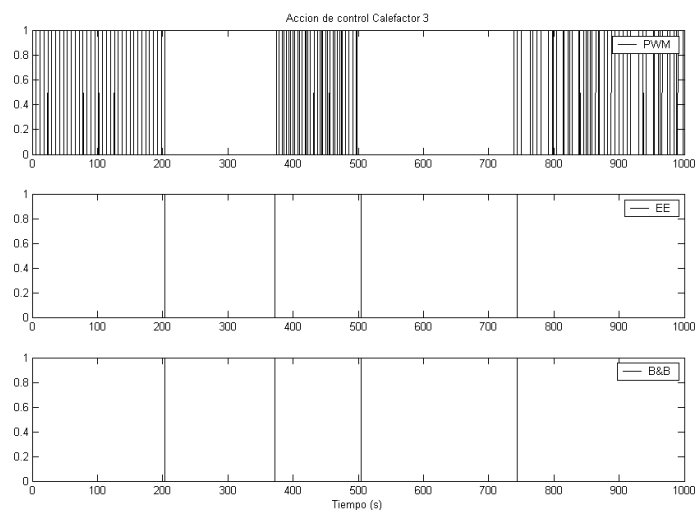


Figura 5. Secuencias de encendido y apagado. Calefactor 3.



#### 4.4 Control predictivo híbrido con enumeración explícita

El controlador predictivo con enumeración explícita permite trabajar con señales continuas y discretas.

El algoritmo utilizado por el controlador puede predecir las salidas futuras del sistema, basado en la información actual, pasada y en las acciones de control futuras. Se re-escribe la función objetivo a minimizar en función sólo de las acciones de control futura y se evalúa con todas las posibles combinaciones. Se escoge como vector de salida a aquella combinación de acciones de control con menor valor de la función objetivo. En la etapa final del algoritmo, se obtiene un número entre 0 y 3 (señal entera), que corresponde a la cantidad de calefactores que se deben encender durante el siguiente periodo de muestreo. Este número se traduce en tres señales de control, una para cada calefactor, y determinan el estado en que se debe encontrar cada uno.

#### 4.5 Control predictivo híbrido con Branch & Bound

El controlador predictivo con Branch & Bound minimiza la función objetivo aplicando el método descrito en 2.3. En la etapa final del algoritmo, se obtiene un número entre 0 y 3 (señal entera), que corresponde a la cantidad de calefactores que se deben encender durante el siguiente periodo de muestreo. Este número se traduce en tres señales de control, una para cada calefactor.

#### 4.6 Análisis comparativo

En la Figura 3 se presentan la respuesta del sistema y la variable manipulada con el controlador predictivo con restricciones y PWM, el controlador predictivo híbrido basado en enumeración explícita y el controlador predictivo híbrido basado en el método de Branch & Bound. En la figura 4 se presenta un detalle de la figura 3 para el tiempo de simulación entre 400 y 500 segundos.

Se aprecia que la temperatura y la acción de control, dada por la potencia total de los tres calefactores, son las mismas con los algoritmos de enumeración explícita y Branch & Bound pues se resuelve el mismo problema de optimización con variables enteras.

La temperatura obtenida con el controlador GPC con restricciones y PWM, según la figura 4, entrega una respuesta con más sobreoscilación y mayor error en estado estacionario. Esto se debe a que la acción de control resultante es más oscilatoria debido a la conversión de la solución continua entregada por el GPC con restricciones a un valor discreto a través del PWM.

A modo de ejemplo del funcionamiento de los calefactores, en la figura 5, se presenta la secuencia de encendido y apagado para el calefactor 3. Se observan mayores cambios con el controlador GPC con restricciones y PWM.

En la tabla 1 se presentan los valores medios para la función objetivo definida en la ecuación (4) y cada una de sus componentes, considerando las diferentes estrategias de control propuestas. Se aprecia que los valores medios son iguales con los controladores predictivos con enumeración explícita y Branch & Bound, lo cual es evidente pues se resuelve el mismo problema de optimización con variables enteras. Además, se observa una disminución significativa con estos algoritmos en comparación con el controlador GPC con restricciones y PWM en el valor medio de  $J_2$ , que representa la energía de control.

Tabla1. Función objetivo

Estrategia de control predictiva	$J_1$	$J_2$	$J = J_1 + \lambda J_2$
Con restricciones y PWM	66.92	14.07	66.937
Enumeración explícita	65.01	2.03	65.013
Branch & Bound	65.01	2.03	65.013

## V. CONCLUSIONES

En este trabajo, se presentan tres alternativas de solución para un problema de control predictivo híbrido dado por las variables manipuladas enteras.

En particular, el método de enumeración explícita implica un mayor esfuerzo computacional dado que en cada instante se evalúan todas las posibles soluciones del problema. El método Branch & Bound permite obtener el mismo resultado en forma más eficiente, considerando una búsqueda inteligente de la solución. Esta ventaja se reflejaría significativamente para un problema con varias

entradas enteras y/o un horizonte de predicción muy largo.

De esta manera, la incorporación de características híbridas de un sistema en su formulación permite aproximar de mejor manera el funcionamiento del proceso. Esta mejora puede ser aprovechada por las técnicas de control predictivo que minimizan una función objetivo.

Por su parte, las saturaciones o métodos como PWM que aproximan a un entero las señales de control continuas no incorporan las características híbridas propias del sistema, y por lo tanto no logran alcanzar óptimos globales de funcionamiento.

Actualmente, se está trabajando en la implementación on-line de los algoritmos propuestos. Además, se considera continuar con el desarrollo de algoritmos de optimización mixta entera que permitan mejorar la rapidez de la búsqueda de soluciones.

#### **AGRADECIMIENTOS**

Los autores agradecen al proyecto FONDECYT N° 1040698 “Sistemas de control predictivo híbrido con variables continuas y cuantizadas” y y al Dr. Carlos Muñoz por sus comentarios y observaciones. El coautor Dr. Albornoz agradece al proyecto USM 28.04.21 “Técnicas de descomposición en modelos de planificación de dos etapas”, financiado por la Dirección General

de Investigación y Postgrado, Universidad Técnica Federico Santa María.

#### **REFERENCIAS**

- Bemporad A., Heemels W., De Schutter B. (2002). On hybrid systems and closed-loop MPC systems IEEE Transaction on Automatic Control, Vol. 47, N°5, pp. 863-869.
- Bemporad, A., Borrelli, F., Morari, M. (2002a). On the optimal control law for linear discrete time hybrid systems. Proc. 5th International Workshop, Hybrid Systems: Computation and Control, 25 - 27 March 2002, Stanford, California, USA, Tomlin C. J. and Greenstreet M. R. (eds.), Lecture Notes in Computer Science, no. 2289, pp. 105 - 119, Springer Verlag.
- Bemporad, A., Morari, M. (1999). Control of systems integrating logic, dynamics and constraints. Automatica, Vol. 35, pp. 407-427.
- Bemporad, A., Morari, M. (2000). Predictive control of constrained hybrid systems. Non linear model predictive control. Nonlinear Model Predictive Control. Ed. Allogöwer, F., Zheng, A. Birkhäuser Verlag, Basel, pp. 71-98.
- Clarke, D., Mohtadi, C., Tuffs, P., (1987). Generalized predictive control (Parts I and II) . Automatica, Vol. 23, N°2, pp. 137-160.
- Floudas, C. (1995). Nonlinear and mixed-integer optimization. Oxford University Press, Inc.
- Slupphaug, O., Foss, B. (1997). Model predictive control for a class of hybrid systems. Proceeding of European Control Conference, Brussels, Belgium.
- Slupphaug, O., Vada, J., Foss, B. (1997). MPC in systems with continuous and discrete control inputs. Proceedings of American Control Conference, Albuquerque, NM, USA.
- Tsang, T., Clarke, D. (1988). Generalised predictive control with input constraints. IEE Proceedings, Vol. 135-D, N° 6, pp. 451-460.
- Wolsey, L. Integer Programming. John Wiley & Sons.

## **ANEXO 2: PROBLEMA DE RUTEO DINÁMICO DE UNA FLOTA DE VEHÍCULOS CON UN ENFOQUE DE CONTROL PREDICTIVO HÍBRIDO BASADO EN ALGORITMOS GENÉTICOS.**

Alfredo Núñez V.

Departamento de Ingeniería Eléctrica  
Universidad de Chile, Santiago, Chile  
Fono: (56-2) 678 4207, Fax: (56-2) 695 3881  
**e-mail: [alfnunez@ing.uchile.cl](mailto:alfnunez@ing.uchile.cl)**

Doris Sáez H.

Departamento de Ingeniería Eléctrica  
Universidad de Chile, Santiago, Chile  
Fono: (56-2) 678 4207, Fax: (56-2) 695 3881  
**e-mail: [dsaez@ing.uchile.cl](mailto:dsaez@ing.uchile.cl)**

Cristián E. Cortés

Departamento de Ingeniería Civil  
Universidad de Chile, Santiago, Chile.  
Fono: (56-2) 678 4380, Fax: (56-2) 689 4206  
**e-mail : [ccortes@ing.uchile.cl](mailto:ccortes@ing.uchile.cl)**

### **R E S U M E N**

En este trabajo se presenta una solución basada en algoritmos genéticos para el problema de ruteo dinámico de una flota de vehículos que realiza un servicio de traslado de pasajeros puerta a puerta. Este problema es formulado como un problema de Control Predictivo Híbrido basado en variables de estado para el sistema flota-ciudad-clientes. El tiempo y carga son variables de estado, la secuencia de los vehículos corresponde a la variable de entrada o manipulada y las solicitudes por servicio son perturbaciones.

Para resolver el problema de ruteo dinámico se plantea un controlador predictivo híbrido que asigna las rutas a los vehículos minimizando una función de costos esperada a medida que ocurren solicitudes por servicio en cualquier instante de tiempo. Cuando ocurre un nuevo requerimiento, el controlador ocupa como información las secuencias (rutas) anteriores, las variables de estado y un modelo de predicción de llamadas. Debido al elevado tiempo computacional que toma encontrar la solución óptima global se realiza la optimización en un sub-espacio donde las soluciones satisfacen la restricción de mantener el orden de las rutas preestablecidas para cada vehículo. Como algoritmo de optimización se propone algoritmos genéticos que entregan soluciones cercanas al óptimo dado por enumeración explícita, pero en un tiempo computacional reducido.

## 1. INTRODUCCION

El problema de ruteo de vehículos de servicio de pasajeros puerta a puerta, ha sido formulado y resuelto en la literatura usando tanto algoritmos exactos como heurísticas (Savelsbergh y Sol, 1995). El problema se torna bastante más complejo cuando se trabaja con demanda dinámica que entra al sistema en tiempo real, y en ese caso, la mayoría de las soluciones se basan en algoritmos aproximados (Psaraftis, 1998). Cortés y Jayakrishnan (2004) y Cortés *et al.* (2005) reconocen una complejidad adicional en este problema, subdimensionada en la literatura tradicional, en el sentido de que es necesario incorporar en las funciones de costo el hecho de tener la posibilidad de insertar nuevos requerimientos de servicio en las rutas previamente establecidas, pues esto provoca un costo adicional sobre los pasajeros en servicio. Cortés *et al.* (2005) proponen una formulación del tipo Control Predictivo Híbrido para internalizar estos costos futuros y muestran que incorporar un buen modelo predictor de demanda reduce significativamente los costos reales observados en el sistema completo. Sin embargo, como algoritmo de solución los autores utilizan enumeración explícita, lo cual es bastante ineficiente considerando problemas en tiempo real.

En la literatura especializada en ruteo de vehículos en los últimos años, muchos autores como Bean y Costa (2005), Ceylan y Bell (2004), Le Bouthillier y Crainic (2005), Osman *et al.* (2005), y Mester y Bräysy (2005) han probado la eficiencia de los algoritmos genéticos en encontrar buenas soluciones en problemas similares de ruteo y en un tiempo reducido. Por otro lado, autores como Li *et al.* (2005) han debido utilizar algoritmos heurísticos para resolver problemas de ruteo de varios vehículos debido a que es un problema de optimización del tipo *NP-hard*, obteniendo buenos resultados computacionales. En este trabajo, la incorporación de control predictivo híbrido pretende resolver un problema bastante más complejo que un típico problema de ruteo, al incorporar el elemento predictivo en la toma de decisiones, a través de una formulación de control del tipo híbrido, involucrando tanto variables continuas como enteras.

En la actualidad, la comunidad de control automático ha mostrado un gran interés en sistemas híbridos, que son sistemas que involucran variables continuas, binarias y/o enteras, ya sea en su estado o en la variable manipulada. El Control Predictivo Híbrido (HPC) ha mostrado ser muy útil para el control de sistemas híbridos y autores como Bemporad y Morari (1999), Bemporad *et al.* (2002), y Thomas *et al.* (2004), entre otros, han reportado excelentes resultados por simulación en comparación con otras técnicas de control más simplificadas. Las estrategias de HPC más usadas involucran dos algoritmos de optimización: Enumeración Explícita (EE) y Branch and Bound (BB). Ambos permiten resolver problemas de optimización entera-mixta (MIOP) (Floudos, 1995), pero el elevado esfuerzo computacional, especialmente en el caso de EE, hace que sean ineficientes para resolver problemas en tiempo real. Como solución a esto se han propuesto Algoritmos Genéticos (GA) como herramienta eficiente para resolver MIOP.

En este trabajo se presenta el enfoque analítico del problema original descrito por Cortés *et al.* (2005) para el ruteo dinámico de vehículos, con información histórica agregando capacidad predictiva al modelo, y se investiga la incorporación de GA para resolver el problema en forma más eficiente que con EE. Se muestran ventajas en términos de flexibilidad en la formulación y en la posibilidad de obtener soluciones más genéricas al utilizar este nuevo esquema de solución.

## 2. FORMULACIÓN DEL PROBLEMA

### 2.1 Generalidades

El problema está descrito por una flota de vehículos que debe satisfacer óptimamente la demanda por servicio de pasajeros puerta a puerta dentro de una red de transporte. El servicio consiste en recoger y entregar pasajeros en ciertas coordenadas dadas por su lugar de origen (*recogida*) y su destino (*entrega*). El ruteo debe ser automático y rápido debido a que aleatoriamente ocurren nuevas solicitudes por servicio que deben ser satisfechas mientras los vehículos están en movimiento. El ruteo se hace en base a la minimización de una función objetivo que depende de las variables de estado de los vehículos. Estas variables de estado incluyen todas las características relevantes de los vehículos, y son los tiempos estimados de llegada y el vector de carga entre paradas,  $T_j^i(k)$  y  $L_j^i(k)$  para el vehículo  $j$ , parada  $i$  e instante de tiempo  $k$ . Se cuenta además con un registro histórico de las coordenadas de *recogidas* y *entregas* de las llamadas ocurridas hasta la fecha en diversas horas del día el cual es utilizado para tomar mejores decisiones futuras.

### 2.2 Formulación analítica: modelación dinámica con restricciones

Se considera una formulación en variables de estado para el sistema flota-ciudad-clientes, donde el vector de tiempo y de carga entre destinos son las variables de estado, la secuencia de cada vehículo corresponde a la variable manipulada y las llamadas que solicitan servicio son otras variables de entrada no manipuladas. Cada vehículo  $j$  en un instante de tiempo  $k$ , tendrá una secuencia  $S_j(k)$ , un vector de tiempo  $T_j(k)$  y un vector de carga  $L_j(k)$  que contendrán toda la información de cada trayecto que une las  $w_j(k)$  paradas asignadas al vehículo  $j$  en el orden que establece la secuencia respectiva. Analíticamente:

$$S_j(k) = \begin{bmatrix} S_{11} & S_{12} & P_1 & label_1 \\ S_{21} & S_{22} & P_2 & label_2 \\ \vdots & \vdots & \vdots & \vdots \\ S_{w_j(k)1} & S_{w_j(k)2} & P_{w_j(k)} & label_{w_j(k)} \end{bmatrix}, T_j(k) = \begin{bmatrix} T_j^1 \\ T_j^2 \\ \vdots \\ T_j^{w_j(k)} \end{bmatrix}, L_j(k) = \begin{bmatrix} L_j^1 \\ L_j^2 \\ \vdots \\ L_j^{w_j(k)} \end{bmatrix} \quad (1)$$

donde  $S_{i1}$  y  $S_{i2}$  indican si la parada  $i$  es una *recogida* o una *entrega*,  $P_i$  es la coordenada de la parada planificada  $i$  en la que se entrega o se toma al pasajero *label* $i$ .  $T_j^i$  y  $L_j^i$  indican la hora estimada de arribo a la parada  $i$  y la carga del vehículo al dejar esa parada. Es necesario señalar que se requiere conocer además las coordenadas actuales de los vehículos, sus cargas antes de llegar al primer destino y un reloj que sirva de referencia para estimar horas de llegada:  $P_0, L_j^0, T_j^0$ . El tiempo de viaje en áreas urbanas fluctúa debido a una gran variedad de factores, como por ejemplo accidentes, condiciones del tránsito y clima (Ceylan y Bell, 2004). Por simplicidad se ocupará una red conceptual y se considerará como estimador de tiempo distancias con norma euclidiana y velocidad de vehículos constante, lo cual se mejorará en trabajos posteriores. Notar que en este caso, la única fuente de aleatoriedad en los tiempos finales de viaje estará dada por las nuevas llamadas que aparecen en tiempo real, y que obliga a re-rutear algún vehículo. En la figura 1 se muestra un ejemplo de secuencia con información de tiempo y carga en cada tramo y la derecha  $S_I(k), T_I(k)$  y  $L_I(k)$  indicarán la secuencia, el vector de tiempo y la carga que el vehículo 1 tiene en el instante  $k$  en forma matricial. Se consideró velocidad igual a 1 unidad de medición y de esa forma las distancias son iguales a los tiempos esperados de viaje.

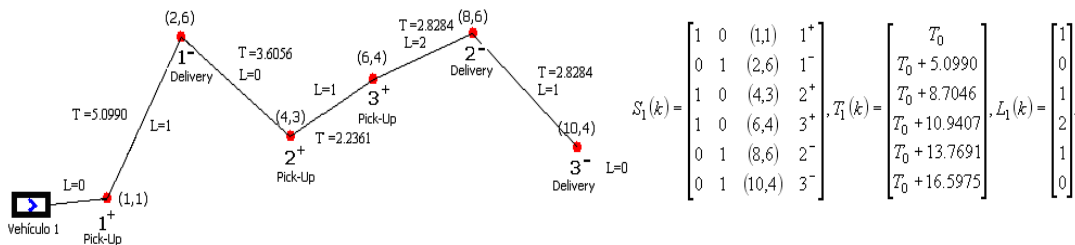


Figura 1. Ejemplo de secuencia gráfica para un vehículo y su notación en variables de estado.

Las restricciones del modelo son las siguientes:

**Restricción 1.-**  $S_{1i}$  y  $S_{2i}$  son variables binarias distintas pues una parada puede ser una *recogida* o una *entrega* pero nunca ambos.  $S_{i1}, S_{i2} \in \{0,1\}$ ,  $S_{i1} \neq S_{i2} \Leftrightarrow S_{i1} + S_{i2} = 1$

**Restricción 2.-** De precedencia. No puede ocurrir la *entrega* de un pasajero antes que su *recogida*. De acá se deduce que toda ruta acaba con una *entrega*:  $S_{n1} = 0, S_{n2} = 1$ .

**Restricción 3.-** Un destino  $P_i$  se debe recorrer una sola vez. Puede tomar un valor igual a la coordenada de algún destino dado por una lista de requerimientos. Un solo vehículo debe satisfacer el requerimiento tanto de *recoger* como de *entregar*.

**Restricción 4.-** Restricción de carga máxima. Un vehículo no podrá llevar en un trayecto más pasajeros que su carga posible máxima.  $L^i(k+1) \leq L_{\max}$ .

## 2.3 Controlador Predictivo Híbrido

### 2.3.1 Función objetivo

Para rutear la flota de vehículos se define como criterio la minimización de la siguiente función objetivo en el instante actual de tiempo  $k$  en el cual ocurrió el nuevo requerimiento por servicio:

$$J(S(k)) = \sum_{t=1}^{N_{pred}} \sum_{j=1}^{N_v} \sum_{h=1}^{h_{prob}(k+t)} p_h(k+t) \cdot \left( \Delta C_j(k+t) \Big|_{S_j(k+t-1),h} \right) \quad (2)$$

$$C_j(k+t) \Big|_{S_j(k+t-1),h} = \sum_{i=1}^{w_j(k+t)} \left\{ \underbrace{\left[ L_j^{h,i-1}(k+t+1) + 1 \right] \left( T_j^{h,i}(k+t+1) - T_j^{h,i-1}(k+t+1) \right)}_{J \text{ de tiempo de viaje}} + \underbrace{r_j^{h,i}(k+t+1) \alpha \left( T_j^{h,i}(k+t+1) - T_j^{h,0}(k+t+1) \right)}_{J \text{ de tiempo de espera}} \right\}$$

donde  $k+t$  es el instante en que ocurre el  $t$ -ésimo requerimiento después de ocurrido el requerimiento en el instante  $k$ ,  $N_{pred}$  es el horizonte de predicción,  $N_v$  el tamaño de la flota,  $h_{prob}(k+t)$  la cantidad de requerimientos probables en el instante  $k+t$ ,  $p_h(k+t)$  es la probabilidad de ocurrencia del requerimiento  $h$  en el instante  $k+t$ ,  $\Delta C_j(k+t) = C_j(k+t) - C_j(k+t-1)$ ,  $C_j(k+t) \Big|_{S_j(k+t-1),h}$  es la función de costos del vehículo  $j$  en el instante  $k+t$  dada una secuencia en el instante anterior  $S_j(k+t-1)$  y un nuevo requerimiento  $h$  que tiene probabilidad de ocurrencia  $p_h(k+t)$ ,  $w_j(k+t)$  es el número de destinos por recorrer del vehículo  $j$  en el instante  $k+t$ ,  $L_j^{h,i-1}(k+t+1)$  es la carga en el trayecto que une los destinos  $i-1$  con  $i$ ,  $T_j^i(k+t+1) - T_j^{i-1}(k+t+1)$  es el tiempo que demora el vehículo en ir del destino  $i-1$  al destino  $i$ ,  $r_j^{h,i}(k+t+1)$  es el factor de espera por servicio,  $\alpha$  es un ponderador del tiempo de espera respecto del tiempo de viaje (por simplicidad asumimos  $\alpha=1$ ),  $T_j^{h,i}(k+t+1) - T_j^{h,0}(k+t+1)$  es el tiempo que le falta al vehículo  $j$  para llegar desde su posición actual hasta el destino  $i$ .

El primer término de la función de costos ( $J$  de tiempo de viaje) considera el tiempo de viaje de cada trayecto ponderado por su carga más uno con el fin de considerar el costo de mover vehículos sin pasajeros. El segundo término ( $J$  de tiempo de espera) representa al tiempo que un pasajero espera antes de ser recogido. Es importante minimizar esta función de espera pues si el vehículo demora mucho tiempo en buscar a un cliente, es posible que este prefiera acceder a otro servicio de transporte (aunque al ocupar este se

demore más tiempo en llegar a su destino).  $r_j^{h,i}(k)$  se activa cuando la parada  $i$  es una *recogida*.

Por otro lado, la función objetivo requiere información de instantes futuros que se extraen de un estudio de demanda histórica. Se puede llegar a obtener que en una cierta hora del día hay diferentes “viajes tipo” con mayor probabilidad de ocurrencia que otros. Por ejemplo, en el horario punta mañana muchas solicitudes pueden ser de traslados de un sector residencial a uno empresarial o a un colegio, etc. El criterio a  $n$  pasos buscará la inserción óptima de una llamada actual de modo de minimizar los costos finales de todos los escenarios futuros posibles ponderados por la probabilidad de ocurrencia de cada secuencia probable de  $n$  llamadas futuras. Con el criterio a un paso, la expresión para  $J$  en (2) con  $N_{pred} = 1$ , queda como

$$J = \sum_{t=1}^1 \sum_{j=1}^{N_v} \sum_{h=1}^{h_{prob}(k+t)} p_h(k+t) \cdot \Delta C_j(k+t) \Big|_{S_j(k+t-1),h} \quad (3)$$

donde  $h_{prob}(k+1) = 1$  pues el nuevo requerimiento es uno y es conocido, por lo que tiene probabilidad 1. La función objetivo (3) se puede escribir como:

$$J = \sum_{j=1}^{N_v} \Delta C_j(k+1) \Big|_{S_j(k),1} \quad (4)$$

A un paso, la búsqueda del óptimo se basa sólo en la información de la secuencia anterior y la llamada actual. La decisión no incorpora información de las llamadas futuras más probables por lo que el ruteo es miope y puede ser costoso. Con el criterio a dos pasos, la función objetivo en (2) con  $N_{pred} = 2$  es como sigue:

$$J = \sum_{t=1}^2 \sum_{j=1}^{N_v} \sum_{h=1}^{h_{prob}(k+t)} p_h(k+t) \cdot \Delta C_j(k+t) \Big|_{S_j(k+t-1),h} =$$

$$\sum_{j=1}^{N_v} \left[ C_j(k+1) \Big|_{S_j(k-1),1} - C(k) + \sum_{h=1}^{h_{prob}(k+2)} p_h(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \sum_{h=1}^{h_{prob}(k+2)} p_h(k+2) \cdot C_j(k+1) \Big|_{S_j(k-1),1} \right] =$$

$$\sum_{j=1}^{N_v} \left[ \sum_{h=1}^{h_{prob}(k+2)} p_h(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \overbrace{C(k)}^{cte. conocida} \right] \quad (5)$$

En este caso, la optimización buscará la secuencia que incorpore la siguiente llamada, tal que se minimice la suma de los costos finales de todos los casos probables de llamadas futuras ponderados por la probabilidad de ocurrencia de cada caso. En total son  $h_{pred}(k+2)$  llamadas posibles con un probabilidad  $p_h(k+2)$  de que ocurra la llamada



h. Con el criterio a dos pasos no se minimiza el costo de la etapa intermedia  $C(k+1)$  en la cual sólo se insertó el nuevo requerimiento, sino más bien el costo del trayecto completo que incluye tanto la llamada en  $k+1$  como las llamadas más probables en  $k+2$ . En este trabajo se mostrarán resultados con la función objetivo a 2 pasos.

### 2.3.2 Planteamiento para resolver el problema de optimización

El problema de optimización consiste en encontrar las secuencias óptimas de los vehículos de la flota que minimizan la función objetivo. Se denotará por  $P_i$  a la parada  $i$  desconocida que debe ser calculada, mientras que  $f_i$  será un destino o coordenada conocida y puede ser una *recogida* o una *entrega* del pasajero  $label_i$ . Una parada desconocida  $P_i$  se puede escribir como una combinación lineal de todas las paradas conocidas utilizando los factores de activación binarias  $g_{jk}$  como se muestra:

$$P_i = g_{i1}f_1 + g_{i2}f_2 + \dots + g_{in}f_n \quad (6)$$

donde  $n = w_j(k)$ ,  $g_{ik} \in \{0,1\}$  y vale 0 si  $f_k$  no es la parada  $i$  y vale 1 si  $f_k$  es la parada  $i$ . El óptimo global del problema de ruteo se puede obtener escogiendo óptimamente la activación de los factores  $(g_{ik})_{\substack{i=1..n \\ k=1..n}}$  de un matriz G, para cada vehículo. De la restricción 3,

sección 3.1 resulta:

$$g_{i1} + g_{i2} + \dots + g_{in} = 1, \quad \forall i = 1, \dots, n \quad (7)$$

$$g_{1j} + g_{2j} + \dots + g_{nj} = 1, \quad \forall j = 1, \dots, n \quad (8)$$

Dentro de los conjuntos de destinos que son conocidos existe una subdivisión que es conocida y es cuales destinos son *recogidas* y cuales *entregas*. Será posible entonces ordenar los destinos según sean *recogidas* o *entregas*. Si se ocupa la matriz G como incógnita, en lugar de la secuencia, la carga  $i$  se puede expresar como la suma de la carga inicial más todas las activaciones de las *recogidas* anteriores menos las activaciones de todas las *entregas*. La restricción de carga (restricción 4, sección 3.1) en función de los factores de activación de la matriz G esta dada por:

$$L_j^{h,i-1}(k+1) = L_j^{h,0}(k) + \sum_{j=1}^i \left( \sum_{k_p: f_{k_p} \text{ pickup}} g_{jk_p} - \sum_{k_d: f_{k_d} \text{ delivery}} g_{jk_d} \right) \leq L_{\max} \quad i = 2, \dots, n-1 \quad (9)$$

El vector de tiempo ocupando la matriz G está dado por:

$$T(k+1) = \left[ T^0(k) \quad T^0(k) + G^{(1)}T(G^{(2)})^T \quad \dots \quad T^0(k) + \sum_{k=1}^{i-1} G^k T(G^{k+1})^T \quad \dots \quad T^0(k) + \sum_{k=1}^{w_j-1} G^k T(G^{k+1})^T \right]^T \quad (10)$$

con  $G^{(i)}$  = Fila  $i$  de  $G$ ,  $T$  es una matriz con los tiempos estimados entre todos los destinos. En cada instante de tiempo se debe generar una matriz  $T$  auxiliar que indique el tiempo de viaje entre todos los puntos posibles en el mismo orden que las paradas en la matriz  $G$ . Ocupando la matriz  $G$ , la función objetivo a un paso (ecuación (4)) se puede describir como:

$$J = \sum_{j=1}^{N_p} \Delta C_j(k+1) \Big|_{S_j(k+t-1), h} \quad (11)$$

$$C_j(k+t) \Big|_{S_j(k+t-1), h} = \sum_{i=1}^{w_j} \left[ L^0(k) + 1 + \sum_{i=1}^{w_j-2} \left( \sum_{k_p: f_{k_p} \text{ pickup}} g_{ik_p} - \sum_{k_d: f_{k_d} \text{ delivery}} g_{ik_d} \right) \right] \left[ G^{(i-1)} T(G^i)^T + \alpha \cdot r^i(k) \left( T^0(k) + \sum_{k=1}^{i-1} G^k T(G^{k+1})^T \right) \right]$$

La formulación analítica del problema permite generalizar los criterios de optimización a  $n$  pasos y evaluar a futuro diversas técnicas de optimización entera mixta no lineal. En un vehículo con varios destinos, la inserción de una nueva llamada provoca que una búsqueda de la solución óptima ocupando enumeración explícita sea muy lenta y no factible como método de optimización para el problema que se debe solucionar en tiempo real reducido. Más aún si se considera que la inserción se debe realizar entre una flota grande de vehículos y con una función objetivo a  $n$  pasos.

Es por esta razón que se proponen los algoritmos genéticos como técnica de solución heurística. En este trabajo se realizará la optimización en un sub-espacio de soluciones que satisfacen la restricción de *no-swapping*. Este criterio lo cumplen secuencias que ubican la *recogida* y *entrega* de la última llamada dentro de la misma secuencia anterior. Con este criterio se encontrarán soluciones factibles del problema, en poco tiempo, que incorporan la información de la optimización en el instante anterior, y que son sub-óptimas en el sentido de no considerar todas las posibles secuencias.

Con este criterio de *no-swapping*, si  $f_1 \rightarrow f_2 \rightarrow \dots \rightarrow f_{n-2}$  es la secuencia anterior a la nueva llamada y si  $f_{n-1}$  y  $f_n$  son la *recogida* y la *entrega* de la nueva llamada, se obtiene la expresión (12) para las paradas posibles. En este caso se mantiene la secuencia  $f_1 \rightarrow f_2 \rightarrow \dots \rightarrow f_{n-3} \rightarrow f_{n-2}$  pero se puede intercalar una sola vez la secuencia  $f_{n-1} \rightarrow f_n$  en cualquier posición. Muchas incógnitas de la matriz  $G$  son cero, lo cual reduce considerablemente el problema de optimización.

En sección 3, se describen el método de optimización basado en algoritmos genéticos que servirá para resolver el problema de ruteo formulado analíticamente en esta sección.

$$P_i = \begin{cases} g_{11}f_1 + g_{1,n-1}f_{n-1} = (x_1, y_1) & \text{si } i=1 \\ g_{21}f_1 + g_{22}f_2 + g_{2,n-1}f_{n-1} + g_{2,n}f_n = (x_2, y_2) & \text{si } i=2 \\ g_{i,i-2}f_{i-2} + g_{i,i-1}f_{i-1} + g_{i,i}f_i + g_{i,n-1}f_{n-1} + g_{i,n}f_n = (x_i, y_i) & \text{si } i=3, \dots, (n-2) \\ g_{n-1,n-3}f_{n-3} + g_{n-1,n-2}f_{n-2} + g_{n-1,n-1}f_{n-1} + g_{n-1,n}f_n = (x_{n-1}, y_{n-1}) & \text{si } i=n-1 \\ g_{n,n-2}f_{n-2} + g_{n,n}f_n = (x_n, y_n) & \text{si } i=n \end{cases} \quad (12)$$

### 3. ALGORITMOS GENETICOS

#### 3.1 Principios básicos

El problema de optimización se resolverá utilizando Algoritmos Genéticos (GA). Este método es muy fácil de aplicar en problemas de Optimización Entera-Mixta debido a que la codificación binaria de las variables enteras no es necesaria. Los genes de los individuos (soluciones posibles) están dados directamente por las variables enteras de optimización que en este caso son las distintas paradas que se deben recorrer y el vehículo que cumplirá el requerimiento. El cálculo de gradiente no es necesario lo que permitirá ahorrar un significativo tiempo computacional. El algoritmo genético simple que se ocupará se puede resumir como (Man *et al.*, 1998):

- Inicializar una población aleatoria de individuos, que corresponden a posibles soluciones.
- Evaluar la función objetivo para cada individuo de la población actual.
- Seleccionar padres aleatorios.
- Aplicar operadores genéticos como *crossover* y/o mutación a los padres, para una nueva generación.
- Evaluar la función objetivo de todos los individuos de la nueva generación
- Escoger los mejores individuos según los mejores valores de la función objetivo.
- Reemplazar los individuos más débiles de la generación anterior por los mejores de la nueva generación generada en el punto 6.
- Si el valor de la función objetivo alcanza una cierta tolerancia o el número máximo de generaciones se ha alcanzado, entonces se debe detener el algoritmo. En otro caso, se regresa al punto 2

#### 3.2 Aplicación al problema de ruteo

Los individuos deben ser secuencias factibles, cumpliendo con las restricciones de precedencia y de carga. Se considera que tienen tres componentes: el vehículo  $j$  en el cual se plantea realizar la inserción y la posición dentro de la secuencia anterior donde se realizará la *recogida* y la *entrega* de la nueva llamada  $(x,y)$ . Una posible población aleatoria  $i$  de 7 individuos para el vehículo  $j$  se muestra en figura 2. Los individuos

pueden repetirse debido a que la generación de una nueva población debe ser aleatoria. Nótese que en el ejemplo el Individuo 2 y el Individuo 6 son iguales. En la generación aleatoria de individuos ya sea en una nueva generación o luego de aplicar los operadores genéticos mutación o *crossover*, se deben eliminar a aquellos que no satisfagan la condición de carga con un filtro que elimine a los no factibles. Para evitar que el programa no encuentre solución se puede imponer un individuo en la generación inicial a alguno que sea siempre factible como por ejemplo el  $(j, w_j - 1, w_j)$ . Si  $w_j$  es el número de paradas incluyendo la última llamada, la cantidad de individuos posibles para un vehículo  $j$  incluyendo los que no satisfacen la restricción de carga son  $(w_j - 1) \cdot w_j / 2$ .

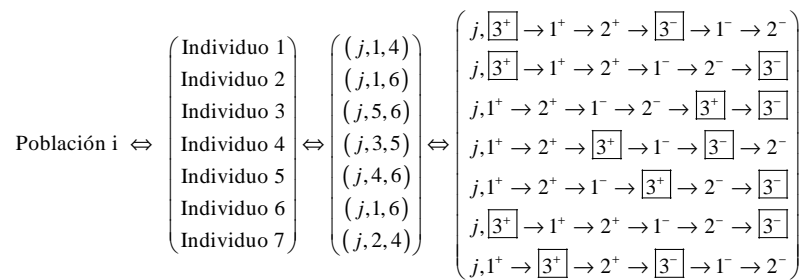


Figura 2.- Ejemplo de población aleatoria  $i$  con siete individuos, para veh.  $j$

Se tomarán entonces poblaciones totales (individuos por generaciones) con un número inferior a las combinaciones totales posibles para evitar resolver el problema como una enumeración explícita. El operador genético *crossover* no se aplicará dado que el problema se define sin *swapping* y por lo tanto el *crossover* generaría soluciones no factibles que no cumplirían con la condición de *no-swapping* dentro del espacio de búsqueda. En la figura 3 se muestra un esquema del sistema de control, donde se aprecia la relación entre el sistema real de Flota-Ciudad-Clientes y el controlador que asigna las rutas. El controlador ocupa las variables de estado, un modelo de predicción de llamadas y la información del nuevo requerimiento para calcular la secuencia óptima o variable manipulada.

## 4. RESULTADOS

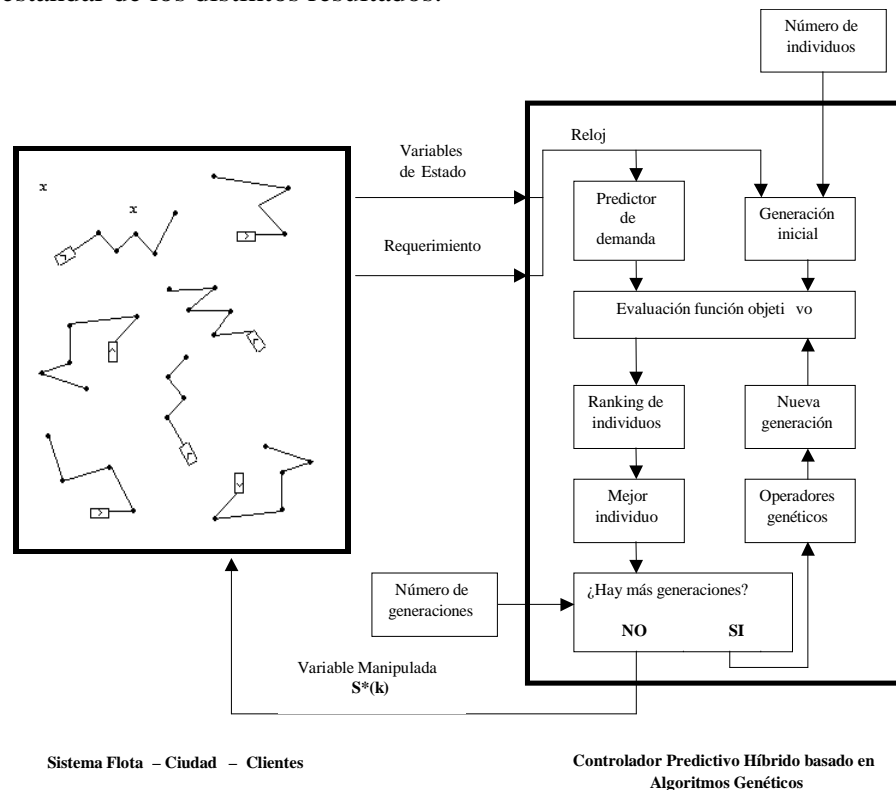
### 4.1 Bases de comparación (*benchmark*)

Se realizan pruebas comparativas entre los métodos de EE y AG para evaluar su rendimiento, graficando la función objetivo y el tiempo computacional que consumen. Se ocupará un sistema con 4 vehículos y predicción a dos pasos con 6 llamadas potenciales a ocurrir. Los vehículos inician su recorrido promediando una velocidad comercial para una red urbana de 20 kilómetros por hora, con un área total de servicio de 400 km<sup>2</sup> (Cortés *et al.*, 2005). Las pruebas que se realizan para comparar los métodos son: i) frente a una acumulación de llamadas, ii) en una operación normal y iii) una

decisión con Genético cuando los individuos aumentan su número de genes o paradas. El método basado en AG tiene como parámetros de ajuste el número de individuos y de generaciones. En este trabajo solo se muestran resultados para tres casos distintos de ajustes fijos y se deja pendiente el determinar el ajuste óptimo de estas variables frente a distintas condiciones de operación. La solución genética G1 considera 5 individuos y 5 generaciones, G2 ocupa 10 individuos y 10 generaciones y G3 ocupa 20 individuos y 20 generaciones.

#### 4.2 Primera prueba: frente a una acumulación de llamadas

En esta prueba se observa el comportamiento de los métodos cuando ocurren muchas llamadas en poco tiempo. Se debe planificar cada nueva llamada dentro de un número cada vez mayor de posiciones posibles y sujetos a la decisión en el instante anterior. En la figura 4 se muestran los gráficos tiempo computacional y función objetivo en función de los instantes  $k$  en los que ocurren nuevas llamadas. En tabla 1 se muestra la media y desviación estándar de los distintos resultados.



**Figura 3.- Esquema del sistema de control para el problema de ruteo.**

Se observa en la figura 4 que en el caso de que ocurra una congestión de requerimientos, los AG presentarán un costo acumulativo en cada nueva llamada debido a que la decisión tomada en el instante anterior no siempre es el óptimo global.

Además, se observa que el tiempo computacional ocupando EE crece exponencialmente con el número de paradas, mientras que con los AG se mantiene prácticamente constante. Según la figura 4 y tabla 1, al aumentar el número de individuos y el número de generaciones se observa un mejor seguimiento de la función objetivo óptima global (G3, en especial).

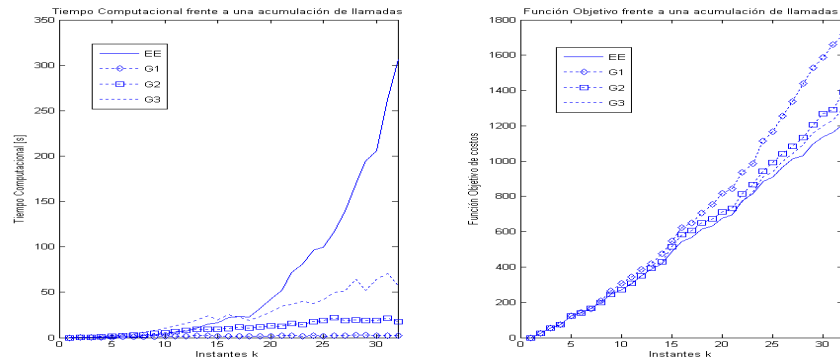


Figura 4.- Gráficos Tiempo computacional y Función Objetivo en función de los instantes k.

Tabla 1.- Media y Desviación Estándar de la función objetivo y tiempo computacional.

Estrategia de Control (Acumulación de llamadas)	Función Objetivo		Tiempo Computacional	
	Media	Desviación Estándar	Media	Desviación Estándar
Enumeración Explícita	561.7069	379.7575	62.6997	83.8158
A. Genéticos G1	708.1025	536.1053	1.6450	0.7910
A. Genéticos G2	602.4819	424.0337	10.5133	7.0684
A. Genéticos G3	576.6193	401.0675	26.5277	21.3254

### 4.3 Segunda prueba: frente a una operación normal

En esta prueba, se consideran secuencias cortas no superiores a los 10 destinos por vehículo. El error de seleccionar una secuencia sub-óptima no será acumulativo pues en la mayoría de los casos se alcanza a recorrer paradas planificadas. En la figura 5 se muestran los gráficos tiempo computacional y función objetivo en función de los instantes  $k$ , mientras que en la tabla 2 se muestra la media y la desviación estándar de los distintos resultados. Frente a una operación normal, en la figura 5 se observa que los AG se aproximan en buena forma al óptimo de EE. No se observa ventajas significativas en cuanto a tiempo computacional. En la tabla 2 se observa que a medida que se aumentan los individuos y generaciones la función objetivo se aproxima más a la solución óptima de EE. En particular, los resultados de G3 coinciden con los de EE, debido a que genético abordó todos los casos posibles debido a su elevada cantidad de población.

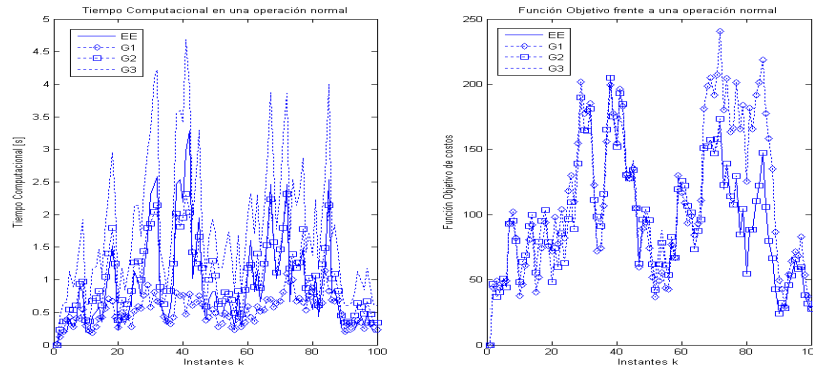


Figura 5.- Gráficos Tiempo computacional y Función Objetivo en función de los instantes k.

Tabla 2.- Media y Desviación Estándar de la función objetivo y tiempo computacional.

Estrategia de Control (Acumulación de llamadas)	Función Objetivo		Tiempo Computacional	
	Media	Desviación Estándar	Media	Desviación Estándar
Enumeración Explícita	94.4609	43.2513	0.9260	0.7232
A. Genéticos G1	110.9015	57.5904	0.5169	0.2200
A. Genéticos G2	95.3830	44.2636	1.0293	0.5431
A. Genéticos G3	94.4609	43.2513	1.8081	0.9981

#### 4.4 Tercera prueba: decisión genético frente a secuencias óptimas con un número elevado de paradas.

En este caso, se analizan los AG cuando las secuencias son largas. Se realizó una prueba similar a la acumulación de llamadas, pero en cada instante se define como secuencia anterior la óptima global dada por la EE. De esta manera se mide el error con AG cuando la secuencia es cada vez más larga. En la figura 6 se muestran los gráficos función objetivo y error de la función objetivo cuando se aumentan los instantes  $k$  y en la tabla 3 la media y la desviación estándar.

De la figura 6 y de la tabla 3 se desprende que la decisión tomada por algoritmos genéticos es en cada instante muy aproximada a enumeración explícita sin importar la cantidad de paradas planificadas.

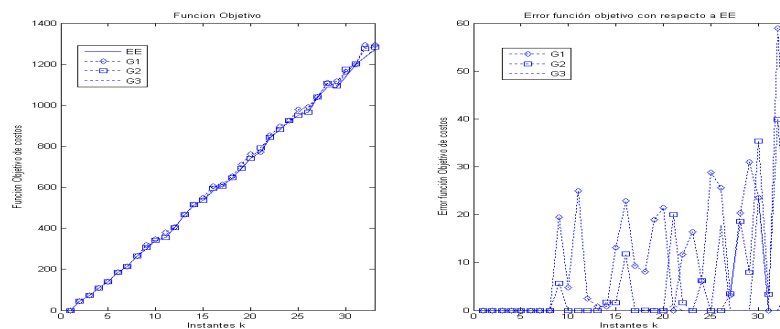


Figura 6.- Gráficos Función Objetivo y error en función de los instantes k.

**Tabla 3.- Media y Desviación Estándar de la función objetivo y error.**

Estrategia de Control (Acumulación de llamadas)	Función Objetivo		Error	
	Media	Desviación Estándar	Media	Desviación Estándar
<b>Enumeración Explícita</b>	604.1539	376.4069	-----	-----
<b>A. Genéticos G1</b>	615.7978	384.6132	11.9823	13.4860
<b>A. Genéticos G2</b>	609.0659	381.9743	5.1809	9.9973
<b>A. Genéticos G3</b>	605.4726	378.0871	1.4405	5.0791

## 5. CONCLUSIONES

En este trabajo se desarrolló una formulación analítica estricta para el problema dinámico de ruteo de vehículos en base a un enfoque de control predictivo híbrido considerando información histórica de demanda en las decisiones, que permite por una parte generalizar la especificación de la función objetivo, y por otra plantear algoritmos de solución más sofisticados que enumeración explícita. Puntualmente, se ha utilizado en forma novedosa algoritmos genéticos para hacer más eficiente la resolución del problema en términos computacionales. Se reporta varias pruebas realizadas para diferentes escenarios posibles, obteniéndose en prácticamente todos los casos resultados promisorios con algoritmos genéticos en términos de la relación entre calidad de soluciones y esfuerzo computacional, al compararse con el óptimo dado por enumeración explícita. En particular, se concluye que para pocas paradas planificadas se sugiere escoger un número de individuos y generaciones reducidos en el algoritmo genético. Además, con este tipo de enfoque es posible resolver problemas más complejos (secuencias más largas, funciones objetivos con más términos, problemas menos restrictivos, etc.) manteniendo el tiempo computacional, aunque en estos casos la elección del número de individuos y generaciones debe ser más cuidadosa.

Como investigación en curso se analizan problemas más generales, búsqueda del óptimo global relajando la restricción de *no-swapping*, horizontes de predicción mayores a dos pasos, incorporación de la variable de estado velocidad de los vehículos introduciendo la estocasticidad en los tiempos de viaje y el peso  $\alpha(label_i)$  se plantea tratarlo como función de cada pasajero con el objeto de implementar una regla de servicio en que se acote la espera de los pasajeros.

## AGRADECIMIENTOS

Esta investigación ha sido parcialmente financiada por los proyectos FONDECYT 1030700 y 1040698, y por el Núcleo Milenio, en su proyecto “Sistemas Complejos de Ingeniería”.



## REFERENCIAS

- Bean N., y A. Costa (2005). An analytic modelling approach for network routing algorithms that use “ant-like” mobile agents. **Computer Networks**, en prensa.
- Bemporad A., W. Heemels y B. De Schutter (2002). On hybrid systems and closed-loop mpc systems, **IEEE Transaction on Automatic Control**, 47, 863–869.
- Bemporad A. y M. Morari (1999). Control of systems integrating logic, dynamics and constraints. **Automatica**, 35, 407–427.
- Ceylan H., y M.G.H. Bell (2004). Traffic signal timing optimization based on genetic algorithm approach, including drivers’ routing, **Transportation Research B**, 38, 329-342.
- Cortés C.E. y R. Jayakrishnan (2004). Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems, **Proceeding of the Triennial Symposium on Transportation Analysis (TRISTAN) V**, Guadeloupe, French West Indies.
- Cortés C.E., D. Sáez, F. Murcia y A. Núñez (2005). Hybrid predictive control for a real-time routed transit system, enviado a **Transportation Science**.
- Floudos C. (1995), **Non-linear and mixed integer optimization**, Oxford University Press.
- Le Bouthillier A., y T.G. Crainic (2005). A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. **Computers & Operations Research**, 32, 1685-1708.
- Li F., B. Golden y E. Wasil (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. **Computers & Operations Research**, 32, 1165-1179.
- Mester D., y O. Bräysy (2005). Active guided evolution strategies for large-scale vehicle routing problems with time windows. **Computers & Operations Research**, 32, 1593-1614.
- Man K.,K. Tang y S. Kwong (1998). **Genetic Algorithms**, Concepts and Designs. Springer.
- Osman M.S., M.A. Abo-Sinna y A.A. Mousa (2005). An effective genetic algorithm approach to multiobjective routing problems (MORPs), **Applied Mathematics and Computation**, 163, 769-781.
- Psaraftis, H. (1998). Dynamic vehicle routing problems, B.L. Golden and A.A. Assad editors, en **Vehicle routing methods and studies**, 223-248.
- Savelsbergh M.W.P., y M Sol (1995). The general pickup and delivery problem. **Transportation Science**, 29(1), 17-29.
- Thomas J., D. Dumur., y J. Buisson (2004). Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition, **Proceedings of American Control Conference**, Boston, Massachusetts, USA.

## ANEXO 3: HYBRID PREDICTIVE CONTROL BASED ON FUZZY MODEL

Alfredo Núñez, *Student Member IEEE*, Doris Sáez, *Senior Member IEEE*,  
Simon Oblak, *Student Member IEEE*, and Igor Skrjanc, *IEEE Member*

*Abstract*— In the paper, the hybrid predictive control based on a fuzzy model is presented. The identification methodology for a nonlinear system with discrete state-space variables by combining fuzzy clustering and principal component analysis is proposed. The fuzzy model is used for hybrid predictive control design where the optimization problem is solved by the use of genetic algorithms. An illustrative experiment on a hybrid tank system is conducted to present the benefits of the proposed approach.

### Introduction

Most industrial processes contain continuous and discrete components, for example discrete valves, discrete pumps, on/off switches or logical overrides. These hybrid systems can be defined as hierarchical systems given by continuous components and/or discrete logic [1]. New tools need to be developed for hybrid-system identification and control design in industrial-process control.

Hybrid systems have received much attention from computer science and from the community of control; nevertheless, there is no general design methodology for identification of hybrid systems yet [2].

In recent years, some hybrid-fuzzy-identification methods have been proposed. Palm and Driankov [3] present a hierarchical identification of the resulting fuzzy switched systems. The proposed method considers a black-box fuzzy identification by using fuzzy clustering and some prior knowledge about the discrete states; however, the last step is not generalized. Next, Girimonte and Babuska [4] describe two structure-selecting methods for nonlinear models with mixed discrete and continuous inputs. The first method based on fuzzy clustering, uses fuzzy sets to obtain the relevant inputs. The second approach is an induction algorithm included in a search method. The results show that the fuzzy clustering is faster in terms of computation time. However, the drawback of the methods is the increase of computational time according to the increase of the search horizon.

On the other hand and regarding to hybrid predictive-control design, Slupphaug [5] and Slupphaug & Foss [6] describe a predictive controller with continuous and integer-input

variables that is tuned using nonlinear mixed-integer programming. It was shown that it performs better than a predictive control strategy with separation of continuous and integer variables. In this case the proposed algorithms were applied to simulate the control of the level and temperature in a tank system.

Bemporad & Morari [7] present a predictive control scheme for hybrid systems including operational constraints. In this case, the problem is solved using mixed-integer quadratic programming (MIQP). The proposed algorithm is applied by simulation to a gas system, which incorporates integer-manipulated variables. The main problem of the MIQP is computational complexity that increases the time to find the solution. To overcome this problem, Thomas et al. [8] propose partitioning in the state-space domain. In every partition some variables change while the others remain constant. This approach reduces the computation time.

Potočník et al. [9] propose a hybrid predictive-control algorithm with discrete input based on reachability analysis. The computation time is reduced by building and pruning an evolution tree.

All the previous works related to hybrid predictive control are based on linear models. In our work we propose the use of a fuzzy model in hybrid predictive control. As the optimization of the objective function in the case of the hybrid fuzzy-predictive control (HFPC) is a highly non-linear problem, the genetic optimization algorithm was employed [10]. Sarimveis [11] used genetic algorithms in fuzzy-predictive control without discrete variables to provide reasonable solutions in a reduced computation time.

The outline of the paper is as follows. In Section II fuzzy modeling of a switching hybrid system with discrete states, based on fuzzy

clustering and principal-component analysis, is presented. In Section III hybrid fuzzy-predictive control design based on genetic algorithms is discussed. Section IV gives simulation results of the control of a hybrid tank system.

### Hybrid Fuzzy Modeling

In the paper we are dealing with the hybrid discrete-time models with mixed continuous and discrete states. We consider the systems where the continuous states remain continuous even when the discrete states are changed. The transition of a system state occurs when one or more continuous states satisfy conditions defined for each transition. This type of a hybrid system is known as Witsenhausen type [12]. It is described in general form as:

$$\begin{aligned} x_{k+1} &= f_{q_k}(x_k, u_k) \\ q_k &= g(x_k, q_{k-1}) \end{aligned}$$

where  $x_k \in R^n$  is the state vector,  $u_k \in R^m$  is the input vector, and  $q_k \in S$  where  $S = \{1, \dots, s\}$  is the switching-region state vector. This means that the hybrid-system states are described at any time instant by the set of states  $(x_k, q_k)$  in the domain  $\square^n \times \square$ . In general the switching state (discrete state)  $q_k$  depends on the state  $x_k$  and the previous switching state  $q_{k-1}$ . The local behavior of the system is described by the function  $f_{q_k}$ , and  $g$  is the transition function of the discrete switching-region states. This type of a system can be represented by the following two-level fuzzy system, which was described by Tanaka et al. [13]. These two levels are the switching-region level and the local-fuzzy level. The

classical form of the two-level fuzzy system is described as

$$\begin{aligned} x_{k+1} &= \sum_{i=1}^s \sum_{j=1}^{R_i} v_i(q_k) \beta_{ij}(z_k) (a_{ij} x_k + b_{ij} u_k + r_{ij}) \\ v_i(q_k) &= \begin{cases} 1, & q_k \in S_i \\ 0, & \text{otherwise} \end{cases} \\ \beta_{ij}(z_k) &= \frac{\prod_{r=1}^p A_{ij,r}(z_{k,r})}{\sum_{j=1}^{R_i} \prod_{r=1}^p A_{ij,r}(z_{k,r})} \end{aligned}$$

where  $s$  is a number of switching regions,  $r_i$  is the number of the rules in the  $i$ th switching region  $S_i$ ,  $v_i(q_k)$  is a crisp switching-region weighting function which is defined by the current switching state  $q_k$  and defines the current switching region, and  $\beta_{ij}(z_k)$  is a local fuzzy-membership value defined by the local premise vector  $z_k = [z_{k,1} \ z_{k,2} \ \dots \ z_{k,p}]^T$ ,  $A_{ij,r}$  ( $r=1, \dots, p$ ) are the local fuzzy sets, and  $A_{ij,k}(z_{k,j})$  is the membership degree of the premise variable  $(z_{k,j})$  to the membership set  $A_{ij,k}$ .

The two-level fuzzy form is very appealing in the case when the switching regions are exactly or very precisely known. However, we are dealing with the case where the switching regions are not known in advance and have to be estimated from the output-input measurements. Therefore, the model will be structured as a one-level fuzzy model with a modified membership-function distribution. The model can be described as

$$\begin{aligned} x_{k+1} &= \sum_{i=1}^{s \cdot R_i} \mu_i(z_k) (a_i x_k + b_i u_k + r_i) \\ \mu_i(z_k) &= \frac{\prod_{r=1}^p B_{i,r}(z_{k,r})}{\sum_{i=1}^{s \cdot R_i} \prod_{r=1}^p B_{i,r}(z_{k,r})} \end{aligned} \tag{1}$$

where  $\mu_i(z_k)$  stands for the membership degree of the product between the crisp switching-region membership value and the local fuzzy membership value. This mixed nature of the system behavior requests more attention in terms of membership-function arrangement. The criterion for the membership-function distribution is based on analyzing the eigenvalues and the eigenvectors of the covariance matrices of the clusters, obtained by clustering algorithms. Let the centers of the clusters be  $v_i$ , the eigenvalues of the clusters  $\{\lambda_{i_1}, \lambda_{i_2}, \dots, \lambda_{i_m}\}$ , and the eigenvectors be  $\{\phi_{i_1}, \phi_{i_2}, \dots, \phi_{i_m}\}$ , where the eigenvalues and the corresponding eigenvectors are arranged in descending order of the eigenvalues. By analyzing the most important eigenvectors (principal vectors or principal components in which directions the most information is given), the switching region can be detected. The criterion is known as the procedure to merge the clusters, which can be usefully applied to detect the switching regions.

## Hybrid Fuzzy Predictive Control based on Genetic Algorithms

### Hybrid fuzzy-predictive control

The HFPC strategy is a generalization of model predictive control (MPC), where the prediction model includes both discrete/integer and continuous variables. Here we propose a hybrid predictive control based on a fuzzy model, described in Section II.

In general, hybrid predictive control minimizes the following objective function

$$\min_{\{u(k), u(k+1), \dots, u(k+N_u-1)\}} J = J_1 + \lambda J_2$$

$$J_1 = \sum_{j=N_1}^{N_y} (\hat{y}(k+j) - r(k+j))^2, J_2 = \sum_{j=N_1}^{N_u} \Delta u(k+j-1)$$

where  $J$  is the objective function,  $\hat{y}(k+j)$  corresponds to the  $j$ -step-ahead prediction for the controlled variable,  $r(k+j)$  is the reference,  $\Delta u(k+j-1)$  is the increment of the control action, and  $\lambda$  is the weighting factor.  $N_1, N_y$  and  $N_u$  are the prediction horizons and the control horizon, respectively. The optimization results in a control sequence  $\{u(k), \dots, u(k+N_u-1)\}$ .

As we assume that the hybrid predictive control problem includes discrete input variables, the optimization could be solved by explicit evaluating for all possible feasible solutions (HPC-EE), Branch & Bound and other algorithms [14]. Next, we explain in detail an efficient optimizer based on GA for this problem.

### Optimization based on Genetic Algorithm

Genetic algorithm is used to solve the optimization of the objective function because it can efficiently cope with mixed-integer non-linear problems.

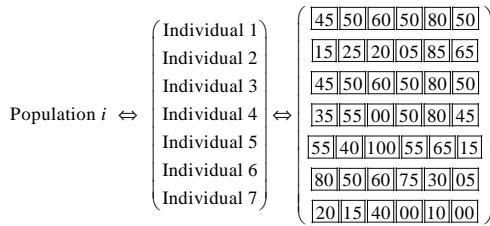
A potential solution of the genetic algorithm is called individual. The individual can be represented by a set of parameters related to the genes of a chromosome and can be described in a binary or integer form. The individual represents a possible control-action sequence  $\{u(k), \dots, u(k+N_u-1)\}$ , where each element is a gene, and the individual length corresponds to the control horizon  $N_u$ .

Using genetic evolution, the fittest chromosome is selected to assure the best offspring. The best parent genes are selected, mixed and recombined for the production of an offspring in the next generation. For the recombination of genetic population, two fundamental operators are used: crossover and mutation. For the crossover mechanism, the portions of two chromosomes are exchanged with a certain probability in order to produce the offspring. The mutation operator alters each portion randomly with a certain probability [10].

For the sake of clarity in terms of control-law derivation based on the simple genetic algorithm (SGA) [10], the following example will be described. We

assume that the range of the manipulated variable is [0,100] quantized by steps of size 5, so there are 21 possible inputs at each instant time. Thus, the procedure for this example is given by the following steps:

1. Initialize a random population of individuals, i.e., create random integer feasible solutions of manipulated variables for the hybrid fuzzy predictive control problem. In this case, as the control horizon is 6, there are  $21^6$  possible individuals. In next example, the size of population is 7 individuals per generation.



For example Individual 1 means that the vector of future control action is:

Individual 1 =

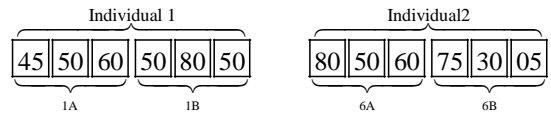
$$\begin{bmatrix} u(k) & u(k+1) & u(k+2) & u(k+3) & u(k+4) & u(k+5) \end{bmatrix}^T = [45 \ 50 \ 60 \ 50 \ 80 \ 50]^T$$

2. Evaluate the fitness function for all initial individuals of the population using Eq. (2). Note that the prediction  $\hat{y}(k+j)$  is calculated recursively by using the future control action. In general

$$\hat{y}(k+j) = f(\hat{y}(k+j-1), \dots, u(k+j-1), \dots)$$

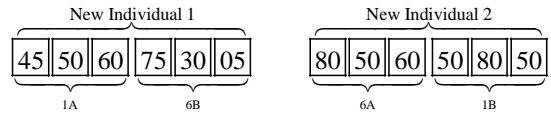
where  $f$  is a non-linear function defined by a fuzzy model.

3. Select random parents from population (different vectors of the future control actions). For example, Individual 1 and Individual 6 are chosen as the parents.

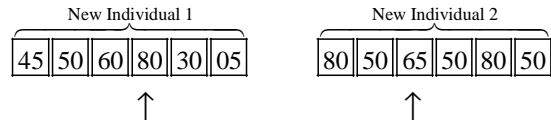


4. Apply crossover and mutation to the parents in order to generate an offspring.

After crossover step



After mutation step



5. Evaluate the fitness given by the objective function (2) of all the individuals of the offspring population.

6. Select the best individuals according to the objective function.

7. Replace the weakest individuals from the previous generation with the strongest individuals of the new generation selected in step 6.

8. If the objective-function value reaches the defined tolerance or the maximum generation number is reached (stopping criteria), then stop. Else, go to step 2.

The genetic algorithm approach in HFPC provides a sub-optimal discrete control law close to the optimal one. The tuning parameters of the GA method are the number of individuals, number of generations, crossover probability, mutation probability and the stopping criteria.

## Tank System

### Process Description

The tank system behavior, shown in Figure 1, is defined by the following nonlinear differential equations and algebraic equations, which define the switching regions:

$$\frac{dh_1}{dt} \cdot \pi \cdot \frac{R_1^2}{H_1^2} h_1^2 = K_{CP} \cdot u + \overbrace{\phi_{ONOFF2}}^{\phi_{v1}} - \overbrace{V_1 h_1}^{\phi_{v1}} - \phi_{ONOFF1}$$

$$\frac{dh_2}{dt} \cdot \pi \cdot R_2^2 = \overbrace{V_1 h_1}^{\phi_{v1}} + \overbrace{\phi_{ONOFF1}}^{\phi_{v2}} - \overbrace{V_2 h_2}^{\phi_{v2}} - \phi_{ONOFF2}$$

if  $(h_2 \geq H_{2min})$  and  $(h_1 < H_{1max})$  then  $\phi_{ONOFF2} = K_{ONOFF2}$   
 if  $(h_1 \geq H_{1max})$  and  $(h_2 < H_{2max})$  then  $\phi_{ONOFF1} = K_{ONOFF1}$

where  $h_1$  and  $h_2$  stand for the level of the liquid in the first and the second tank and  $H_{1min}, H_{1max}, H_{2min}, H_{2max}$  stand for switching levels.

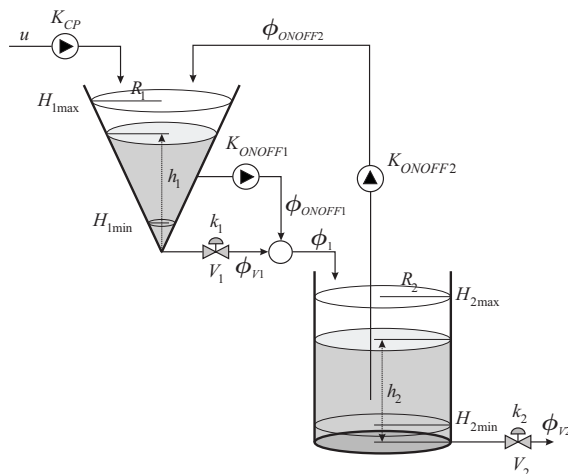


Fig. 1. The tank system plant.

The controlled variable in our case will be the level in the first tank  $h_1$ , and the manipulated variable is the voltage of the pump in the inlet ( $u$ ), which has discrete levels. It is also assumed that both levels  $h_1$  and  $h_2$  are measured. The excitation and the output signals of the plant are shown in Fig. 2 and Fig.3. The signals were sampled with  $T_s = 10$  s. Note that the rules in Eq. (3) represent the switching or hybrid behavior.

### Fuzzy Modeling of Hybrid System

The hybrid system behavior will be modeled by the fuzzy-model structure from (1). The design of the membership-function distribution is the key element of the modeling procedure. In our case it is obtained by analyzing the principal eigenvectors of the covariance matrices of the clusters. The clusters are obtained from the data matrix which is composed of the measurements (the variables  $h_1$  and  $h_2$ ). The experiment should be designed to get the information about all the possible clusters (behaviors of the system). The analysis of the principal components (principal eigenvectors) is in our case realized as the analysis of the principal-eigenvector elements. The ratio between the principal-eigenvector elements is calculated for each cluster.

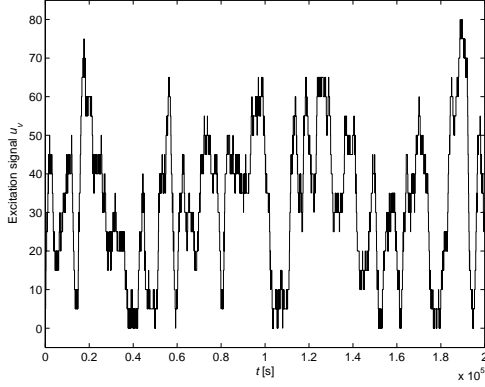


Fig.2. The input signal  $u$

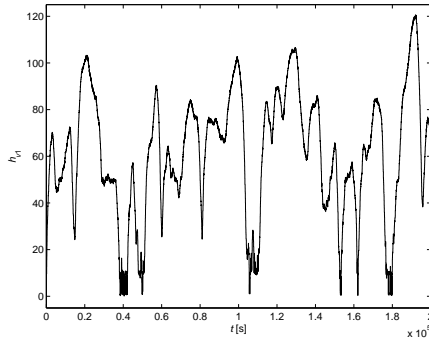


Fig. 3. The output signal  $h_l$

The analysis of the principal eigenvectors for all clusters is presented in Fig. 4 where eigenvector-element ratio corresponds to its own cluster. It is shown that in some clusters the principal eigenvectors locally drastically change. This change implies on the change of the system behaviour and potentially indicates the switching region of the system. The idea is to put two membership functions around each local extremum (minimum and maximum of eigenvector ratios). This is done because the switching region cannot be exactly defined (especially in the case of noisy data). This idea involves a tolerance band around the switching regions. In Fig. 4 the corresponding membership functions are also shown.

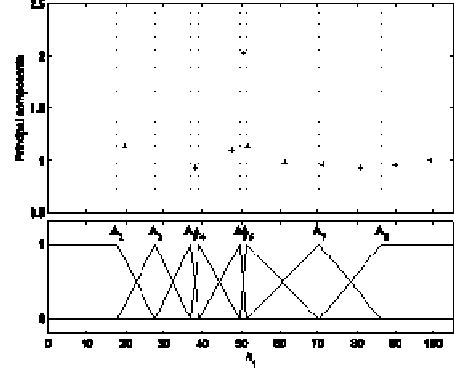


Fig. 4. The analysis of eigenvectors and the corresponding membership functions

The structure of the fuzzy model is defined as in Eq. (1) where the variable in premise is  $z_k = h_{1,k}$  and the consequent vector is equal to  $x_c = [h_{1,k} \ u_k \ 1]^T$ . The parameters of the fuzzy model ( $\theta_i = [a_i \ b_i \ r_i]$ ), obtained by linear least-squares estimation, are given in Table I.

TABLE I  
FUZZY-MODEL PARAMETERS

$i$	$a_i$	$b_i$	$r_i$
1	0.8376	0.3403	0.0386
2	0.9764	0.0522	0.0511
3	0.9873	0.0290	0.0305
4	0.9747	0.0196	0.7656
5	0.9933	0.0125	-0.0136
6	0.9946	0.0091	0.0265
7	0.9987	0.0066	-0.2163
8	1.0015	0.0045	-0.4334

The validation of the designed fuzzy model is shown in Fig. 5. The proposed model gives a very good estimation of the process output, and inherently incorporates the hybrid (switching) nature of the system.

### Hybrid Predictive Control based on Fuzzy Model (HFPC).

The tuning parameters of the objective function in (2) are given by  $N_1 = 1$ ,  $N = N_y = N_u = 3$ , and  $\lambda = 0.001$ . The total computation time required for the HFPC will be evaluated by using a Pentium<sup>®</sup> 4, 2.80 GHz, and 1.00 GB RAM. The total simulation time is 6000 s.

For HFPC-GA a mutation probability equal to 0.001, a crossover probability equal to 0.7 and the maximum number of generations as stopping criterion is used.

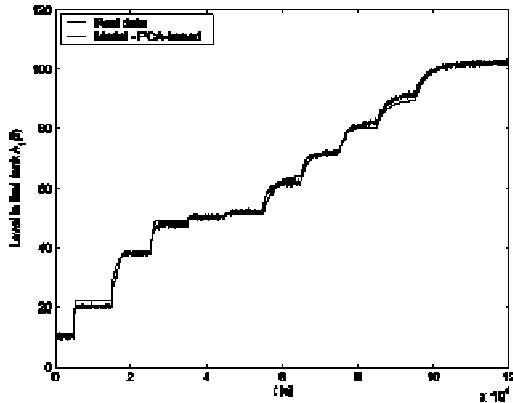


Fig. 5. Validation of the fuzzy model

Fig. 6 shows the objective function as a function of the generation number (and for different number of individuals). Based on this figure, 150 generations with 14 individuals are selected in our example.

Fig. 7 presents the computation time as a function of the number of generations (and for different number of individuals). The computation time is linearly dependent on the generation number, and its slope slightly increases with the number of individuals. Thus, the computation time is reasonably smaller than the simulation time (6000 s). This means that all proposed control strategies are suitable for a real-time control in the

sense of time consumption. For 150 generations with 14 individuals, the computation time was approximately 600 s (a tenth of the total simulation time) as well as the time at each iteration was smaller than the sampling time.

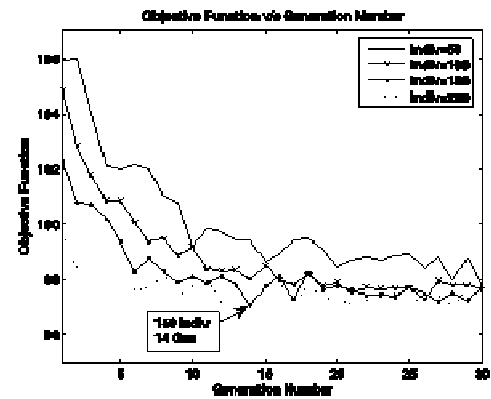


Fig. 6. Evolution of objective function

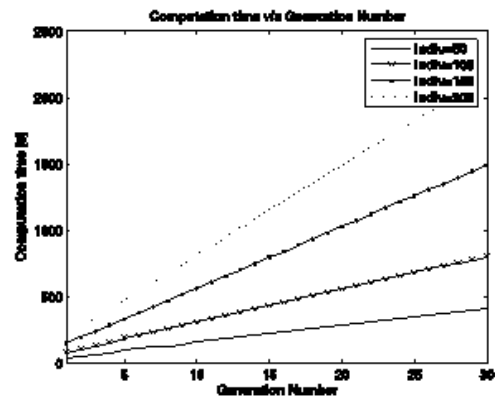


Fig. 7. Evolution of computation time

With optimal values of 150 generation with 14 individuals, the results of HFPC-GA are obtained. Fig. 8 and Fig 9 show the controlled variable (conic tank level  $h_1$ ) and the manipulated variable (discrete voltage of pump  $u$ ), respectively for HFPC-GA and HFPC-EE. The latter evaluates all feasible control actions at every instant while the HFPC-GA considers only a reduced space search as



explained in section II. Fig. 10 and Fig. 11 show the response detail for 3500 to 5000 s.

In Table II the mean values of the objective function (2) and the total computation times for the same simulation test are presented. Table III presents statistical values of the controlled and manipulated variables.

As the HFPC-GA is a heuristic search algorithm, some differences from HFPC-EE for the controlled and manipulated variables can be seen in Figs. 8, 9, 10 and 11. However, the HFPC-GA response is near to the optimal solution given by HFPC-EE as shown in Figs. 8 and 9, as well as in Table II.

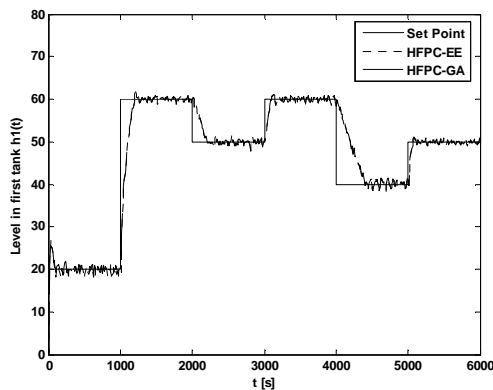


Fig. 8. Simulation test. Controlled variable response

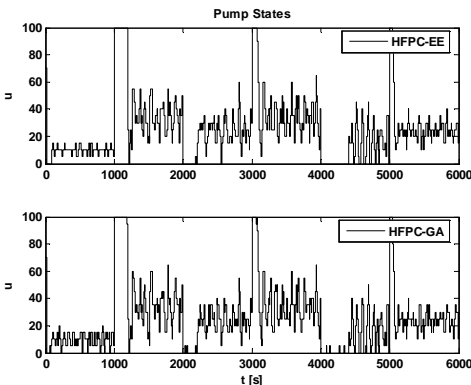


Fig. 9. Simulation test. Manipulated variable

A considerable reduction of computational time is obtained by HFPC-GA in comparison with HFPC-EE. As shown in Table II and Table III, the manipulated variable indices ( $J_2$  and  $\text{std}(\Delta u)$ ) are slightly reduced. However, a little worse tracking response is obtained ( $J_1$  and  $\text{std}(y-r)$ ) due to a narrower space search for HFPC-GA.

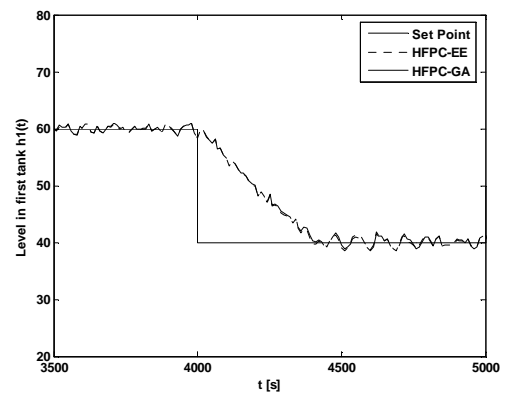


Fig. 10. Simulation test. Controlled variable response. Detail of Fig. 8

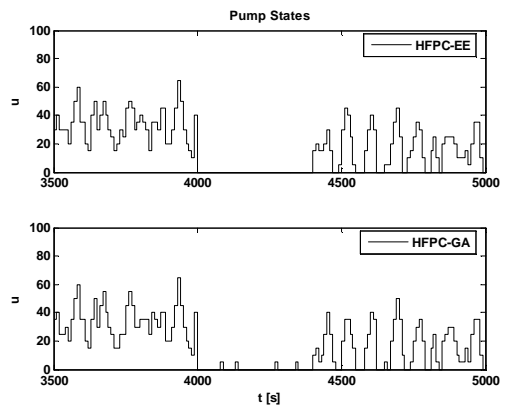


Fig. 11. Simulation test. Manipulated variable response. Detail of Fig. 9.

TABLE II  
MEAN VALUES OF OBJECTIVE FUNCTION AND COMPUTATION TIMES

$N_2=N_u=3,$ $\lambda=0.001$	$J_1$	$J_2$	$J$	Computation Time
HFPC-EE	94.3	487.8	94.8	2850 [s]
HFPC-GA (150,14)	94.8	450.1	95.4	598 [s]

TABLE III  
STANDARD DEVIATIONS OF (Y-R) AND  $\Delta u$

$N_2=N_u=3, \lambda=0.001$	y-r	$\Delta u$
HFPC-EE	5.2095	12.9979
HFPC-GA (150,14)	5.2299	12.4367

## CONCLUSION

This work presents a new approach to control the nonlinear systems with mixed integer and continuous states and inputs. Using a predictive strategy based on a fuzzy model, the problems of the mixed discrete and continuous variables and nonlinearity can be solved together. The key element of the fuzzy model identification in the case of the hybrid systems is the detection and estimation of switching regions, which is realized by the use of fuzzy clustering and principal eigenvector analysis. The optimization of predictive objective function is in the case of hybrid nonlinear systems a NP-Hard problem, which can be efficiently solved by genetic algorithms. The proposed HFPC-GA control algorithm was successfully tested on the hybrid tank system in terms of accuracy and computational time.

In summary, the main contribution of the paper is the combination of using PCA analysis in fuzzy modeling and genetic algorithms in

obtaining the hybrid predictive control law.

Future work will be focused on a general methodology of fuzzy identification for hybrid non-linear systems. Other evolutionary algorithms such as PSO could be investigated. HFPC-GA for a real-time control of tank system is planned to be also applied.

## References

- [1] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics and constraints," *Automatica*, vol. 35, pp. 407-427, 1999.
- [2] A. Bemporad, W. Heemels, and B. De Schutter, "On hybrid systems and closed-loop mpc systems," *IEEE Transaction on Automatic Control*, vol. 47, pp. 863-869, 2002.
- [3] R. Palm and D. Driankov, "Fuzzy switched hybrid systems -Modeling and identification," in *Proc. of the 1998 IEEE ISCI/CIRA/SAS Joint Conf., Gaithersburg MD*, Sept.14-17, pp. 130-135, 1998.
- [4] D. Gririmonte and R. Babuska, "Structure for nonlinear models with mixed discrete and continuous inputs: a comparative study," *Proc. of IEEE International Conf. on Systems, Man and Cybernetics*, pp. 2392-2397, 2004.
- [5] O. Slupphaug, J. Vada, and B. Foss, "MPC in systems with continuous and discrete control inputs," in *Proc. of American Control Conference*, Albuquerque, NM, USA, 1997.
- [6] O. Slupphaug and B. Foss, "Model predictive control for a class of hybrid systems," in *Proc. of European Control Conference*, Brussels, Belgium, 1997.
- [7] A. Bemporad and M. Morari, "Predictive control of constrained hybrid systems," *Automatica*, pp. 71-78, 2000.
- [8] J. Thomas, D. Dumur, and J. Buisson, "Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition," in *Proc. of American Control Conference*, Boston, Massachusetts, USA, 2004.
- [9] B. Potocnik, G. Music and B. Zupancic, "Model predictive control systems with discrete inputs," in *Proc.12th IEEE Mediterranean Electrotechnical Conference*, Dubrovnik, Croatia, 2004, pp:383-386.
- [10] K. Man, K. Tang, and S. Kwong, *Genetic Algorithms, Concepts and Designs*. Springer, 1998.
- [11] G. Sarimveis, "Fuzzy model predictive control of nonlinear processes using genetic algorithms," *Fuzzy Sets and Systems*, N° 139, pp. 59-80, 2003.
- [12] H. S. Witsenhausen, "A class of hybrid-state continuous time dynamic systems," *IEEE Trans. On Automatic Control*, vol. 11, No. 2, pp. 161-167, 1966.
- [13] K. Tanaka, M. Iwasaki and H. O. Wang, "Switching control of an R/C Hovercraft: stabilization and smooth switching," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 853-863, Dec. 2001.
- [14] C. Floudos, *Non-linear and mixed integer optimization*, Oxford University Press, 1995.

## **ANEXO 4: HYBRID ADAPTIVE PREDICTIVE CONTROL FOR THE MULTI-VEHICLE DYNAMIC PICKUP AND DELIVERY PROBLEM BASED ON GENETIC ALGORITHMS AND FUZZY CLUSTERING.**

Doris Sáez<sup>a,1</sup>, Cristián E. Cortés<sup>b</sup>, Alfredo Núñez<sup>a</sup>

<sup>a</sup>Electrical Engineering Department , Universidad de Chile, Av. Tupper 2007, Santiago, Chile.

Phone: 56-2-9784207, Fax: 56-2-6720162, Email: dsaez@ing.uchile.cl

<sup>b</sup>Civil Engineering Department, Universidad de Chile, Blanco Encalada 2002, Santiago, Chile.

Phone: 56-2-9784380 Fax: 56-2-6894206, Email: ccortes@ing.uchile.cl

### **ABSTRACT**

In this paper we develop a family of solution algorithms based upon computational intelligence for solving the dynamic multi-vehicle pickup and delivery problem formulated under a hybrid predictive adaptive control scheme. The scheme considers future demand and prediction of expected waiting and travel times experienced by customers.

In addition, this work includes an analytical formulation of the proposed prediction models that allow us to search over a reduced feasible space. Predictive models consider relevant state space variables as vehicle load and departure time at stops. A generic expression of the system cost function is used to measure the benefits in dispatch decisions of the proposed scheme when solving for more than two-step ahead under unknown demand. The demand prediction is based on a systematic fuzzy clustering methodology, resulting in appropriate call probabilities for uncertain future.

As the dynamic multi-vehicle routing problem considered is NP-hard, we propose the use of Genetic Algorithms (GA) that provide near-optimal solutions for the three, two and one-step ahead problems. Promising results in terms of computation time and accuracy are presented through a simulated numerical example that includes the analysis of the proposed fuzzy clustering, and the comparison of myopic and new predictive approaches solved with GA.

**KEYWORDS:** Dynamic Pick-up and Delivery Problem, Predictive Control, Fuzzy clustering, Genetic algorithms.

---

<sup>1</sup> Corresponding author

## 1. INTRODUCTION AND BACKGROUND

The pick-up and delivery problem (PDP), with or without time windows, has been widely studied in the field-related literature by many authors, from where various formulations as well as solution methods have been proposed to deal with different versions of the PDP [1]. Most exact and heuristic methods have been developed to solve real instances of static and dynamic problems under either stochastic or deterministic demand. In most dynamic versions of the PDP, (with demand that appears in real-time) it is assumed that the dispatcher manages reliable advanced information with regard to service requests. However, it is not very usual to find real-time routing decision rules considering potential future requests entering the system while vehicles are in operation. In this paper, we focus on developing efficient algorithms to solve dynamic multi-vehicle routing problems for passengers by considering future information (prediction) into the current vehicle dispatch decisions. The algorithms are developed based on a hybrid predictive control framework for the dynamic PDP formulated by the authors in a previous work.

Over the last years, the interest in studying the dynamic and stochastic versions of the PDP (associated with dial-a-ride systems) has rapidly grown, mainly due to the access to communication and information technologies, as well as the current interest in real time dispatching and routing. According to [2], stochastic vehicle routing problems are characterized by stochastic demands [3-6], stochastic customers [7] or a mix of both [2-3],[8-9]. In the case of dynamic versions of the vehicle routing problem, our analysis focuses on dynamic pick-up and delivery problem (DPDP), whose final output is a set of routes for all vehicles, which are dynamically changing over time [10-13]. In addition, some applications to real problems of such systems have been implemented [14-15].

Regarding the use of future information to improve dispatching decisions, mainly in the context of the dynamic pick-up and delivery problem, there are just few examples of research on this area. Powell and his team solve the problem of dynamically assigning drivers to loads that arise randomly over time motivated from long-haul truckload trucking applications [16]. Ichoua et al. [17] develop a strategy based on probabilistic knowledge about future request arrivals to manage the fleet of vehicles for real-time vehicle dispatching. Their strategy introduces forecasted customers in vehicle routes to provide a good coverage of the territory. Solution approaches found in this research line are diverse, with formulations based upon dynamic network models [18], dynamic and stochastic programming schemes [19-20], etc.

Recently, Cortés and Jayakrishnan [21] realized that the problem could be modeled under a hybrid adaptive predictive control (HAPC) scheme, considering that potential rerouting of vehicles could affect the current dispatch decisions, through the extra cost of inserting real-time service requests into predefined vehicle routes while vehicles are moving. Cortés et al. [22] describes a formal formulation of the DPDP as an HAPC problem, based on state space variables. The system state is defined in terms of

departure time and vehicle loads at stops (stochastic state space variables), the system inputs (control actions) are routing decisions, the system outputs are effective departure time to stops, and the demand requests are modeled as disturbances. The authors use discrete model with variable step size, with a dispatch objective function incorporating the predictive effect via probabilities computed from historical data regarding typical demand patterns.

Cortés et al. [22] test their model via simulated data, concluding that the inclusion of predictive power improves the performance of system customers, mainly through savings in effective waiting times, when compared with a myopic dispatch decision model. Nevertheless, the authors use an inefficient algorithm (explicit enumeration EE) to solve the problem, which is very restrictive for solving large-scale problems with high demand intensity. In addition, and with regard to the predictive model, the authors use a classic clustering technique (classic zoning) in order to estimate the spatial-temporal probabilities to forecast future pick-up and delivery points.

We recognize that the sophistication of both, solution algorithms and clustering techniques, is crucial to obtain optimal results under more realistic scenarios. Thus, by developing efficient algorithms to solve the problem, a more general cost function formulation can be tested and calibrated via sensitivity analysis. In addition, other model-specific parameters (associated with spatial zoning and probability computations) can also be estimated by using better algorithms.

Therefore, in this paper we develop more efficient solution algorithms (based on GA heuristic techniques) as well as more accurate trip pattern prediction method based on historical data (Fuzzy Zoning). Broadly speaking, the major contributions of this paper are threefold. First, we develop formal analytical formulations of the state space models, which allow the modeler directly use a variety of numerical optimization methods to face different problem conditions. Second, fuzzy zoning is a generic method that computes trip patterns and their probabilities from historical data, so we can have more accurate trip patterns predictions under more realistic scenarios. Finally, and based on such an analytical approach, Genetic Algorithms are proposed and tested based upon a simulated example.

Next, and for the sake of completeness, we review recent literature in the use of heuristic and metaheuristic methods for solving different kinds of vehicle routing problems (VRP), either dynamic or static [23-27].

Over the last few years, several modifications of the well-known Tabu search method have been developed to solve VRP variants, such as granular Tabu search and adaptive memory-based on Tabu search [12], [25], [28]. Another heuristic method for the dynamic VRP is a priority-based solver proposed by [29].

As VRP is NP-Hard, GA based on evolutionary techniques have been analyzed in the

specialized literature. Specifically, GA have been applied to different versions of the VRP, considering various chromosome representations and genetic operators according to the particular problem. Skrllec et al. [30] propose a GA optimization approach with handy heuristic techniques for the single VRP that allows further reducing the computation time by using a certain selection of the initial population. In addition, in [31] the same approach was applied to a multi-vehicle routing problem.

Moreover, [32] describes specialized genetic algorithms based on adaptive parameters to solve the static VRP with time windows that prevents the solution search from a premature convergence and improves the results when compared with the typical GA method. Tong et al. [33] considers a GA method for the static VRP with time windows under uncertain fleet size. To solve this problem, a special gene codification associated with the number of vehicles and routes is considered. Recently, [34] applied a GA optimization method for the multi-vehicle dynamic VRP with time-dependent travel time and soft time windows. This proposition provides promising results in terms of computation times.

Jih and Yung-Jen [35] and Osman et al. [36], present a successful comparison of GA against dynamic programming in terms of computation time. The former method is used to solve the DVRP with time windows and capacity constraints while the latter one is addressed to solve a Multiobjective VRP. Moreover, a hybrid method including both algorithms is described, from which accurate results are obtained in reasonable computation time.

With regard to other heuristics used in the context of the Dynamic VRP, new metaheuristic inspired by the behavior of real ant colonies (ant colony methods) have been applied to solve such problems [37-38]. These methods are especially appropriate to efficiently solve combinatorial optimization problems, and are characterized by the combination of a constructive and a memory-based approach on learning mechanisms [39]. Montemanni et al. [37] also apply Ant Colony Optimization to a realistic case study that obtains promising results. Dréo et al. [38] present good results for a static VRP by optimizing the fleet size as well as the vehicle route plans.

The two general metaheuristics described above (GA and ant colony) have been applied only on myopic dynamic VRP formulations without considering future demand scenarios for improving current dispatch decisions. In this paper, we will show an application of GA on a non-myopic formulation for the dynamic VRP, based upon an HAPC scheme.

The structure of the paper is as follows. In the next section, the model proposed by [22] is summarized, and a formal analytical formulation of the HAPC state space model for DPDP is developed. Next, in Section 3, the fuzzy zoning model to deal with the flexible and systematic use of historical demand patterns is shown and calibrated. In Section 4,

we propose and test the use of GA to solve the analytical model presented in Section 2. Finally, the advantages when using the new fuzzy HPC based GA scheme of solution are quantified by conducting simulation experiments.

## **2. ANALYTICAL FORMULATION: HYBRID ADAPTIVE PREDICTIVE CONTROL (HAPC) APPROACH**

In this section, we formalize the DPDP under a HAPC scheme. The system is formalized in terms of state space variables and the objective function. The fleet size is assumed known, and the cost function does not include time windows on either pick-up or delivery points. The operational cost is approximated by the total vehicle time traveled (see Section 2.2 for details) and the user cost considers both waiting and travel time.

The service consists of picking up (from) and delivering passengers (to) specific spatial coordinates, which are known only after the corresponding real-time request is received by the dispatcher. Vehicles have to be quickly rerouted (from their original sequence of tasks) in order to schedule the new requests into predefined vehicle routes while vehicles are in movement. Routing decisions are taken based on the minimization of an objective function that depends on state space variables associated with the real-time status of vehicles. These state variables should include all the important features of vehicles, which are in our case, the expected departure time and the expected vehicle load at stops. In addition, we assume that historical data are available, regarding pick up and delivery positions (in terms of coordinates) as well as occurrence time of the call. This information feed the predictive model as explained next.

In Section 2.1 the stricter dynamic model, as an extension of the proposal by [22], is formulated for the specific problem of routing a fleet of  $F$  vehicles for serving real-time demand, distributed over a delimited urban area. Travel time conditions and network structure are simplified by considering a constant vehicle average speed when moving from one stop to another. Next in Section 2.2, the general objective function formulation proposed by [22] is summarized, adding the special case of three step ahead prediction. Finally, an operational policy is modeled through an analytical formulation to show the general problem structure and visualize its advantages.

### **2.1 Dynamic model formulation and logical feasibility constraints**

In the context of control theory, hybrid systems are characterized by both continuous and discrete/integer variables. Specifically, hybrid systems can be expressed as a non-linear state space system given by

$$\begin{aligned}x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k))\end{aligned}\tag{1}$$

where  $x(k)$  are the continuous and/or discrete (integer) state space variables,  $u(k)$  are the continuous and/or discrete input or manipulated variables and  $y(k)$  define the continuous and/or discrete system outputs. In general, a hybrid predictive control design minimizes the following generic objective function [40]:

$$\min_{\{u(k), u(k+1), \dots, u(k+N-1)\}} J(u(k), \dots, u(k+N-1), \hat{x}(k+1), \dots, \hat{x}(k+N), \hat{y}(k+1), \dots, \hat{y}(k+N))\tag{2}$$

where  $J$  is a objective function,  $k$  is the current time,  $N$  the prediction horizon,  $\hat{x}(k+t)$ ,  $\hat{y}(k+t)$  are the expected state space vector and the expected system output at instant  $k+t$  respectively, and  $\{u(k), \dots, u(k+N-1)\}$  represents the control sequence, which corresponds to the vector of optimization variables. Once expression (2) is optimized, only the first element of the control vector is used to update the system conditions, based upon the receding horizon methodology.

The DPDP modeling requires a variable stepsize ( $\tau$ ), unlike traditional HPC approaches in which stepsizes are normally fixed. In this case, system events are triggered by specific actions, justifying a variable stepsize as a proxy of expected time interval between calls.

At any instant  $k$ , each vehicle that belongs to the dispatch fleet has associated with a sequence of tasks (stops). Analytically  $S_j(k)$  represents the sequence of stops assigned to vehicle  $j$  at instant  $k$ . As introduced in the previous section, the state-space variables considered here are the estimated departure time and load after vehicles leave each stop belonging to their assigned sequence. In short,  $T_j^i(k)$  and  $L_j^i(k)$  represent the estimated departure time and load when vehicle  $j$  leaves stop  $i$ , computed at instant time  $k$ , respectively. The set of sequences  $S(k) = [S_1(k), \dots, S_j(k), \dots, S_F(k)]$  associated with vehicles corresponds to the manipulated variable  $u(k)$  and the requests asking for service are the model disturbances.

In summary, at instant time  $k$ , each vehicle  $j$  has associated with a sequence of assigned stops  $S_j(k)$ , and two vectors containing the estimated departure time  $T_j^i(k)$  and the estimated load  $L_j^i(k)$  at stops, each of dimension  $w_j(k)+1$ , where  $w_j(k)$  is the number of stops assigned to vehicle  $j$  at time  $k$ . Analytically:



$$S_j(k) = \begin{bmatrix} z_j^0(k) & 1-z_j^0(k) & P_j^0(k) & id_0(k) & \mu(id_0(k)) \\ z_j^1(k) & 1-z_j^1(k) & P_j^1(k) & id_1(k) & \mu(id_1(k)) \\ z_j^2(k) & 1-z_j^2(k) & P_j^2(k) & id_2(k) & \mu(id_2(k)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_j^{w_j(k)}(k) & 1-z_j^{w_j(k)}(k) & P_j^{w_j(k)}(k) & id_{w_j(k)}(k) & \mu(id_{w_j(k)}(k)) \end{bmatrix} \quad (3)$$

$$T_j(k) = \begin{bmatrix} T_j^0(k) \\ T_j^1(k) \\ T_j^2(k) \\ \vdots \\ T_j^{w_j(k)}(k) \end{bmatrix}, L_j(k) = \begin{bmatrix} L_j^0(k) \\ L_j^1(k) \\ L_j^2(k) \\ \vdots \\ L_j^{w_j(k)}(k) \end{bmatrix} \quad (4)$$

where

$$z_j^i(k) = \begin{cases} 1 & \text{if stop } i \text{ defined at } k \text{ is a pick-up} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

Note that  $z_j^i(k)$  indicates if the stop  $i$  is either a pick up or a delivery,  $P_j^i(k)$  quantifies the geographical position of stop  $i$  assigned to vehicle  $j$  in terms of spatial coordinates,  $id_i(k)$  identifies the passenger who is calling and  $\mu(id_i(k))$  quantifies the number of passengers to be served between the origin and destination associated to request  $id_i(k)$ .  $P_j^0(k), L_j^0, T_j^0$  (first row of the sequence matrix) represents the vehicle conditions at time  $k$  (spatial position, load and reference clock time).

Vehicle sequences and state space variables have to satisfy a set of constraints that depend on the real conditions of the modeled DPDP. Thus, we identify precedence constraints, capacity constraints and others, which can be written as logical conditions, as follows:

**Constraint 1.-** Constraint of precedence. The delivery of a passenger cannot happen before its pick up. Therefore, the final node of every sequence has to be a delivery. In short,  $z_j^{w_j(k)}(k) = 0, \quad \forall j: 1 \dots F$ .

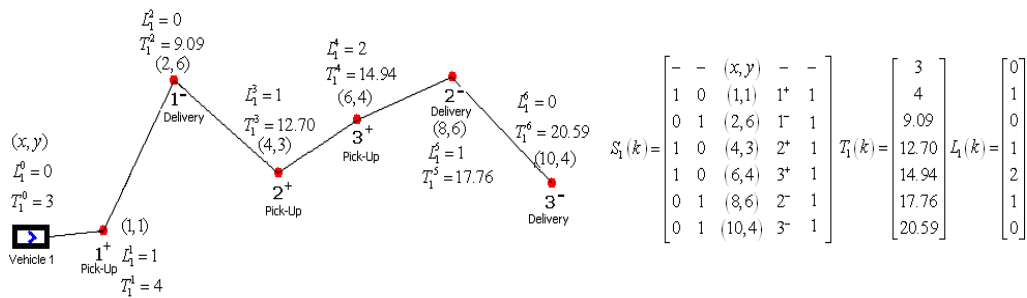
**Constraint 2.-** A destination  $P_j^i(k)$  must be visited only once, and is assigned to only one label (customer). This constraint is analytically treated in equations (8)-(12), ahead in this section.

**Constraint 3.-** Consistency. Once a group of passengers get on a specific vehicle, they have to be delivered to the destination by the same vehicle. This constraint is also analytically treated in equations (8)-(12), ahead in this section.

**Constraint 4.-** Capacity load constraint. A vehicle will not be able to carry more passengers than its maximum load, that is  $L_j^i(k) \leq L_{\max}$ .

In general, travel time in urban areas present a high variability over a normal day, due to many factors such as accidents, traffic and weather conditions [41]. For simplicity, in our applications a conceptual network with Euclidean norm as a distance estimator and constant speed of vehicles were considered. Notice that, in our modeling scheme, the only source of stochasticity with regard to effective travel and waiting times is caused by new call requests coming up in real time, through real-time rerouting of some vehicles to insert such new clients into predefined vehicle routes. Additional stochasticity sources can be added to the modeling scheme, which imply to change the state space model (by adding new variables) and the objective function (by modifying certain components depending on the new state space model). In the next section, some of the potential adjustments of the state space model are discussed in the context of the new analytical formulations as well as the new family of powerful algorithms to deal with real DPDP applications.

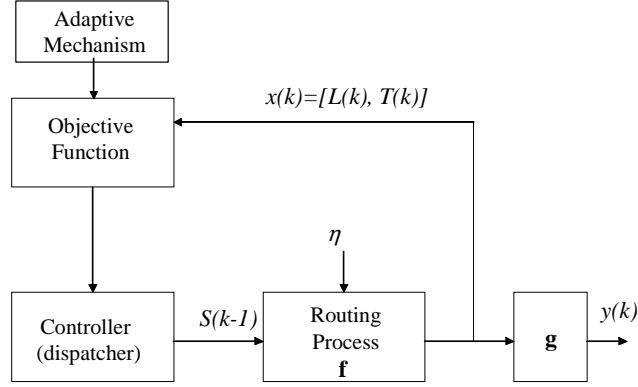
In Fig. 1 an example of a specific vehicle sequence is shown, reporting the sequence, the estimated load (pax/veh) and departure time at stops (min) at time  $k$ , namely  $S_1(k)$ ,  $T_1(k)$  and  $L_1(k)$ . Hereafter, pick up and delivery points are represented with a *plus* and a *minus* superscript on the customer *id*. In short, for client  $r$ ,  $r^+$ ,  $r^-$  denote his(her) associated pick-up and delivery, respectively.



**Fig. 1. Example of vehicle 1 sequence and associated state space variables.**

Next, an analytical formulation is presented. Fig. 2 shows the hybrid predictive controller represented by the dispatching module, which takes routing decisions in real-time based on the information it has from the routing system (process) and the expected values for travel times and attributes of its vehicle fleet (model). An adaptive mechanism for the proposed controller is also added in the figure, representing the necessity of adapting the size of the model when either a new request arrives or a request has been

satisfied. Moreover, this adaptive behavior also affects the controller under uncertain future scenarios depending on demand ( $\eta$ ) through a fuzzy clustering method (see Section 3).



**Fig. 2. Overall block diagram of a Hybrid Predictive approach for DPDP**

The authors propose the following scheme:

$$\hat{x}(k+1) = \begin{bmatrix} E\{L(k+1)/k\} \\ E\{T(k+1)/k\} \end{bmatrix} = \begin{bmatrix} \hat{L}(k+1) \\ \hat{T}(k+1) \end{bmatrix} = \begin{bmatrix} f_L(L(k), S(k)) \\ f_T(T(k), S(k)) \end{bmatrix} \quad (6)$$

$$y(k) = T(k) = [T_1(k), \dots, T_j(k), \dots, T_F(k)]$$

where

$$\begin{aligned} T(k) &= [T_1(k), \dots, T_j(k), \dots, T_F(k)] \\ L(k) &= [L_1(k), \dots, L_j(k), \dots, L_F(k)] \\ S(k) &= [S_1(k), \dots, S_j(k), \dots, S_F(k)] \end{aligned} \quad (7)$$

The functions  $f_L$  and  $f_T$  in equations (6), corresponding to the state space model, are defined in [22]. In this work, we replace the original manipulated variable  $S(k)$  by a matrix of binary activation values  $G = (g_{ir})_{i=1..n, r=1..n}$  that is associated with  $P_j^i(k)$  and (which is a component of  $S(k)$ ). Thus,  $n = w_j(k)$  and the matrix element  $g_{ir} \in \{0, 1\}$  represents the  $r^{\text{th}}$  activation of stop  $i$ .

Finally, a stop  $P_j^i(k)$  associated with passenger  $id_i(k)$  assigned to vehicle  $j$ , can be written as a linear combination of all the known stops  $(f_1, f_2, \dots, f_n)$  using the binary factors of activation  $g_{ir}$ . Analytically

$$P_j^i(k) = g_{i1}f_1 + g_{i2}f_2 + \dots + g_{ii}f_i + \dots + g_{in}f_n \quad (8)$$

where

$$g_{ir} = \begin{cases} 0 & f_r \text{ is not stop } i \\ 1 & f_r \text{ is stop } i \end{cases} \quad (9)$$

Therefore, the stop position vector  $P_j(k)$ , excluding the initial condition  $P_j^0(k)$ , can be written as follows

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1(n-1)} & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2(n-1)} & g_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1} & g_{(n-1)2} & \cdots & \cdots & g_{(n-1)(n-1)} & g_{(n-1)n} \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{n(n-1)} & g_{nm} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G \cdot f \quad (10)$$

From this modeling framework, the constraint 2 above can be written in terms of logical constraints. Thus, the following new constraints in terms of the  $g_{ir}$  values are generated:

$$g_{i1} + g_{i2} + \dots + g_{in} = 1, \quad \forall i = 1, \dots, n \quad (11)$$

$$g_{1r} + g_{2r} + \dots + g_{nr} = 1, \quad \forall r = 1, \dots, n \quad (12)$$

Among the set of stops, we know which one is either a pick-up or a delivery. By respecting the precedence stops as well as all other logical constraints defined above in this section, we can state analytical relations between elements of the  $G$  matrix in order to satisfy such constraints (a pick up has to happen before the associated delivery, etc.). When matrix  $G$  is used as the optimization variable instead of the sequence, the expected load can be expressed as the sum of the initial load plus all the activations of the previous pick-ups less the activations of all previous deliveries, as shown in (13) next:

$$\hat{L}_j(k+1) = \begin{bmatrix} L_j^0(k) & \cdots & L_j^0(k) + \sum_{m=1}^i \left( \sum_{r \in P} \mu(id_r(k)) g_{mr} - \sum_{r \in D} \mu(id_r(k)) g_{mr} \right) & \cdots & \cdots & 0 \end{bmatrix}^T \quad (13)$$

where  $\mu(id_r(k))$  equals de number of passenger at stop  $f_r$  (this value depends on the request) and  $P = \{r : f_r \text{ is a pick-up}\}$ ,  $D = \{r : f_r \text{ is a delivery}\}$

By using (13), the capacity load constraint (constraint 4) can be written based on the activation factors of the matrix  $G$ . Analytically:

$$L_j^0(k) + \sum_{m=1}^i \left( \sum_{r \in P} \mu(id_r(k)) g_{mr} - \sum_{r \in D} \mu(id_r(k)) g_{mr} \right) \leq L_{\max} \quad i = 2, \dots, n-1 \quad (14)$$

In addition, and to complete the state space model, the departure time vector can be expressed as function of the matrix  $G$ . In short,

$$\hat{T}(k+1) = \begin{bmatrix} T^0(k) & T^0(k) + G^1 Q(k) G^{2^f} & \cdots & T^0(k) + \sum_{r=1}^{i-1} G^r Q(k) G^{r+1^f} & \cdots & T^0(k) + \sum_{r=1}^{n-1} G^r Q(k) G^{r+1^f} \end{bmatrix}^T \quad (15)$$

with  $G^r$  denotes the  $r^{th}$  row of  $G$ ,  $Q(k)$  is a matrix containing the network and transfer times computed between stops (from estimations based on Euclidean distance and constant speed).

In this model, most matrices and vectors change in dimension from iteration to iteration. In fact, to capture the dynamic effect caused by the real operation, an expansion and reduction matrix size technique is developed (adaptive behavior). The idea is to either increase or reduce the stop position vector shown in equation (10), resulting in changes on the load and time vectors as well. For example, when certain vehicle accepts a new service request, the dimension of the position vector increases in two rows, accounting for the customer pick-up and delivery stops. Additionally, when a vehicle reaches any stop, that point has to be removed from the original position vector, reducing its dimension in two rows.

## 2.2 Reduction of feasible search space: No swapping case.

In this application, the optimization is performed over a reduced space of solutions that satisfy the *no-swapping* constraint. This criterion provides sequences that locate the pick up and delivery of the last call within the previous sequence (the order of previous stops does not change).

There are practical reasons for considering the no-swapping case in the model instead of exploring over a larger feasible search space. First, any other re-optimization strategy is very time consuming for our algorithm, and not needed in most cases as discussed next. In fact, in all dynamic systems, it is necessary to use the previous information in order to take real-time decisions. Therefore, the configuration of the previous sequences (those scheduled before the insertion) must be considered as a relevant input to the optimization process. Additionally, in most pick-up and delivery problem configurations, the optimal solution of inserting a new request does not alter the order of previous sequences, as shown from simulation experiments by [42]. He found that the no-swapping strategy was optimal in more than 70% of the cases, and in the remainder not-optimal cases, the gap to optimality was negligible.

The global optimum of the dynamic routing problem in terms of the new optimization matrix  $G$  can be obtained by optimally choosing the activation factors  $g_{ir}$ , for each vehicle in the fleet. Indeed,  $G$  determines an optimal sequence of stops  $P_j(k)$  for each vehicle  $j$  that minimizes the objective function defined in the next section, whenever a new real-time request has to be inserted into some previous sequence. Explicitly, the optimal  $P_j(k)$  vector is given by:

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1(n-1)} & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2(n-1)} & g_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1} & g_{(n-1)2} & \cdots & \cdots & g_{(n-1)(n-1)} & g_{(n-1)n} \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{n(n-1)} & g_{nm} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G \cdot f \quad (16)$$

where  $f$  is a vector containing the list of scheduled stops in the whole system at time  $k$ . In the *no-swapping* case, new calls are inserted directly on previous assigned sequences; by keeping the order of previously scheduled stops (only insertions on previous segments are allowed). As previous sequences hold,  $(f_1, f_2, \dots, f_{n-2})$ , the new insertion added to the  $f$  vector at the bottom (pick-up, delivery), and denoted by  $(f_{n-1}, f_n)$ , imposes the following conditions on relation (16) above. Analytically,

$$P_i(k) = \begin{cases} g_{11}f_1 + g_{1,n-1}f_{n-1} = (x_1, y_1) & \text{if } i = 1 \\ g_{21}f_1 + g_{22}f_2 + g_{2,n-1}f_{n-1} + g_{2,n}f_n = (x_2, y_2) & \text{if } i = 2 \\ g_{i,i-2}f_{i-2} + g_{i,i-1}f_{i-1} + g_{i,i}f_i + g_{i,n-1}f_{n-1} + g_{i,n}f_n = (x_i, y_i) & \text{if } i = 3, \dots, (n-2) \\ g_{n-1,n-3}f_{n-3} + g_{n-1,n-2}f_{n-2} + g_{n-1,n-1}f_{n-1} + g_{n-1,n}f_n = (x_{n-1}, y_{n-1}) & \text{if } i = n-1 \\ g_{n,n-2}f_{n-2} + g_{n,n}f_n = (x_n, y_n) & \text{if } i = n \end{cases} \quad (17)$$

where  $(x_i, y_i)$  are the spatial coordinates of the  $i$ -stop. For example, the first term of (17) ( $i = 1$ ) represents the first component of the stop sequence that must be either the new pick up or the first stop of the previous sequence. The second term ( $i = 2$ ) represents the second component of the stop sequence that has more options, either the first stop of the previous sequence, the second stop of the previous sequence, the new pickup stop request or the new delivery stop, and so on.

Equation (17) can also be written in the form of general expression (16), obtaining the following sparse  $G$  matrix (optimization decision matrix):

$$G = \begin{bmatrix} g_{11} & 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{1(n-1)} & 0 \\ g_{21} & g_{22} & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{2(n-1)} & g_{2n} \\ g_{31} & g_{32} & g_{33} & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{3(n-1)} & g_{3n} \\ 0 & g_{42} & g_{43} & g_{44} & 0 & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{4(n-1)} & g_{4n} \\ 0 & 0 & g_{53} & g_{54} & g_{55} & 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{5(n-1)} & g_{5n} \\ 0 & 0 & 0 & g_{64} & g_{65} & g_{66} & \dots & \dots & \dots & \dots & 0 & 0 & 0 & g_{6(n-1)} & g_{6n} \\ \vdots & \vdots & \vdots & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & g_{(n-4)(n-6)} & g_{(n-4)(n-5)} & g_{(n-4)(n-4)} & 0 & 0 & g_{(n-4)(n-1)} & g_{(n-4)n} \\ \vdots & \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \cdot & 0 & g_{(n-3)(n-5)} & g_{(n-3)(n-4)} & g_{(n-3)(n-3)} & 0 & g_{(n-3)(n-1)} & g_{(n-3)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & g_{(n-2)(n-4)} & g_{(n-2)(n-3)} & g_{(n-2)(n-2)} & g_{(n-2)(n-1)} & g_{(n-2)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & 0 & g_{(n-1)(n-3)} & g_{(n-1)(n-2)} & g_{(n-1)(n-1)} & g_{(n-1)n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 & 0 & 0 & g_{i(n-2)} & 0 & g_{in} \end{bmatrix}$$

This analytical problem formulation allows us to generalize the  $N$ -step ahead optimization criteria defined in the next section and to evaluate different nonlinear mixed integer optimization methods, as the GA method described in Section 4. If the *no-swapping* operational constraint is relaxed, the search space for optimization increases,

resulting in a less sparse matrix  $G$ , allowing the optimization procedure to obtain a solution closer to a less restrictive global optimum. An intermediate case (*partial swapping*) is currently being studied as discussed in the further research section.

Once the state space variables are analytically defined in Section 2.1 (equations (13) and (15)), and the search space conditions are stated (Section 2.2), the objective function of such an optimization procedure is needed, in order to complete the description of the model. Moreover, the two state space models defined in Section 2.1 along with the objective function, permit the prediction at one, two and more step ahead, which are necessary for implementing the HAPC control strategy. Next, the objective function is presented and discussed.

### 2.3 Objective Function.

In Section 2.1, the problem-specific state space formulation was analytically developed. Here, the concept of cost function is added in order to have a performance measure for deciding the optimal predicted vehicle routes by the controller. In this case, we consider both total expected waiting and travel time for passengers. The idle travel time (vehicles moving around without passengers) is also included in the formulation, as explained next.

The major issue in the definition of the objective function is to define a reasonable horizon for prediction  $N$ , which depends on the studied problem, and also on the intensity of the unknown events entering the system in real-time. In cases where the decision is taken at instant  $k$ , but considering a predictive horizon greater than one, the decision maker (controller) adds the predictive feature into the formulation, since decisions made in  $k+1$  will depend on possible events (new service requests) occurring in future instants ( $k+2$ ,  $k+3$ , ...etc). Thus, the central dispatcher (controller) computes the decisions for the entire control horizon  $N$ , i.e.,  $\{S(k), \dots, S(k+N-1)\}$ , and applies just the next step sequence  $S(k)$ , based on receding horizon control. The routing decisions will depend on how well the system predicts the impact of rerouting passengers due to unknown insertions.

The objective function for a generic prediction horizon  $N$ , can be written as follows

$$\underset{\{S(k), S(k+1), \dots, S(k+N-1)\}}{\text{Min}} J = \sum_{t=1}^N \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} P_h^{\Delta T(k+t)} \cdot \left( (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2), h} \right) \quad (18)$$



$$C_j(k+t) \Big|_{S_j(k+t-2),h} = \sum_{i=1}^{w_j(k+t)} \left\{ \underbrace{\left[ \hat{L}_j^{i-1}(k+t) + 1 \right] \left( \hat{T}_j^i(k+t) - \hat{T}_j^{i-1}(k+t) \right)}_{\text{J travel time}} + \underbrace{z_j^i(k+t) \alpha \left( \hat{T}_j^i(k+t) - T_j^0(k+t) \right)}_{\text{J waiting time}} \right\} \Big|_{S_j(k+t-2),h} \quad (19)$$

where  $k+t$  is the instant at which the  $t^{\text{th}}$  request enters the system, measured from time interval  $k$ .  $h_{\max}(k+t)$  is the number of probable requests at instant  $k+t$ ,  $p_h^{\Delta T(k+t)}$  is the probability of occurrence of the  $h^{\text{th}}$  request type (associated with a specific pair of zones, as discussed ahead in this paper) during time interval  $\Delta T(k+t)$ , noting that  $\Delta T(k+t)$  specifies the time interval to which time step  $k+t$  belongs.  $C_j(k+t) \Big|_{S_j(k+t-2),h}$  in (19) is the cost function of vehicle  $j$  at instant  $k+t$ , which depends on the previous sequence at  $k+t-1$ ,  $S_j(k+t-2)$  and a new potential request  $h$  with probability  $p_h^{\Delta T(k+t)}$ ,  $w_j(k+t)$  is the number of stops estimated for vehicle  $j$  at instant  $k+t$ ,  $S_j(k+t-2), h$  is the new sequence provided that  $h$  occurs. Notice that a variable time step is considered and determined by interval time between two consecutive requests and this step will be tuned using a sensitivity analysis. For the sake of flexibility and economic consistency, the waiting cost component is weighted by a coefficient  $\alpha$ .

As mentioned before, the cost function  $C_j(k+t)$  as shown in (19), can be split into two pieces: a waiting time and a travel time component. Both of them are written as function of the load and departure time and they are computed as the departure time between consecutive stops times.

In addition, expression (19) depends on the sequence matrix  $S(k)$ , which also can be expressed in terms of the matrix  $G$  and its components. Analytically,

$$C_j(k+t) \Big|_{S_j(k+t-2),h} = \sum_{i=1}^{w_j(k+t)} \left[ 1 + L_j^0(k+t-1) + \sum_{m=1}^i \left( \sum_{r \in P} \mu(id_r(k+t-1)) g_{mr} - \sum_{r \in D} \mu(id_r(k+t-1)) g_{mr} \right) \right] \left[ G^{(i-1)} Q(k+t) G^{i^T} \right] + \alpha \cdot z_j^i(k+t) \left( \sum_{r=1}^{i-1} G^r Q(k+t) G^{r+1^T} \right) \quad (20)$$

The probabilities of occurrence associated with each scenario are parameters in the objective function, and they could be computed based on either real-time data, historical data, or a combination of both. In this particular application, we propose a fuzzy zoning

to compute systematically these probabilities from historical data (off line implementation) as Section 3 describes.

The one-step ahead strategy means that the prediction horizon is  $N = 1$ , and  $h_{\max}(k+1) = 1$  since the new requirement is one and known, and therefore its probability is equal to 1, obtaining the following expression for the objective function, by using (18),

$$J = \sum_{j=1}^F \left( C_j(k+1) - \overbrace{C_j(k)}^{\text{known constant}} \right) \Big|_{S_j(k-1)} \quad (21)$$

The difference  $(C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1}$  means that the cost is evaluated considering the control action at the previous instant, represented by  $S_j(k-1)$ . Conceptually,  $J$  represents the insertion cost when the system accepts a new call, computed in real time and considering the entire vehicle fleet. Note that there are many possible alternatives to insert the new request. Thus, the vehicle sequence finally chosen by the controller is obtained by solving (21).

The two-step ahead prediction cost function is slightly different from the previous one, in the sense that now a computation of a closed expression for  $J_j$  is not straightforward as before, since we do not know with certainty the position of the call that will enter the system two steps ahead. However, in the formulation we postulate that the decision for the imminent assignment must depend on the potential future insertions. A probabilistic approach is used in order to incorporate the most likely position of the next potential call into the cost function expression. A distribution for the time interval between successive calls is also assumed in order to compute time interval probabilities.

The two-step ahead hybrid predictive controller selects the vehicle's sequence that minimizes the general two-step ahead objective function. This objective function represents the potential insertion cost when a possible new call will appear near to the spatial vehicle trajectory into a specific interval time given by a probability  $p_h^{\Delta T(k+2)}$ .

The corresponding expression for the objective function, by using (18), in this case is as follows:

$$J = \sum_{j=1}^F \left[ \sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \overbrace{C_j(k)}^{\text{known constant}} \right] \quad (22)$$

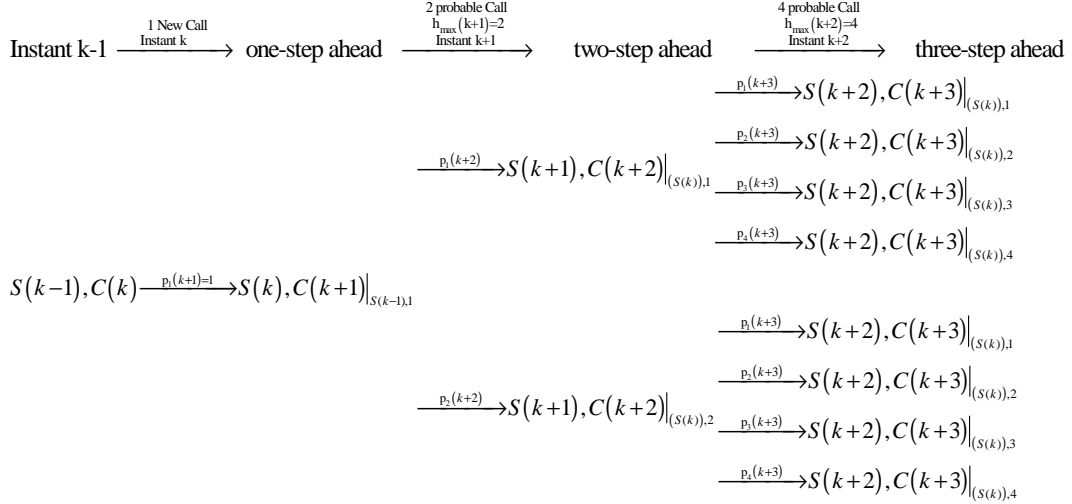
Another interesting case, is the three-step ahead objective function, again computed from the generic expression (17), as follows:

$$J = \sum_{j=1}^F \left[ \sum_{h_2=1}^{h_{\max}(k+2)} p_{h_2}(k+2) \cdot \left( \sum_{h_3=1}^{h_{\max}(k+3)} p_{h_3}(k+3) \cdot C_j(k+3) \Big|_{S_j(k), h_2, h_3} \right) - \overbrace{C(k)}^{\text{known constant}} \right] \quad (23)$$

For illustrating the proposed methodology as shown in Fig. 3, let us concentrate on the three step-ahead prediction case (*no swapping*). In the example, two origin-destination pairs at two-step ahead are likely to occur, and four at three-step ahead, so, the strategy would be evaluated in the chain of scenarios shown in Fig. 3.

For the sake of completeness, consider that at instant  $k-1$ , vehicles follow certain sequence vector  $S(k-1)$  associated with a total cost  $C(k)$ . Whenever a new service request enters the system, there are several feasible sets of sequences  $S(k)$  to be evaluated by the controller (each alternative inserting the new pick-up and delivery in feasible segments of the sequence of a specific vehicle). At one-step ahead, one call is considered (instant  $k$  with probability equals to 1). At two-step ahead, we fix two potential calls appearing in the next time step  $k+1$ , with probabilities  $p_1(k+2)$  and  $p_2(k+2)$  respectively. At three-step ahead, we fix four potential calls appearing in the next time step  $k+2$ , with probabilities  $p_1(k+3)$ ,  $p_2(k+3)$ ,  $p_3(k+3)$  and  $p_4(k+3)$  respectively in order to incorporate the dynamic nature of the problem, and consequently to have good estimations of both travel and waiting times for the cost function decision. Finally, eight potential cases are evaluated for all possible scenarios, containing three new sequential insertions each (the known new call that comes up at one-step ahead and the potential calls that appear at two and three steps ahead)

In order to perform a good estimation of future scenarios in the objective function expressions, we analyze the historical data through a systematic methodology for determining the future trip patterns and their corresponding occurrence probabilities. In the next section, a fuzzy clustering approach is proposed to deal with this issue.



**Fig. 3. Potential combinations of sequences at future.**

### 3. FUZZY ZONING

In this section, a systematic zoning methodology is developed to split the space into conceptual regions for a better representation of historical demand patterns, which can be obtained from demand data associated with a representative operation day. This proposal turns out to be an alternative to the classic zoning approach proposed by [22], where the total area is divided into homogeneous and not overlapping-areas. The classic zoning approach could perform badly in cases where typical origin-destination patterns do not match any of the predefined pair of zones according to the classic method. In fact, a wrong zoning methodology could impact the computation of probabilities in the objective function for more than two-step ahead predictions. The systematic zoning proposed here is based on a fuzzy clustering method that allows us to classify the typical origin-destinations calls in representative and flexible clusters. For simplicity and considering the problem features, we adopt the fuzzy C-means to model such a spatial classification.

#### 3.1 Fuzzy DPDP for probability calculation

The fuzzy C-means (FCM) method proposed by Bezdek [43] is a data clustering technique where each data point belongs to a cluster with a unique membership degree. In other words, the FCM shows how to split the space into a specific number of representative clusters. The FCM considers fuzzy partitioning, such that a data point on the space can belong to more than one cluster, but with different membership degree (which varies from 0 to 1). FCM is an iterative algorithm that allows the modeler to find cluster centers (centroids) that minimize the following objective function

$$S(c) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^m \|x_k - v_i\|^2 \quad (24)$$

where  $n$  is the data number,  $c$  is the number of clusters,  $\mu_{ik}$  is the fuzzy partition between 0 and 1,  $v_i$  represents the center of cluster  $i$  and  $m \in [1, \infty]$  is a weighting factor. The details of the fuzzy C-means algorithm are found in [44]. In this application, the FCM method is used to determine the representative centers associated with historical origin-destination patterns, which will allow us to calculate the corresponding predictive probabilities.

We explicitly propose to compute the probability of each cluster associated with a given origin-destination pair, by following the procedure stated next:

- Step 1: the fuzzy clusters are obtained from historical demand data by using the FCM method.
- Step 2: membership degrees associated with each call from the historical database are computed for every fuzzy cluster obtained in Step 1.
- Step 3: Each call is associated with only one fuzzy cluster, corresponding to that with the biggest membership degree.
- Step 4: Calls with a membership degree smaller than a chosen threshold are not considered in the process.
- Step 5: A probability of occurrence of a new request on a specific origin-destination pair is computed as the number of calls that belong to a fuzzy cluster divided by the total number of calls (after removing the negligible data as explained in Step 4).
- Step 6: Perform a FCM recalculation of cluster center position from historical demand data without considering the negligible data removed in Step 4.

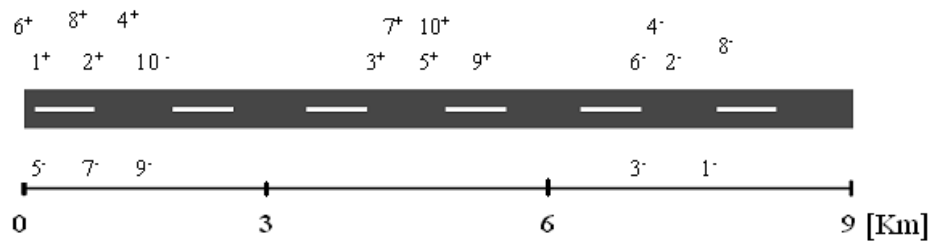
Notice that the optimal number of clusters determines the number of trip patterns for each time period. The number of potential calls (each one occurring with certain probability) for the  $n$ -step ahead will depend on the time period the  $n$  instant belongs, according to the aforementioned clustering method.

In summary, the FCM method permits the modeler to obtain more realistic origin-destination patterns from historical data, and consequently, allows him (her) to systemize and improve the probability calculations. This procedure could improve the prediction power of future uncertainty resulting from the unknown future calls asking for service once they appear, in models with control horizons longer than one-step.

For example, the FCM model performs quite well for jumbled up trip patterns, in which representative zones could be spatially overlapped. Next, a one-dimension example is shown to illustrate the application of the method in the context of the DPDP.

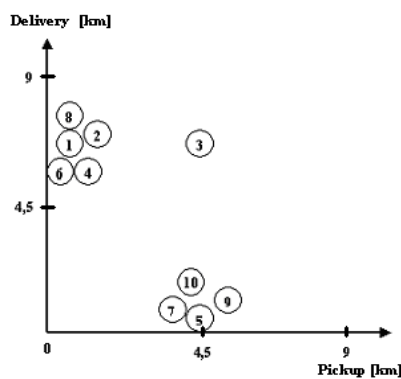
### Illustrative example of the FCM method application

A simple example for a single-vehicle dynamic routing problem is presented in Fig. 4 in order to clarify the application of the FCM for zoning classification as previously described in section 3.2. Let us assume door-to-door requests occurring on a one-dimensional path of nine kilometers, for pick-up and delivery positions. In the example, suppose that ten call requests occur over certain time-period (Fig. 4), and suppose that all stops are considered to determine the optimal zoning and the corresponding probabilities associated with such a partition.



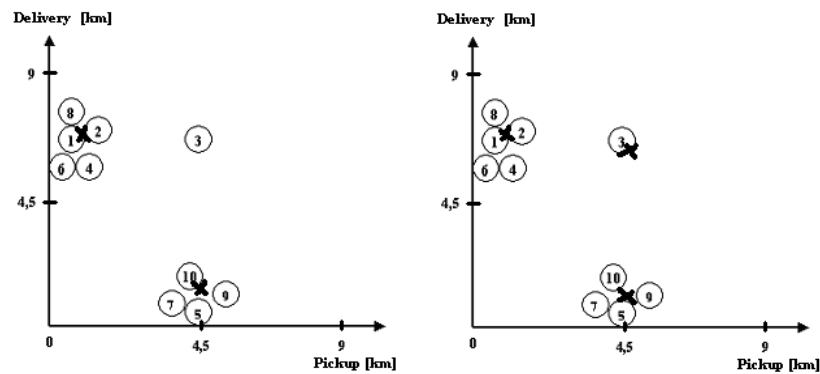
**Fig. 4. Single vehicle requests in a certain period of time.**

Fig. 5 shows a two-dimension representation of pick-up and delivery coordinates, for those requests shown in Fig. 4. By looking at Fig. 5, trip patterns could be identified just by looking at the points and identify those that are close by, since the problem is defined on a one-dimensional path. However, when the problem is defined on a two-dimensional path, the analysis needs an automatic methodology as fuzzy clustering proposed.



**Fig. 5. Pickup-Delivery coordinates of historical demand over a certain time period.**

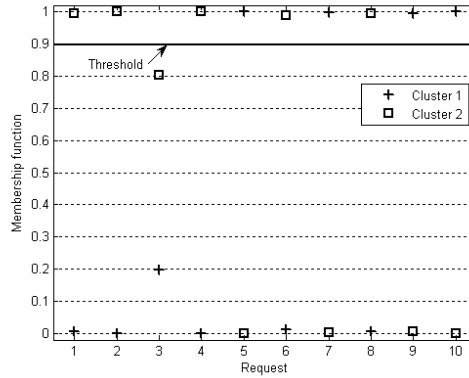
From the historical data shown in Fig. 5, the fuzzy C-means is used in order to obtain the optimal zoning associated with such a database. To do this, a fixed number of fuzzy clusters are selected and thus, Fig. 6 shows the results of FCM for 2 and 3 fuzzy clusters, respectively. As explained in section 3.2, the cluster centers are obtained and denoted by “x” marks in the figure.



**Fig. 6. Cluster centers for 2 and 3 clusters selected.**

Then, the mass centers are obtained after applying the FCM method corresponding to the resulting trip patterns, for this particular example. From an analysis of Fig. 6, it seems reasonable to use 2 clusters instead of 3, since most requests are grouped around two mass centers. In general, stating the number of clusters is not as easy as in this example, and in such cases, the modeler should use methodologies that are more systematic as for example, the fuzzy cluster merging method [44].

Fig. 7 shows the membership degree as function of the ten call requests for 2 fuzzy clusters. As shown in Fig. 7, the threshold selection determines that call 3 does not belong to any of the two fuzzy clusters, and therefore that datum has to be removed from the historical data.



**Fig. 7. Membership degree of historical demand over a certain time period for 2 clusters.**

Finally, and using the FCM procedure, the probabilities associated with trip patterns are shown in Table 1 for 2 fuzzy clusters.

**Table 1. Probabilities for the trip patterns using 2 fuzzy clusters**

Trip pattern	Pickup position	Delivery position	Probability
Fuzzy cluster 1	0.7194	6.9800	4/9
Fuzzy cluster 2	4.4748	0.2750	5/9

The proposed FCM methodology is applied to a more complex simulated example of a DPDP in Section 5.2, and is compared with a classical zoning approach.

#### 4. HAPC based on Genetic Algorithm for a DPDP

Currently, the community of automatic control has shown a great interest in hybrid systems, which are systems that involve both continuous and integer variables, in either state or manipulated variable. The Hybrid Predictive Control (HPC) has shown to be very useful for the control of hybrid systems and authors like [40], [45-46], have reported excellent results by simulation in comparison with other simplified techniques of control. The most used strategies of HPC involve two optimization algorithms: Explicit enumeration (EE) and Branch and Bound (BB). Both allow solving mixed integer optimization problems (MIOP) [47], but the elevated computational effort, especially in the case of EE, results in inefficient solutions for real-time problems. On the contrary, Genetic Algorithms (GA) has proved to be an efficient tool to solve MIOP [48]. Thus, as VRP problems are NP Hard, we consider adaptive HPC (HAPC) based on GA optimization to face the DPDP problem we are dealing with, as explained in the next section.



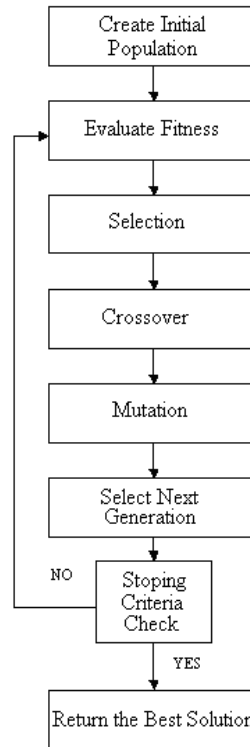
#### 4.1 Genetic Algorithm (GA)

The GA method is suitable for the DPDP since optimization variables are discrete, and therefore the binary codification is not necessary. In other words, genes of the individuals (feasible solutions) are given directly by the integer optimization variables. In addition, gradient computations are not necessary as in conventional non-linear optimization solvers, which allow us to significantly save computation time.

The optimization based on GA [48], presented in Fig. 8, can be described by the following steps:

1. Initialize a random population of individuals corresponding to feasible solutions.
2. Evaluate the objective function for each individual of the current population.
3. Select random parents from current population.
4. Apply genetic operators like *crossover* and/or *mutation* to the parents, for a new generation.
5. Evaluate the objective function for all the individuals of the generation.
6. Choose the best individuals according to the best values of the objective function.
7. Replace the weakest individuals of the previous generation by the best ones of the new generation obtained in step 6.
8. If either the value of the objective function reaches certain tolerance or the maximum number of generations has been reached, then the algorithm stops. Otherwise, go to step 2.

In summary, the proposed genetic algorithm solution provides a solution near the optimum. The GA method tuning parameters are the number of individuals, the number of generations, crossover probability, mutation probability and stopping criteria.



**Fig. 8. GA flowchart**

## 4.2 GA for DPDP

We propose the GA as an efficient optimization solver for the DPDP problem, where the optimization variables identify the stops that must be satisfied by the vehicle fleet. The individuals are the feasible sequences, fulfilling the load, precedence and no swapping constraints defined in Section 2.1. The gene of an individual considers the following three components: the vehicle  $j$  used for the new insertion and the sequence position of the new call (for both pickup and delivery) within the previous sequence, assuming the *no-swapping* policy.

To explain the gene codification, a simple example for one individual is presented. Let us assume the following vector  $P_j(k-1)$ , as defined in section 2.1, associated with the sequence at the previous instant  $k-1$  ( $S_j(k-1)$ ).

$$P_j(k-1) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \end{bmatrix}}_f \quad (25)$$

where  $b(x)$  denotes the position of stop  $x$ . For this example, a new customer labeled as 3 enters the system, and has to be inserted. The new optimization variable can be represented in terms of  $P_j(k)$  as shown in the following matrix equation system, by adding the request in the last two rows of vector  $f$ , increasing the dimension of matrix  $G$ .

$$P_j(k) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \\ P_j^5 \\ P_j^6 \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11} & 0 & 0 & 0 & g_{15} & 0 \\ g_{21} & g_{22} & 0 & 0 & g_{25} & g_{26} \\ g_{31} & g_{23} & g_{33} & 0 & g_{35} & g_{36} \\ 0 & g_{24} & g_{34} & g_{36} & g_{45} & g_{46} \\ 0 & 0 & g_{35} & g_{37} & g_{55} & g_{56} \\ 0 & 0 & 0 & g_{38} & 0 & g_{66} \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \\ b(3^+) \\ b(3^-) \end{bmatrix}}_f \quad (26)$$

Due to the precedence and *no swapping* constraints, the previous sequence is held, and the decision variables are given by the last two columns of matrix  $G$ . By using the proposed gene codification, a feasible population of 7 individuals for vehicle  $j$  is presented by considering the previous sequence (given by expression (25)) and the new call request:

$$\text{Population} \Leftrightarrow \begin{pmatrix} \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \\ \text{Individual 4} \\ \text{Individual 5} \\ \text{Individual 6} \\ \text{Individual 7} \end{pmatrix} \Leftrightarrow \begin{pmatrix} (j,1,4) \\ (j,1,6) \\ (j,5,6) \\ (j,3,5) \\ (j,4,6) \\ (j,1,6) \\ (j,2,4) \end{pmatrix} \Leftrightarrow \begin{pmatrix} j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow \boxed{3^-} \rightarrow 1^- \rightarrow 2^- \\ j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^+} \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow 2^+ \rightarrow \boxed{3^+} \rightarrow 1^- \rightarrow \boxed{3^-} \rightarrow 2^- \\ j, 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow \boxed{3^+} \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, \boxed{3^+} \rightarrow 1^+ \rightarrow 2^+ \rightarrow 1^- \rightarrow 2^- \rightarrow \boxed{3^-} \\ j, 1^+ \rightarrow \boxed{3^+} \rightarrow 2^+ \rightarrow \boxed{3^-} \rightarrow 1^- \rightarrow 2^- \end{pmatrix} \quad (27)$$

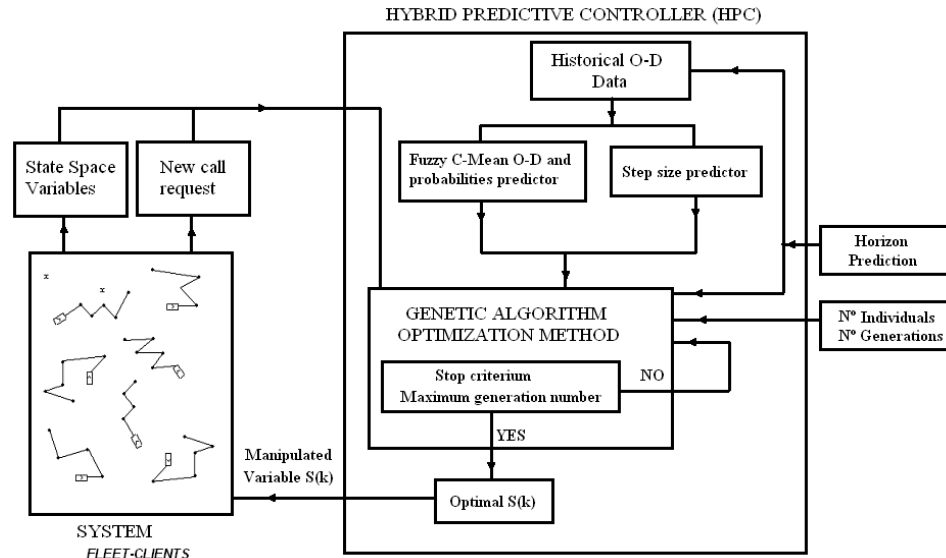
For example, the individual  $(j,1,4)$  in terms of  $P_j(k)$  can be written as:

$$\text{Individual } 1 \Leftrightarrow P_j(k) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \\ P_j^5 \\ P_j^6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \\ b(3^+) \\ b(3^-) \end{bmatrix}}_f \quad (28)$$

In short, the last two columns of matrix  $G$  are the new optimization variables associated with the sequence at instant  $k$ . As the individuals of a generation are randomly selected, the same individuals can be repeated in the next population. For example, individuals 2 and 6 are the same in (27),  $(j,1,6)$ .

Note that as GA considers random generation of individuals, the genetic operators (mutation or crossover) could provide infeasible solutions that have to be removed (typically through the capacity constraint). In order to have at least one feasible solution of the population, an always feasible individual, such as  $(j, w_j-1, w_j)$  must be used ( $w_j$  is the number of stops including the last call). The number of individuals for each population has to be smaller than the total number of feasible combinations in order to avoid solving the explicit enumeration method. The crossover operator is not applied here since the *no swapping* constraint has to be satisfied.

Fig. 9 presents the proposed hybrid adaptive predictive control system scheme. The real system of fleet-clients assigns the sequences using the HAPC controller based on the state space variables, on a call prediction model and on the new call request information.



**Fig. 9. Overall block diagram of an HAPC for DPDP**

Next, an example of the application of GA in the context of DPDP is summarized, to visualize the advantages of that method when compared with explicit enumeration, mainly in computation time saving.

### 4.3 Example

In this section, illustrative tests using explicit enumeration (EE) and GA methods are conducted to evaluate the performance through the proposed objective function (see Section 2.3) and the corresponding computation times.

A DPDP system with 4 vehicles and an objective function of two-step ahead with 6 potential calls are considered. Vehicles cover an urban service area of around 81 km<sup>2</sup>, traveling at an average speed of 20 kilometers per hour [22].

The simulations tests considered are:

- i) Dynamic vehicle routing under high demand conditions,
- ii) Dynamic vehicle routing under normal demand conditions and
- iii) Dynamic vehicle routing considering a mixed solution (combining GA and EE methods).

As mentioned before, the GA method considers the number of individuals and generations, and mutation probability as tuning parameters. Results for three different cases of tuning parameters are presented. The first genetic solution G1 considers 5

individuals and 5 generations, G2 uses 10 individuals and 10 generations, and finally G3 considers 20 individuals and 20 generations.

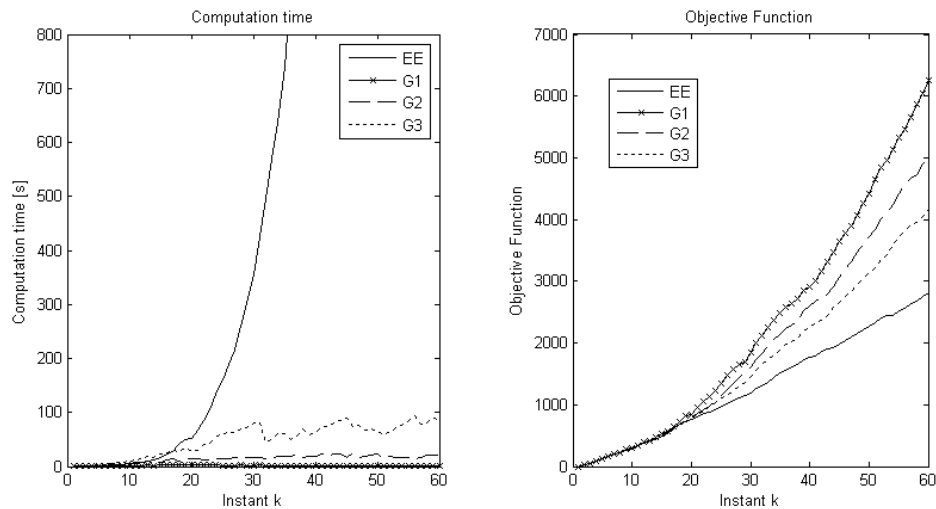
The simulation tests were conducted in Matlab version 6.5.1 release 13, on a Pentium IV processor.

#### 4.3.1 Test 1: Dynamic vehicle routing under high demand conditions.

In this case, many call requests enter the system over a short time period, generating long sequences and consequently, longer computation times due to a larger search space. Fig. 10 shows the computation times and the objective function for a certain period over which a lot of calls enter the system (note that the step size in the model is variable, and depends on when the new call is received by the dispatcher).

From Fig. 10, the request congestion is observed, and therefore GA presents a cumulative cost (see objective function) at each new call because the decision taken at the previous instant (previous sequence) does not always correspond to the global optimum. In addition, the computation time increases exponentially by using EE while the number of stops increases, unlike GA showing stable computation times regardless of the call intensity.

In Table 2, the mean value of the objective function and computation time are reported by using the data presented in Fig. 10. According to Fig. 10 and Table 2, when the number of individuals and the number of generations increase, a better tracking of the global optimum objective function is observed (G3, in special) with a significantly short computation time.



**Fig. 10. Computation time and objective function evolution.**

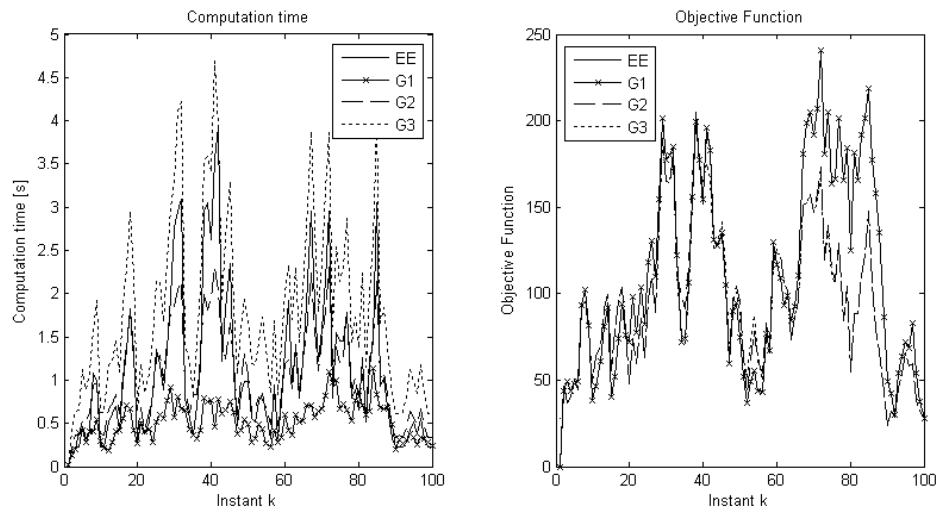
**Table 2. Objective function and computation time mean.**

<b>Control Strategy Test 1</b>	<b>Objective function mean</b>	<b>Computation time mean</b>
<b>Explicit Enumeration EE</b>	1297.4	1536.7
<b>Genetic Algorithms G1</b>	2288.2	1.4
<b>Genetic Algorithms G2</b>	1945.8	13.9
<b>Genetic Algorithms G3</b>	1694.6	49.7

#### 4.3.2 Test 2: Dynamic vehicle routing under normal demand conditions

In this case, few call requests enter the system over the studied time period. The selection of sub-optimal solutions is not very relevant due to the existence of short sequences since most stops are reached while the system is working..

Fig. 11 and Table 3 show computation times and objective function values. By looking at the objective function evolution in Fig. 11, the GA behavior looks similar to the optimal one (EE), while a non-significant computation time effort is observed using GA. Table 3 shows that as the number of individuals and generations increase, the solution converges to the optimal global solution (EE). Notice that the G3 solution is the same as that provided by EE, because G3 computes almost all possible solutions, consuming a longer computation time though.



**Fig. 11. Computation time and objective function evolution.**

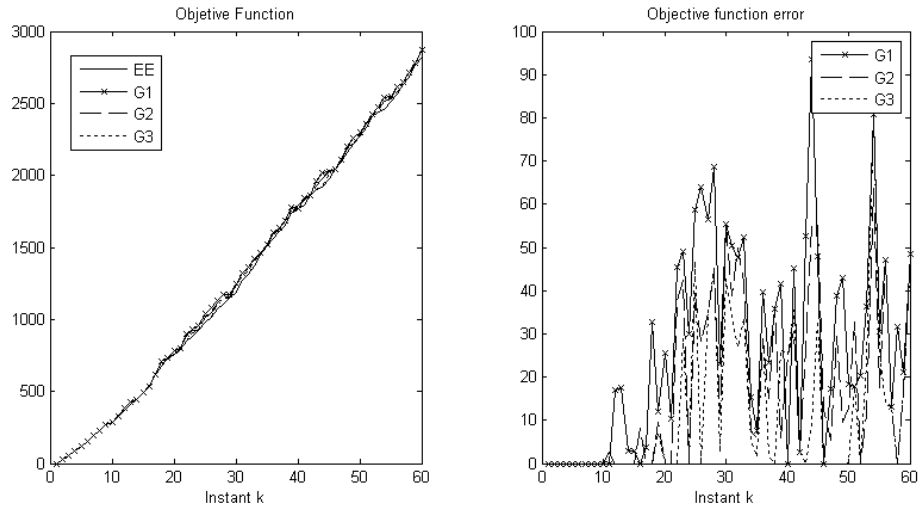
**Table 3. Objective function and computation time mean.**

<b>Control Strategy Test 2</b>	<b>Objective function mean</b>	<b>Computation time mean</b>
<b>Explicit Enumeration EE</b>	94.5	1.1
<b>Genetic Algorithms G1</b>	110.9	0.5
<b>Genetic Algorithms G2</b>	95.4	1.1
<b>Genetic Algorithms G3</b>	94.5	1.8

#### 4.3.3 Test 3: Dynamic vehicle routing considering a mixed solution (combining GA and EE methods)

This case is similar to Test 1, but here the previous sequences for the GA method are calculated by EE, that is to say, at any instant optimization, a good initial solution is used. Fig. 12 and Table 4 show the objective function evolution and its corresponding error with respect to the optimal solution obtained by the EE method. Although the sequence is longer, the GA objective function error is not significantly increased.

According to Fig. 12 and Table 4, dispatch decisions obtained by GA are very similar to EE, regardless of the number of planned stops.



**Fig. 12. Computation time and objective function evolution.**



**Table 4. Objective function and error mean.**

<b>Control Strategy Test 3</b>	<b>Objective function mean</b>	<b>Error mean</b>
<b>Explicit Enumeration</b>	1297.4	-----
<b>Genetic Algorithms G1</b>	1324.0	26.6
<b>Genetic Algorithms G2</b>	1315.1	17.7
<b>Genetic Algorithms G3</b>	1309.3	11.9

In the next section, an illustrative simulated example is presented, including all policies studied in this paper (FCM and GA for one, two and three-step ahead).

## **5. SIMULATION TESTS**

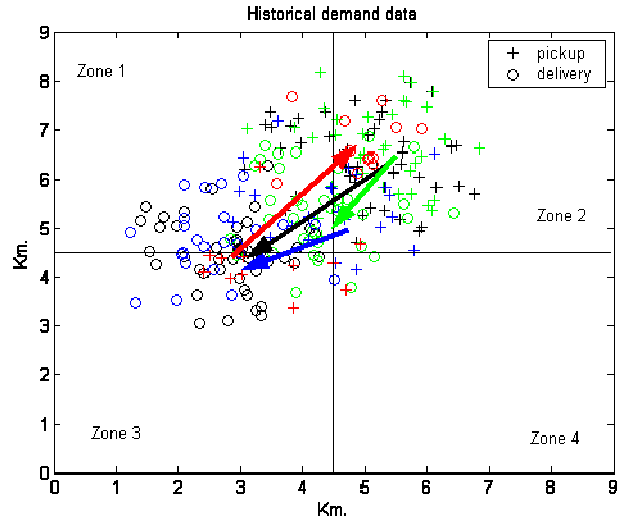
### **5.1 Problem statement.**

A discrete-event system simulation for a two-hour period is conducted in order to evaluate the performance of both fuzzy zoning and genetic algorithm method by using a *no swapping* operational policy. A transportation fleet of nine vehicles, with capacity for four passengers each, is considered. As before, the simulation tests are implemented in Matlab version 6.5.1 release 13 running on a Pentium IV processor.

We assume that the future origin-destination trip patterns are unknown. However, historical demand obtained from the average demand measured over a week before or so, is available. This scenario is not real. However, the demand patterns follow a heterogeneous distribution inspired on real data from the Origin-Destination Survey in Santiago, Chile, 2001.

We consider an urban service area of approximately 81 km<sup>2</sup>. Vehicles are assumed to travel straight between stops at an average speed of 20 km/hr within the region. All simulations are performed over two representative hours (14:00–14:59, 15:00–15:59) of a labour day.

The historical data generated via simulation follows the trips patterns shown in Fig. 13 with arrows.



**Fig. 13. Origin-destination trip patterns.**

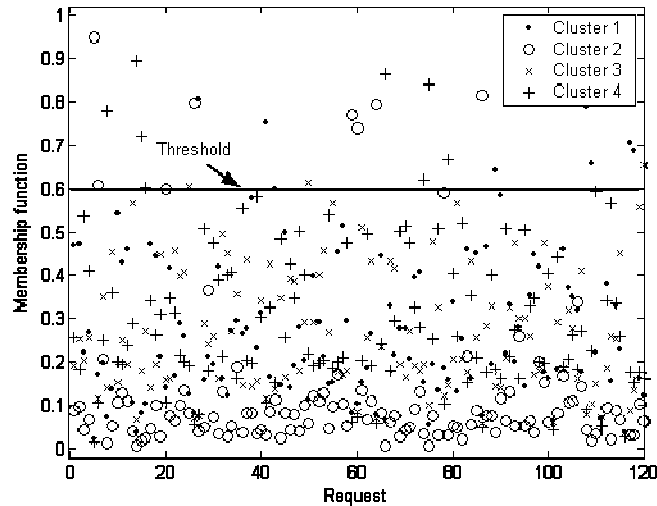
For the simulation test, 120 calls were generated over the whole simulation period of two hours according to a spatial and temporal distribution following the same behavior as that of the historical data. Regarding the temporal dimension, we assume a negative exponential distribution for time intervals between calls with rate of 1 [call/minute] for both the first and second hour of simulation. In terms of spatial distribution, pick-up and delivery points were generated randomly within each corresponding zone. A reasonable warm up period was considered to avoid boundary distortions (10 calls at the beginning and 10 at the end). 50 replications of each experiment were conducted to obtain global statistics. With regard to the cost function, a weight  $\alpha = 1$  was used, which means that travel time is as important as waiting time in the cost function expression.

## 5.2 Fuzzy zoning

In order to compare the performance of the fuzzy zoning proposed with classic zoning (the four squared areas shown in Fig. 13), two-step algorithms were tested and explicit enumeration results were considered for benchmarking.

Fig. 14 shows an application of the procedure described in Section 3.1. In fact, 4 fuzzy clusters are obtained (step 1), next their membership degrees are depicted (step 2). Each call is associated to the biggest membership degree (step 3). In addition, the threshold is fixed and equal to 0.6 in order to consider just the data associated with the relevant trip patterns (step 4). Next the corresponding probabilities are computed (step 5) and the fuzzy cluster centers are obtained again using FCM (step 6).

Table 5 shows the coordinates of fuzzy cluster centers for pickup and delivery points of relevant trip patterns and the corresponding probabilities. On the other hand, Table 6 shows the classic zoning based upon 4 origin-destination pairs.



**Fig. 14. Membership degree for call requests.**

**Table 5. Pickup and delivery coordinates and probabilities: Fuzzy zoning**

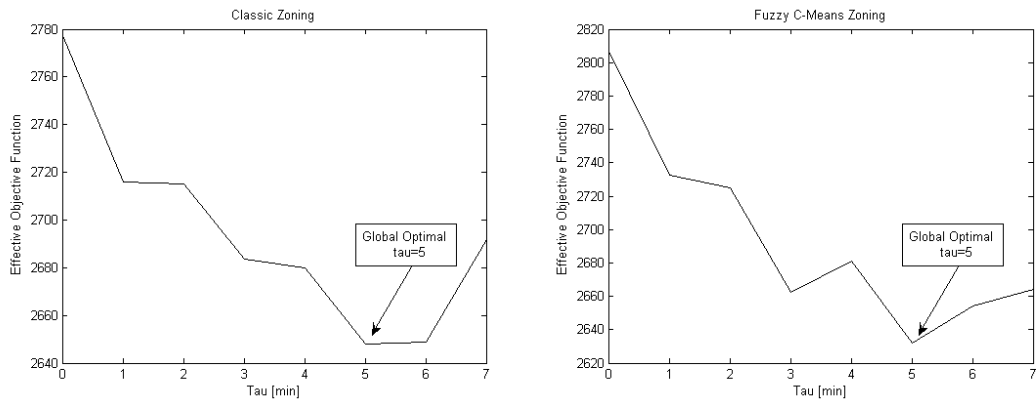
X pickup	Y pickup	X delivery	Y delivery	Probability
4.5540	5.7155	2.9218	4.7514	0.1282
3.7514	4.4812	5.2293	6.2232	0.2051
4.7989	6.6121	3.0751	4.4972	0.2564
5.2595	6.5057	4.3494	5.5161	0.4103

**Table 6. Pickup and delivery coordinates and probabilities: Classic zoning**

X pickup	Y pickup	X delivery	Y delivery	Probability
6.75	6.75	6.75	6.75	0.0968
2.25	6.75	2.25	6.75	0.2151
6.75	6.75	2.25	2.25	0.3118
6.75	6.75	2.25	6.75	0.3763

One relevant fine tuning parameter is the predicted in time between successive calls  $\tau$ , which is relevant when evaluating the performance function of more than one-step ahead algorithms. We found the optimal value of such a parameter by conducting a sensitivity analysis around the observed interarrival times from the historical data report. Fig. 15 shows the effective objective function (considering user as well as operation cost) using

different  $\tau$  values for both classic and fuzzy zonings. Ten replications for each considered  $\tau$  value were used in order to obtain optimal values. For both zoning methods, the resulting optimal  $\tau = 5$ .



**Fig. 15. Sensitivity analysis for  $\tau$  (classic and fuzzy zonings).**

Using the obtained optimal values of  $\tau$ , 50 replications of the two-step ahead algorithm based on explicit enumeration were conducted in order to compare the performance of both zoning methods. Table 7 presents the mean and standard deviations of the waiting, travel and total time for users. The comparison of fuzzy zoning with respect to classic zoning is shown in the same table. We observed that waiting time is significantly reduced (3.36%) while travel time remains almost constant and consequently, total time also decreases (1.71%).

**Table 7. Passenger costs.**

Two step ahead algorithm	waiting time (min)		travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
<b>Classic zoning</b>	6.1437	0.87	10.2358	0.71	16.3795	1.44
<b>Fuzzy zoning</b>	5.9370	0.72	10.1629	0.76	16.0999	1.36
<b>Savings</b>	0.2067		0.0729		0.2796	
<b>Improv. (%)</b>	<b>3.36%</b>		<b>0.71%</b>		<b>1.71%</b>	

Operational costs (mean and std) for the entire vehicle fleet are presented in Table 8. In addition, the total cost including user and operational cost (as in the objective function) is also shown in Table 8. A moderate improvement is observed for both components. However, the proposed fuzzy zoning methodology is a systematic alternative that allows

to determine trip patterns and their corresponding probabilities over a more realistic dynamic dial-a-ride system with jumbled up trip patterns.

**Table 8. Vehicle and passenger operational costs.**

<b>Two step ahead algorithm</b>	<b>Operational costs (min)</b>		<b>Total effective costs (min)</b>	
	<b>Mean</b>	<b>Std</b>	<b>Mean</b>	<b>Std</b>
<b>Classic zoning</b>	117.9	8.81	2699.4	122.84
<b>Fuzzy zoning</b>	115.7	8.12	2651.1	112.86
<b>Savings</b>	2.2618		48.3163	
<b>Improv. (%)</b>	<b>1.92%</b>		<b>1.79%</b>	

### 5.3 Fuzzy HAPC based on GA experiments.

In order to analyze and evaluate the performance of both the proposed fuzzy zoning and the HAPC based on GA, simulation tests were conducted for one, two and three-step ahead algorithms under the same conditions described in section 5.2. We present the results of 50 replications with GA solver by using 20 individuals and 20 generations. We also assume the same trip patterns and probabilities obtained in section 5.2 for the two and three-step ahead scenarios.

Table 9 shows the effective waiting, travel and total times of passengers, by using the fuzzy HAPC based on GA for different prediction horizons.

We observe that waiting time is significantly reduced by using the two-step ahead method (15.04%) and even more from the three-step ahead (22.30%), when compared with the myopic one-step ahead method. In addition, a moderate improvement in travel time is observed. An interesting case is the comparison between the two-step ahead with the three-step ahead predictive method in terms of travel time. In fact, savings in travel time are greater for the two-step ahead method, mainly due to the greater uncertainty as the prediction horizon increases, affecting the reliability of the estimated probabilities. Due to this compensatory fact, the total time saving obtained with the three-step ahead method is almost the same as that of the two-step ahead (9.78% and 9.45% respectively).

Table 10 describes the operational costs (mean and std) for the entire vehicle fleet. In addition, total effective cost is also reported in the table. We observe that vehicle operational costs increase with the two and three-step ahead methods, however, total effective costs are still reduced by running both the two-step ahead (5.9%) and the three step ahead (4.47%) methods. From the results, we can say that the two-step ahead method seems better than the three step ahead algorithm, because the longer the prediction horizon, the less reliable the estimated probabilities are.

**Table 9. Performance comparison for one, two and three step ahead algorithms.**

	waiting time (min)		travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
<b>One Step Ahead</b>	6.969	0.82	10.877	0.89	17.847	1.46
<b>Two Step Ahead</b>	5.921	0.67	10.238	0.79	16.159	1.42
<b>Three Step Ahead</b>	5.415	0.53	10.687	0.65	16.102	1.35
<b>Savings 2 step</b>	1.048		0.639		1.688	
<b>Improv. (%)</b>	<b>15.04%</b>		<b>5.87%</b>		<b>9.45%</b>	
<b>Savings 3 step</b>	1.554		0.190		1.745	
<b>Improv. (%)</b>	<b>22.30%</b>		<b>1.75%</b>		<b>9.78%</b>	

**Table 10. Vehicle and passenger operational costs.**

	Operational costs (min)		Effective total costs (min)	
	Mean	Std	Mean	Std
<b>One Step Ahead</b>	105.04	9.76	2730.0	127.832
<b>Two Step Ahead</b>	105.87	11.68	2568.7	114.516
<b>Three Step Ahead</b>	110.86	11.18	2608.0	112.444
<b>Savings 2 step</b>	-0.84		161.27	
<b>Improv. (%)</b>	<b>-0.79%</b>		<b>5.90%</b>	
<b>Savings 3 step</b>	-5.82		122.05	
<b>Improv. (%)</b>	<b>-5.54%</b>		<b>4.47%</b>	

## 6. CONCLUSIONS AND FURTHER RESEARCH

In this paper an analytical formulation for the DPDP based on a HAPC approach is developed considering historical demand information for a systematic future prediction to improve current dispatch decisions. The major contributions of this paper are three. First, we develop formal analytical formulations of the state space models. Second, fuzzy zoning is utilized to compute probabilities and trip patters from historical data under more realistic scenarios. Third, and based on such an analytical approach, GA are proposed and tested based upon a simulated example.

One major contribution of this formulation is the use of artificial intelligence methods to find better dynamic dispatch decisions under non-myopic scenarios (more than one-step ahead prediction). Particularly, GA is presented as an efficient solver in computation times for this DPDP based upon a detailed analytical formulation. We proved that under certain conditions, a scenario of more than two-step ahead can be solved by using GA in reasonable computation time. The analytical formulation developed in this research can

be potentially utilized to fit other numerical methods to solve the DPDP optimization process.

We conclude that EE works quite well for small problems (for instance, few planned stops and few vehicles). However, as the problem size increases (for example, under more realistic systems), GA becomes an attractive alternative to solve such problems in manageable computation time. We believe that GA applied to this specific problem is a good option to face more complex problems (such as the use of longer sequences, more sophisticated objective functions, relaxed constraint problems, etc.). Note that choosing the number of individuals and generations is a critical point to get reasonable computation time as well as accurate results.

Moreover, we propose a fuzzy zoning based on a fuzzy clustering method to systematically estimate origin-destination patterns from historical data and consequently obtain more reliable computations of the corresponding prediction probabilities. The proposed fuzzy zoning methodology improves the performance of predictive algorithms, mainly under more realistic historical data characterized by jumbled up trip patterns.

The integrated methodology (Fuzzy HAPC based on GA) allows solving for more than two-step ahead prediction to deal with uncertain and heterogeneous demand pattern scenarios.

In further application, we propose to combine historical data (offline) with online information in a more elaborate model able to capture imminent events that could affect the system performance. Second, a more complete rigorous expression for the objective function must be used. That should explore the inclusion of time windows (hard and soft), and a better consideration of operational costs. A sensitivity analysis with regard to both parameters  $\alpha$  and  $\tau$  is planned to be also investigated, for two and three step algorithms. We claim that it is possible to improve the estimation of tuning variables, such as number of probable calls, future step time prediction ( $\tau$ ) which is unknown, prediction horizon ( $N$ ), service policy, search over different feasible solutions structures, etc.

In addition, we plan to relax the *no-swapping* operational policy to test less restrictive dispatching rules, for which the analytical formulation approach would be useful. The incorporation of the network speed within the state space model would allow a better stochastic representation of a real traffic system.

## **Acknowledgments**

This research was partially financed by Fondecyt, Chile, grants 1061156 and 1061261, and the Millennium Nucleus "Complex Engineering Systems".

## REFERENCES

- [1] Savelsbergh M, M Sol. The general pickup and delivery problem. *Transportation Science*, 29(1). 1995. p. 17-29.
- [2] Gendreau M, Laporte G, Séguin R. Stochastic Vehicle Routing. *European Journal of Operational Research*. Vol 88. 1996. p. 3-12.
- [3] Gendreau M, Laporte G, Séguin R. An Exact algorithm for the Vehicle Routing Problem with stochastic Customers and Demands. *Transportation Science* 29. 1995. p. 143-155.
- [4] Yang W, Mathur K, Ballou R. Stochastic Vehicle Routing Problem with Restocking. *Transportation Science* 34(1). 2000. p. 99-112.
- [5] Secomandi N. Exact and Heuristics Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. Ph.D. Dissertation, College of Business Administration, University of Houston (1998).
- [6] Secomandi N. Comparing Neuro-Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. *Computers and Operations Research* 27. 2000. p. 1201-1225.
- [7] Waters C. Vehicle Routing Problems with Uncertainty and Omitted Customers. *Journal of the Operational Research Society* 40. 1989. p. 1099-1108.
- [8] Bertsimas D. A vehicle routing problem with stochastic demand. *Operations Research*, 40. 1992. p. 574-585.
- [9] Bertsimas D, Jaillet P, Odoni A. A priori optimization. *Operations Research*, 38 (6), 1990. p. 1019-1033.
- [10] Psaraftis H. Dynamic vehicle routing problems, B.L. Golden and A.A. Assad editors, Vehicle routing methods and studies. 1998. p. 223-248.
- [11] Psaraftis H. A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem. *Transportation Science* 14(2). 1980. p. 130-154.
- [12] Gendreau M, Guertin F, Potvin J, Taillard E. Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science* 33. 1999. p. 381-390.
- [13] Kleywegt A, Papastavrou J. [The Dynamic and Stochastic Knapsack Problem](#). *Operations Research*, 46. 1998. p. 17-35.
- [14] Madsen O, Raven H, Rygaard J. A heuristics algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60. 1995. p. 193-208.
- [15] Dial R. Autonomous Dial a Ride Transit – Introductory Overview. *Transportation Research - Part C* 3. 1995. p. 261-275.
- [16] Spivey M, Powell W.B. The Dynamic Assignment Problem. *Transportation Science* 38(4). 2004. p. 399-419.



- [17] Ichoua S, Gendreau M, Potvin J.Y. Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching. accepted for publication in *Transportation Science* (2005).
- [18] Powell W.B. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. B.L. Golden and A.A. Assad editors, *Vehicle routing methods and studies*. 1988.
- [19] Godfrey G, Powell W.B. An Adaptive Dynamic Programming Algorithm for Stochastic Resource Allocation Problems I: Single Period Travel Times. *Transportation Science* 36. 2002. p. 21-39.
- [20] Topaloglu H, Powell W.B. A Distributed Decision-Making Structure for Dynamic Resource Allocation Using Non Linear Functional Approximations. *Operations Research* 53(2). 2005. p. 281-297.
- [21] Cortés C.E, Jayakrishnan R. Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems, Proceeding of the Triennial Symposium on Transportation Analysis (TRISTAN) V, Guadeloupe, French West Indies. 2004.
- [22] Cortés C.E, Sáez D, Núñez A, Murcia F. Hybrid predictive control for a real-time routed transit system, under review *Transportation Science* (2006).
- [23] Le Bouthillier A, Crainic T. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research*, 32. 2005. p. 1685-1708.
- [24] Tarantilis C, Ioannou G, Prastacos G. Advanced vehicle routing algorithms for complex operations management problems. *Journal of Food Engineering* 70. 2005. p. 455-471.
- [25] Tarantilis C. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research* 32. 2005. p. 2309-2327.
- [26] Li F, Golden B, Wasil E. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32. 2005. p. 1165-1179.
- [27] Bianchessi N, Righini G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research* (Article in Press). 2006.
- [28] Toht P, Vigo D. The granular Tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, Volume: 15. Fall 2003. Number: 4. p. 0333-0346.
- [29] Tighe A, Smith F, Lyons G. Priority based solver for a Real-Time Dynamic Vehicle Routing. *IEEE International Conference on Systems, Man and Cybernetics*. 2004. p. 6237-6242.
- [30] Skrlec D, Filipec M, Krajcar S. A heuristic modification of genetic algorithm used for solving the single depot capacitated vehicle routing problem. *Proceedings of Intelligent Information Systems, IIS '97*. 1997. p. 184-188.
- [31] Filipec M, Skrlec D, Slavko K. An efficient implementation of genetic algorithms for constrained vehicle routing problem. *Proceedings of IEEE International Conference on System, Man and Cybernetics, SMC'98*. 1998.
- [32] Zhu K. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. *15th IEEE International Conference on Tools with Artificial Intelligence, November 3-5*. 2003. p. 176-183.

- [33] Tong Z, Ning L, Debao S. Genetic Algorithm for Vehicle Routing Problem with Time Window with Uncertain Vehicle Number. *Proceeding of the 5<sup>th</sup> World Congress on Intelligent Control and Automation, June 15-19, Hangzhou, P.R. China.* 2004. p. 2846-2849.
- [34] Haghani A, Jung S. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research* 32. 2005. p. 2959-2986.
- [35] Jih W, Yun-jen J. Dynamic Vehicle routing using hybrid genetic algorithms. *Proceeding of the IEEE International Conference on Robotics & Automation, Detroit, Michigan.* 1999. p. 453-458.
- [36] Osman M, Abo-Sinna M, Mousa A. An effective genetic algorithm approach to multiobjective routing problems (MORPs), *Applied Mathematics and Computation*, 163. 2005. p. 769-781.
- [37] Montemanni R, Gambardella L, Rizzoli A, Donati A. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, Issue: Volume 10, Number 4. 2005. p. 327 – 343.
- [38] Dréo J, Pétrowski A, Siarry P, Taillard E. *Metaheuristics for Hard Optimization Methods and Case Studies*, Springer-Verlag. 2006.
- [39] Dorigo M, Stützle T. *Ant Colony Optimization*. The MIT Press. 2004.
- [40] Bemporad A, Morari M. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35. 1999. p. 407–427.
- [41] Ceylan H, Bell M. Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing. *Transportation Research B*, 38. 2004. p. 329-342.
- [42] Cortés C.E. High Coverage Point-to-Point Transit (HCPPT): A new design concept and simulation-evaluation of operational schemes, Ph.D. dissertation, Civil Engineering Department, University of California at Irvine. 2003.
- [43] Bezdek J. *Fuzzy Mathematics in Pattern Classification*. PhD Thesis, Applied Math. Center, Cornell University, Ithaca. 1973.
- [44] Babuska R. *Fuzzy Modeling for Control*. Kluwer Academic Publishers. 1999.
- [45] Bemporad A, Heemels W, De Schutter B. On hybrid systems and closed-loop mpc systems, *IEEE Transaction on Automatic Control*. Vol. 47. 2002. p. 863–869.
- [46] Thomas J, Dumur D, Buisson J. Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition, *Proceedings of American Control Conference*, Boston, Massachusetts, USA. 2004.
- [47] Floudas C. *Non-linear and mixed integer optimization*, Oxford University Press. 1995.
- [48] Man K, Tang K, Kwong S. *Genetic Algorithms, Concepts and Designs*. Springer. 1998.

## **ANEXO 5: HYBRID PREDICTIVE CONTROL FOR A REAL-TIME ROUTED TRANSIT SYSTEM**

**Cristián E. Cortés<sup>1</sup>, Doris Sáez<sup>2</sup>, Alfredo Núñez<sup>2</sup>**

**<sup>1</sup>Civil Engineering Department  
Universidad de Chile  
Av. Blanco Encalada 2002, Santiago, Chile  
Phone: 562-9784380, Fax: 562-6894206  
Email: [ccortes@ing.uchile.cl](mailto:ccortes@ing.uchile.cl)**

**<sup>2</sup>Electrical Engineering Department  
Universidad de Chile  
Av. Tupper 2007, Santiago, Chile  
Phone:562-9784207, Fax:562-6720162  
Email: [dsaez@ing.uchile.cl](mailto:dsaez@ing.uchile.cl)**

### **ABSTRACT**

This paper presents a hybrid predictive adaptive control approach for estimating the real cost of an insertion based upon future information, in the context of a dynamic pick-up and delivery problem, referred henceforth in this paper as a real-time routed transit system. We recognize in this research is that when the problem is dynamic, an additional stochastic effect has to be considered within the analytical expression of the dispatcher assignment decision cost function in the context of this problem, which is the extra cost associated with potential rerouting arising from unknown requests in the future. The major contribution of this paper is the development of a formal predictive adaptive control framework to model and solve such a problem. Predictive state space formulations are written on the relevant variables (vehicle load and departure time at stops) for the dynamic pick-up and delivery problem model. In addition, an expression of the system cost function is used to measure the benefits of the scheme when predicting to one and two steps ahead in time. Results validating this approach are presented through a simulated numerical example.

## 1. Introduction

One of the most studied problems in the logistics literature is the well known pick-up and delivery problem (with or without time windows), which involves the satisfaction of a set of transportation requests by a vehicle fleet located initially at several depots (Desrosiers *et al.*, 1986; Savelsberg and Sol, 1995). A transportation request consists of picking up a certain number of customers at a predetermined pick-up location during a departure time interval and taking them to a predetermined delivery location within an arrival time interval. Loading and unloading times are incurred at each vehicle stop. The problem can be generalized to the dynamic case, in which a subset of the requests is not known in advance and dispatch decisions have to be taken in real time. The dynamic pick-up and delivery problem has become of great interest in the last decade mainly due to the fast growth in communication and information technologies, as well as the current interest in real time dispatching and routing. The problem can be characterized as a real-time routed transit and has been treated by many authors under different policy schemes in the past (as representative references see Psaraftis, 1988; Madsen *et al.*, 1995; Bertsimas and Van Ryzin, 1991, 1993a, 1993b; Malandraki and Daskin, 1992; Dial, 1995; Gendreau *et al.*, 1999).

In this scenario, if the objective were to transport passengers, inefficient routing decisions could greatly affect the performance of the system as perceived by the users, resulting in poor level of service, low demand, and insufficient productivity. One of the major issues for improving efficiency is the correct definition of a decision cost function for dispatching, including total travel and waiting times for users as well as a performance measure for vehicles. However, when the problem is dynamic, an additional stochastic effect has to be considered when computing an analytical expression for any kind of decision cost function (whether it affects the user or the operator). In words, we recognize that current dispatch actions taken in real time can be affected by potential rerouting decisions decided in the future, affecting most customers already in the system (those waiting as well as those traveling on vehicles).

This issue has been mostly underestimated in the dynamic vehicle routing literature. One assumption behind most of the proposed scheduling-routing rules is that travel and waiting times experienced by customers are considered fixed in the cost expressions, independent of future reroutings. In other words, as stated by Spivey and Powell (2004), the complexity of real time routed transit schemes have generally restricted research to myopic models (for example see Wilson and Weissberg, 1976; Wilson and Colvin, 1977; Psaraftis, 1980, 1988; Madsen *et al.*, 1995; Gendreau *et al.*, 1999; Swihart and Papastavrou, 1999). However, there exists some recent literature in the field of vehicle routing and dispatching trying to exploit information about future events to improve decision-making (Ichoua *et al.*, 2005; Spivey and Powell, 2004). Solution approaches found in this research line are diverse, with formulations based upon dynamic network models (Powell, 1988), dynamic and stochastic programming schemes (Godfrey and Powell, 2002; Topaloglu and Powell, 2005), etc.

From a similar motivation, Cortés (2003) and Cortés and Jayakrishnan (2004) propose a scheme for estimating the effective cost of a real-time request insertion based upon future information, in order to take better dynamic insertion decisions. The authors realized that conceptually the problem perfectly fits within a stochastic predictive control framework, although they did not enter into the control formulation details. Based upon such a motivation, in this paper we develop a consistent framework based upon predictive control theory for optimizing the performance of a dynamic pick-up and delivery problem (real-time routed transit), mainly oriented to passenger movements.

Unlike others non-myopic dynamic vehicle routing approaches, our formulation is based on state space variables. The system state is defined in terms of departure time and vehicle loads (stochastic state space variables), the system inputs (control actions) are routing decisions, the system outputs are effective departure time to stops, and the demand requests are modeled as disturbances. We use a discrete model with variable stepsize equal to the time between successive calls. In order to include future and unknown demand into the current decision, we solve an objective function incorporating the predictive effect via probabilities computed from historical data regarding typical demand patterns.

In summary, we recognize that the major contribution of this research is the development of a hybrid predictive control framework to model the real time routed transit problem. This line of research represents an innovative attempt of developing control based algorithms for modeling and solving dynamic transportation problems, and from here much more detailed techniques have to be devised in the future to study more realistic transportation optimization schemes.

The structure of the paper is as follows: in the next section, the relevant background on dynamic and stochastic vehicle routing is presented. In Section 3, the dynamic pick-up and delivery problem is described in context, and is formulated under an adaptive-predictive control scheme. Thus, the specific state space formulation for the problem is developed, the dispatch cost function associated is shown and the potential solution algorithms are developed to solve the proposed hybrid predictive control scheme. In Section 4, a numerical example is presented to show the benefits of applying predictive control at least two steps ahead in time, finalizing in Section 5 with analysis, comments and further research lines.

## **2. The Dynamic and Stochastic Vehicle Routing Problem: approaches and solution methods**

In this section, the objective is to provide a review on the most relevant dynamic and stochastic vehicle routing problem variants, intensely studied by different authors with different applications in the past 15 to 20 years. We are more interested in the distribution of passengers than in the distribution of goods, although some research by

Powell and his team in freight movement is also discussed here for the relevance of his work in our methodology.

We split this review into two subsections. First of all, we describe the different kind of problems, including both dynamic as well as stochastic VRP. Next, an analysis of solution algorithms (for both static and dynamic problems mostly heuristics and metaheuristics) is presented, closing this review with a description of the different approaches to formulate the DVRP, highlighting those contributions that use real-time information for predictive purposes.

### **2.1. Dynamic and Stochastic VRP**

First of all, we review the static stochastic VRP variants studied in the literature. Indeed, it is important to start with the stochastic version of the well known Traveling Salesman Problem (TSP), called the Probabilistic Traveling Salesman Problem (PTSP), where each customer node has a probability of requiring service. This problem was first studied by Jaillet (1985, 1988) and further investigated by many authors (Laporte et al., 1994; Bertsimas and Howell, 1993). Solving the problem consists of finding a tour of minimum expected length. Gendreau et al. (1996) identify the TSP with Stochastic Customers (TSPSC), the TSP with Stochastic Travel Times (TSPST) and the m-TSP with Stochastic Travel Times. The latter is the multivehicle version of the TSPST, where all vehicles start from the same depot.

Next, we move to the Probabilistic VRP (PVRP) described by Bertsimas et al. (1990) as a standard VRP, but with demands which are probabilistic in nature rather than deterministic. According to the author, such a problem is very hard to be solved. Bertsimas (1992) provides a recursive expression for finding the objective value, and also provides some bounds and asymptotic analysis and several re-optimization policies for the PVRP. Recent developments on such area are found in Bertsimas and Simchi-Levi (1996) and Gendreau et al. (1995).

A generalization of the PVRP is the Stochastic VRP (SVRP), in which some of the elements of the problem are stochastic (Gendreau et al., 1996). According to Gendreau et al., the SVRP can be also split into three categories: the VRP with stochastic demands (VRPSD), the VRP with stochastic customers (VRPSC) and the VRP with stochastic customers and demands (VRPSCD). In the VRPSD, the vehicle capacity is fixed, the customers are known but the service demand is stochastic (Gendreau et al., 1995; Yang et al., 2000; Secomandi, 1998, 2000). In the VRPSC the demand is deterministic, although the customer has certain probability of asking for service (Waters, 1989). The VRPSCD is a mixture of VRPSD and VRPSC (Bertsimas et al., 1990; Bertsimas, 1992; Gendreau et al., 1995, 1996).

The stochastic problems reviewed above are relevant for our purposes, since most of the methodologies used to formulate and solve stochastic problems are similar to those

utilized for dynamic problems, mainly when introducing probabilities to represent certain form of uncertainty.

Next, the dynamic problems are introduced where routes are constructed as unknown requests enter the system in real time. Unlike the static case, dynamic VRP problems are formulated assuming that inputs may change or have to be updated during the execution of the solution algorithm. According to Psaraftis (1988), in the dynamic vehicle routing problem the time dimension is essential, the problem may be open-ended, near-term events are more important, faster computation times are necessary and the objective function may be different compared with that utilized in static problems. In addition, hard time windows are normally replaced by soft time windows forced via penalties on the objective function, otherwise the likelihood of having not feasible solutions is very high.

Larsen (2000) develops a nice characterization of the different dynamic problems, starting again from the TSP, which yields the Dynamic TSP (DTSP) introduced by Psaraftis (1988). This work motivates the development of the Dynamic Traveling Repairman Problem (DTRP), introduced by Bertsimas and Van Ryzin (1991) and next extended by Bertsimas and Van Ryzin (1993). Lately Swihart and Papastavrou (1999), and Thomas and White (2004) formulate and solve two variants of the DTRP. Kleywegt and Papastravrou (1998, 2001), Papastravrou et al. (1996) study a problem called the Dynamic and Stochastic Knapsak Problem (DSKP), in which demands for a given resource occur according to some stochastic process.

The Dynamic Pickup and Delivery Problem (DPDP) that is designed to solve the Dynamic Dial-a-Ride Problem (DDRP) has been intensely studied in the last 20 years (Psaraftis, 1980, 1988; Gendreau et al., 1999; Kleywegt and Papastavrou, 1998). The final output of such a problem is a set of routes for all vehicles, which dynamically change over time. With regard to real applications Madsen et al. (1995) adapt the insertion heuristics by Jaw et al. (1986) and solve a real-life problem for moving elderly and handicapped people in Copenhagen, while Dial (1995) proposes a modern approach to many-to-few dial-a-ride transit operation ADART (Autonomous Dial-a-Ride Transit), currently implemented in Corpus Christi, TX, USA.

## **2.2. Solution methods and formulation approaches**

With regard to solution methods, we found a broad scope of studies that goes from analytical work (normally oriented to compute asymptotic behavior of deterministic dispatch policies computed from expected values for stochastic parameters) and exact methods for simplified cases to suboptimal heuristics and metaheuristics.

By exploring analytical methods and exact algorithms, Bertsimas and Van Ryzin (1991) develop a study based on geometrical probability, combinatorial optimization and queuing theory to derive lower bounds for the average service time in the context of the DTRP, and analyze several service policies (such as first come first served, stochastic

queue median, nearest neighbor, etc.) comparing their performance with the obtained lower bounds. Bertsimas and Van Ryzin (1993) extend their work to the case of homogeneous fleet where each vehicle is permitted to attend up to a fixed number of customers before returning to the depot. Swihart and Papastavrou (1999) minimize the expected travel time for a new variant of the DTRP, testing different vehicle capacity levels, finally deriving lower bounds for the average system time, under light and heavy traffic conditions. Bertsimas et al. (1990) provide bounds and asymptotical analysis for the PVRP, while Gendreau et al. (1995) propose an exact algorithm for the PVRP based on the integer L-shaped method (solving instances up to 46 vertices). They show that stochastic customers are much more difficult to treat than stochastic demands. Gendreau et al. (1996) also discuss an exact integer L-shaped method to solve the SVRP.

Jaillet (1985) formulates the PTSP as an integer nonlinear programming model, which is transformed to an integer programming model and solved using the branch and bound method for reasonably size problems and with probabilities close to one. Laporte et al. (1994) write the PTSP as an integer linear program, which is solved using the branch and cut method for 10 to 50 nodes. A successful real-time implementation of a static approach has been reported in Bell et al. (1983), for the routing and scheduling of a fleet delivering a bulk product stored at a central depot. The routing core of the procedure is the static algorithm of Fisher and Jaikumar (1982), which is based on a mixed integer programming formulation of the problem and a solution using Lagrangian relaxation and multiplier adjustment method.

Some authors formulate and solve their problems under a dynamic programming (DP) approach. Psaraftis (1980) extends the DP approach described for the static immediate request dial-a-ride to the dynamic case. The procedure is an open-ended sequence of updates, each following every new customer request. Jaillet (1985) shows that even though DP seems to be a natural way to solve the PTSP, such an approach does not allow finding the exact solution of the problem. Minkoff (1993) proposes a Markov decision model and decomposition heuristic for dynamic vehicle dispatching. Finally, Secomandi (1998, 2000) develops a stochastic shortest path formulation of the single vehicle VRPSD based on a Markov decision process, for small instances of a problem where customer demands are assumed to follow a discrete probability distribution.

Bertsekas and Tsitsiklis (1996) propose a neuro-dynamic programming model that allows solving the dynamic vehicle dispatching. Secomandi (1998, 2000) also propose a heuristics based upon neuro-dynamic programming for large VRPSD problems.

At this point, we also have to mention those algorithms based on an a priori solution (mostly by solving the static or deterministic version of the problem) combined with some adjustment heuristics to schedule real-time requests (or add the stochasticity in case of uncertain demand). Thus, Jaillet (1988) develops some improvement heuristics for the PTSP based on tour construction and tour improvement, resulting in a modified version of the 1-opt method. Bertsimas et al. (1990) propose several re-optimization



policies to solve the PVRP. Secomandi (1998, 2000) consider the VRPSD with known probability distributions to construct a 2-stage solution. Gendreau et al. (1996) illustrate a two-stage recourse methodology for the SVRP.

A commonly used adaptation of the above mentioned methodologies would be to handle dynamic input updates via a series of local operations, applied via the execution of an insertion heuristic (possibly followed by an interchange heuristic), after the static core algorithm is executed. This would involve running the static algorithm just to initialize the process, and rely on local operations for all subsequent input updates. Roy et al. (1984) propose an insertion heuristics for the transportation of the disabled, in which a fair amount of requests are known in advance (for them an initial solution is constructed). Real-time requests are incorporated in the initial solution framework using an insertion procedure. Madsen et al. (1995) develop another insertion approach for a demand responsive transportation system, where those requests known in advance are scheduled statically through an adaptation of the insertion algorithm of Jaw et al. (1986). Real-time requests are handled in a sequential fashion using an insertion rule that minimizes a weighted sum of components that measure the inconvenience to the new customer, the additional inconvenience to other customers already assigned to that route and various operations costs.

In the last few years, different meta heuristic and heuristic methods have been developed for solving various versions of the vehicle routing problem (Le Bouthillier and Crainic, 2005; Tarantilis, 2005; Tarantilis et al., 2005; Li et al., 2005; Bianchessi and Righini, 2006).

Among these heuristics, we have to mention the well-known Tabu search method, commonly used to solve static VRP problems. Gendreau et al. (1996) develop a modified tabu search version to solve the SVRP. Gendreau et al. (1999) modify the tabu search heuristics to solve the DVRP with soft time windows motivated from courier service applications, which is implemented in a parallel platform. Tabu search methods are derived in more sophisticated versions, such as granular tabu search (Toth and Vigo, 2003) and adaptive memory-based on Tabu search (Tarantilis, 2005).

Regarding heuristic methods for dynamic DVRP, Tighe et al. (2004) propose a priority based solver that considers sub problems of a real-time dynamic vehicle routing in order to obtain an optimal solution in less time by using also fuzzy decisions.

As VRP and VRP with time windows are NP hard, evolutionary techniques have been proposed to handle such problems. Specifically, Genetic Algorithms (GA) are applied for various VRP, considering different chromosome representation and genetic operators according to the particular problem (Skrlec et. al, 1997 for the single vehicle capacitated VRP; Haghani and Jung, 2005 for the multivehicle DVRP with time-dependent travel time and soft time windows). Moreover, Zhu (2003) describes specialized genetic algorithms based on adaptive parameters to solve the VRP with time windows

Jih and Yung-Jen (1999) for and Osman et al. (2005), present successful comparison of the GA against DP in terms of computation time. The former to solve the DVRP with time windows and capacity constraints while the latter to solve a Multiobjective VRP. Inclusive, a hybrid method including both algorithms is described, by obtaining good results in a reasonable time.

On the other hand, ant colony methods, as new metaheuristic inspired on the behavior of real ant colonies, have been applied to DVRP (Montemanni et al. 2005; Dréo et al. 2006). These methods are especially appropriate to efficiently solve combinatorial optimization problems and are characterized by the combination of a constructive approach and a memory based on learning mechanism.

For both, stochastic and dynamic problems, we can classify papers considering the approach the authors use to deal with the either dynamic or stochastic (or both) problem. Roughly speaking, we identify two clear tendencies: myopic and non-myopic approaches. The former research line does not consider explicitly the expected future information of the system to improve the current solution (all papers reviewed so far in this Section), while the latter option consider a mechanism to update information regarding the future to take better decisions at present. Such future data may be imprecise or unknown, and therefore developing consistent information update tools are essential for getting good predictions and take better real time dispatch decisions. That is one of reasons of why very few researchers solve stochastic or dynamic problems considering such a feature.

Powell and his team have worked for many years in a line of research that incorporates explicit stochastic and dynamic algorithms with the current information and probabilities of future events to produce more efficient solutions than those obtained through myopic deterministic strategies. They solve the problem of dynamically assigning drivers to loads that arise randomly over time motivated from long-haul truckload trucking applications.

Powell (1988) first considers the potential advantages of relocating vehicles in anticipation of future demands. He writes a two-stage stochastic program including a recourse function representing the future cost. Powell (1995) studies a mixed assignment and fleet management problem, modeled as a dynamic-stochastic network, and solved with a network simplex algorithm on a rolling horizon basis. Spivey and Powell (2004) propose a very general class of dynamic assignment models, and propose an adaptive, non-myopic algorithm that iteratively solves sequences of assignment problems. The authors also measure the benefits of using information in advance compared with myopic models. Topaloglu and Powell (2005) propose a distributed solution approach to a certain class of dynamic resource allocation problems. They develop a decision-making scheme, that under certain conditions and non linear functional approximations, produce sequences of min-cost flow problems that yield integer solutions.

Larsen (2000) in his thesis also investigated the use of future information, by relocating empty vehicles in anticipation to future demands. However, he concludes that using a-priori information in order to be able to reposition the vehicles expecting to receive new requests does not seem to offer significant performance improvements with respect to the lateness experienced by the customers. Finally, Ichoua et al. (2005), develop a strategy based on probabilistic knowledge about future request arrivals to better manage the fleet of vehicles for real-time vehicle dispatching. The new strategy introduces forecasted customers in vehicle routes to provide a good coverage of the territory. The strategy is assessed through computational experiments performed in a simulated environment, and is solved using a parallel tabu search technique.

The analysis of these models that incorporate future information is crucial for our purposes, since what we are trying to do in this proposal is to formalize the use of future information in dynamic vehicle routing problems through a hybrid predictive control scheme. In the next section, such a scheme is presented in detail.

### **3. Hybrid predictive control approach to solve the multivehicle dynamic pickup and delivery problem (MDPDP)**

In the context of control theory, the notion of hybrid systems arise when the problem conditions are characterized by both continuous and discrete/integer variables. In the last decade, hybrid systems have been studied more intensely by researchers from several study areas, such as computer science and automatic control. A systematic methodology for a general control design of hybrid systems has been developed by Bemporad et al. (1999) and Bemporad et al. (2002).

Specifically, hybrid systems can be expressed as a non linear state space system given by

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k)) \end{aligned} \tag{1}$$

where  $x(k)$  are the continuous and/or discrete (integer) state space variables,  $u(k)$  are the continuous and/or discrete input or manipulated variables and  $y(k)$  define the continuous and/or discrete system outputs. In general, a hybrid predictive control design minimizes the following generic objective function:

$$\min_{\{u(k), u(k+1), \dots, u(k+N-1)\}} J(u(k), \dots, u(k+N-1), \hat{x}(k+1), \dots, \hat{x}(k+N), \hat{y}(k+1), \dots, \hat{y}(k+N)) \tag{2}$$

where  $J$  is a objective function,  $k$  is the current time,  $N$  the prediction horizon,  $\hat{x}(k+t)$ ,  $\hat{y}(k+t)$  are the expected state space vector and the expected system output at instant  $k+t$  respectively, and  $\{u(k), \dots, u(k+N-1)\}$  represents the control sequence, which corresponds to the vector of optimization variables.

Once expression (2) is optimized, only the first element of the control vector  $u(k)$  is used to update the system conditions, based upon the receding horizon methodology. This optimization problem that combines integer/discrete and continuous variables is typically solved by several numerical methods such as Branch & Bound, Generalized Benders, etc. (Floudos, 1995). Next, we characterize the multivehicle dynamic pickup and delivery problem (MDPDP) as a hybrid system showing the advantages of this approach when predicting future conditions under unknown dynamic demand patterns.

### 3.1 Problem statement

The core of this paper is to formulate a generic MDPDP as a hybrid predictive control problem, following the theory explained above, recognizing that the dynamic routing process behind the real time dispatch decisions includes discrete/integer and continuous state space variables as well as discrete input variables.

Specifically, in this paper we develop a framework that incorporates stochasticity into the routing dispatch rules in the context of the dynamic passenger pickup and delivery problem. The focus here is in passenger routing, however the scheme could be generalized to freight too.

The formulation is based on the following premise: the expected travel (or waiting) time that all the already-scheduled users will incur during their trip will be strongly affected by any future reassignment of the vehicle assigned to pick them up at their origin spot, or drop them at their destination in case they are already on the vehicle. We introduce stochastic prediction into the routing rules deciding the assignment in order to incorporate a more realistic measure of travel (waiting) time experienced by the users as well as the operator into the decision cost formulation, which eventually could change some of the dispatching module decisions, resulting in better solutions closer to the desired dynamic optimum.

The idea of using a hybrid predictive control scheme is to compute more accurate dispatching cost functions under stochastic demand, generated dynamically over time without any previous knowledge of these new requests by the dispatching module at the time of the call. Under such a modeling scheme, the use of past information regarding the dynamics of the system to predict behavior, should result in better routing decisions, provided that the proposed objective function captures the relevant user cost components (see section 3.3 for details).

Let us assume an influence area  $A$ , with a transit service network of length  $D$  in distance units. Suppose we have a set of vehicles  $V$  of size  $F$ . The fleet of transit vehicles is currently in operation traveling within the area according to predefined routing rules. The demand for service is unknown and is generated dynamically in real-time (assume a rate  $\mu$  in calls per time units). Routing and scheduling decisions have to be taken in real time, to handle such demand with the available vehicles.

At any time  $k$  we assume that each vehicle  $j \in V$  has been assigned a sequence of tasks that include pickups and deliveries, and can be represented by a function  $S_j(k) = [s_j^0(k) \ s_j^1(k) \ \dots \ s_j^i(k) \ \dots \ s_j^{w_j}(k)]^T$ , in which the  $i^{\text{th}}$  element of the sequence represents a specific  $i^{\text{th}}$  stop along vehicle  $j$ 's route,  $w_j$  is the number of stops. The initial condition (denoted by  $s_j^0(k)$ ) updates the position of vehicle  $j$  at instant time  $k$ . The complete control action or manipulated variable  $u(k) = S(k)$  can be visualized as the dispatching module, which takes the dynamic routing decisions. At time  $k$ , these control actions are represented by the set of sequences assigned to every vehicle at instant  $k$ . Analytically,

$$u(k) = S(k) = [S_1(k), \dots, S_j(k), \dots, S_F(k)] \quad (3)$$

Vehicles will travel according to the predefined sequence vector  $S(k-1)$  while no new calls are received. When a new service request (call) comes in, the controller or central dispatcher calculates the sequences in the next step  $S(k)$  (control action) for the fleet of vehicles including the stops requested by the new customer. Then, each sequence  $S_j(k)$  remains fixed during the whole time interval  $(k, k+1)$ , unless a vehicle reaches a predefined pickup or delivery stop during such an interval, in which case its sequence will decrease in size showing that the scheduled task has been accomplished.

The dispatch decisions are taken by the controller in real time based on a specific insertion-type criterion at the dispatching module. In this application, the rule is to choose  $S_j(k)$  with the minimum incremental insertion cost, by minimizing an objective function  $J$ , based upon passenger's waiting and travel time. The final control action is to insert the new request into the best feasible sequence  $S_j^*(k)$ .

Traditionally used fixed time steps are not directly applicable in this case, since the predictive controller takes a routing decision at the moment a new call enters the system. Hence, in this scheme it is necessary to formulate the problem in terms of a variable time step, which represents the variable time interval between two consecutive requests.

The dynamic routing decisions associated with the cost function specification (see section 3.3) strongly depend on the state of the system at instant  $k$  (i.e., when request  $k$  enters the system). Vehicle attributes include all vehicle features when leaving the  $i^{\text{th}}$  stop of its sequence. In this formulation the vector of attributes include the clock time of departure  $T_j^i(k)$  and the vehicle load  $L_j^i(k)$ , after vehicle  $j$  leaves stop  $i$ . In a more general context, other variables could be added to obtain a better picture of vehicle states, such as vehicle  $j$  speed capturing non-recurrent traffic congestion, the accumulative travel-time experienced by all passengers on board after leaving stop  $i$ , and some measure of the available space in other vehicles for potential passengers in the proximity of segment  $(i, i+1)$  (see Cortes, 2003; Cortés and Jayakrishnan, 2004 for details).

A graphical representation of a typical vehicle sequence computed at certain instant  $k$  is shown in Figure 1.

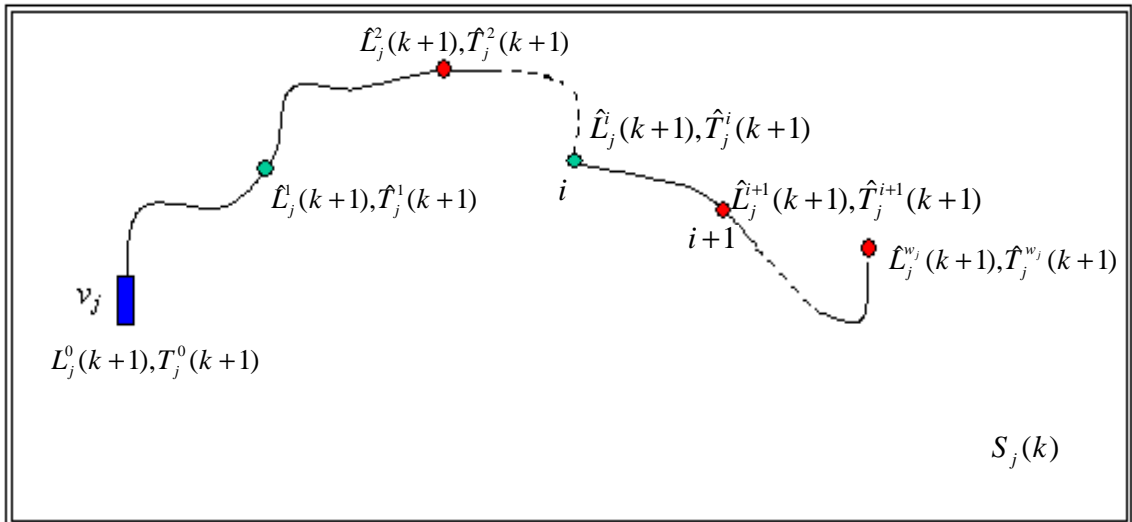


Figure 1 Typical vehicle route at time  $k$

As mentioned before, when a new call enters the system, the controller or central dispatcher decides which vehicle in service will take the new request based on a cost function depending on an estimation of the user waiting and travel times. The effective waiting and travel times can be calculated only when the request is completely satisfied (pickup and delivery).

The output vector  $y(k) = T(k) = [T_1(k), \dots, T_j(k), \dots, T_F(k)]$  is computed by measuring the observed segment travel times  $T_j^{(i,i+1)}(k) = T_j^{i+1}(k) - T_j^i(k)$ , occurring until a new call occurs for all vehicles and for all pair of stops  $(i, i+1)$ . The dynamic dispatching decision depends upon a cost function in terms of segment travel times

$T_j^{(i,i+1)}(k)$  as major components (see section 3.3 for details). Therefore, we could utilize the effective passenger and vehicle travel times as well as the effective passenger waiting times, as good indicators of the performance of the controller or central dispatcher.

Analytically, equation (1) can be written for the MDPDP, by recognizing the dependence of the routing process and the following associated variables:

$$\begin{aligned} x(k) &= [L(k), T(k)]^T \\ y(k) &= T(k) \\ u(k) &= S(k) \end{aligned} \quad (4)$$

where

$$\begin{aligned} T(k) &= [T_1(k), \dots, T_j(k), \dots, T_F(k)] \\ L(k) &= [L_1(k), \dots, L_j(k), \dots, L_F(k)] \end{aligned} \quad (5)$$

As discussed in the next section, the functions  $f$  and  $g$  in equation (1) are nonlinear for the proposed routing system (see section 3.2 for details).

The hybrid predictive control scheme proposed in this paper can be represented by a generic flow chart in figure 2 next.

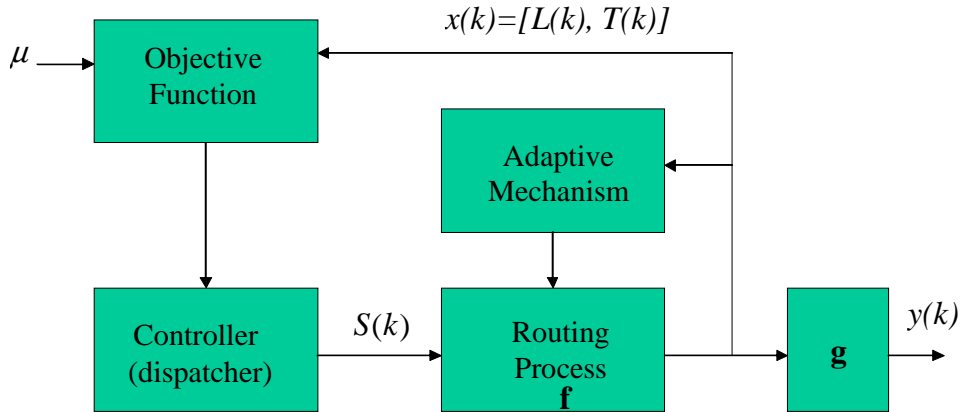


Figure 2. Overall block diagram of a Hybrid Predictive approach for computing Stochastic Rerouting Delay

In the figure, the hybrid predictive controller is represented by the dispatching module, which takes routing decisions in real time based on the information it has from the routing system (process) and the expected values for travel times and attributes of its vehicle fleet (model). An adaptive mechanism is also added in the figure, representing the need of adapting the parameters of the model. Next, the discussed state space model

is analytically described, highlighting the treatment of both the departure time and load components. Hereafter, the objective is to define the specific state space formulation for the dynamic pickup and delivery problem focused on passengers, as described above in this section.

### 3.2 Predictive dynamic model

In this section, we start discussing the functions describing the state space components in expressions (4) and (5). This research considers hybrid state space models for both the vehicle load and the departure time at stops (as a function of segment travel times). Basically, the idea is to explicitly find expressions for such functions.

Both the clock time of departure  $T_j^i(k)$  and the vehicle load  $L_j^i(k)$  are stochastic variables, because they depend on the evolution of the system and in that sense their values change according to the dispatcher routing decisions, which are unknown in advance. Therefore, and in order to work with deterministic values, reasonable estimations of the load and departure time vectors have to be obtained. Analytically,

$$\hat{x}(k+1) = \begin{bmatrix} E\{L(k+1)/k\} \\ E\{T(k+1)/k\} \end{bmatrix} = \begin{bmatrix} \hat{L}(k+1) \\ \hat{T}(k+1) \end{bmatrix} = \begin{bmatrix} f_L(L(k), S(k)) \\ f_T(T(k), S(k)) \end{bmatrix} \quad (6)$$

where E denotes the expected value of the state space vector. Note that the functions  $f_L$  and  $f_T$  normally include matrices and vectors in their specifications, as usual in most state space models. In this case however, most matrices and vectors change in dimension from iteration to iteration. In fact, to capture the dynamic effect caused by the real operation, an expansion and reduction matrix size technique is developed (adaptive behavior). The idea is to either increase or reduce the vehicle sequence matrix, resulting in changes on the load and time vectors as well. For example, when certain vehicle accepts a new service request, the dimension of the sequence matrix increases in two rows, accounting for the customer pick-up and delivery stops. Additionally, when a vehicle reaches any stop, that point has to be removed from the original sequence, reducing its dimension in two rows.

A detailed representation of the manipulated variable  $S(k)$ , defined in equation (3), is shown in expression (7) next:

$$u(k) = S(k) = [S_1(k), \dots, S_j(k), \dots, S_F(k)]$$



$$S_j(k) = \begin{bmatrix} p_j^0(k) & d_j^0(k) & t_j^0(k) & label_j^0(k) \\ p_j^1(k) & d_j^1(k) & t_j^1(k) & label_j^1(k) \\ \vdots & \vdots & \vdots & \vdots \\ p_j^i(k) & d_j^i(k) & t_j^i(k) & label_j^i(k) \\ \vdots & \vdots & \vdots & \vdots \\ p_j^{w_j+2}(k) & d_j^{w_j+2}(k) & t_j^{w_j+2}(k) & label_j^{w_j+2}(k) \end{bmatrix}_{(w_j+3) \times 4} \quad (7)$$

The first and second columns represent the pickup  $p_j^i(k)$  and delivery  $d_j^i(k)$  at stop  $i$  respectively, where

$$p_j^i(k) = \begin{cases} 1 & \text{if } label_j^i \text{ is a pick-up stop} \\ 0 & \text{Otherwise} \end{cases}$$

$$d_j^i(k) = \begin{cases} 1 & \text{if } label_j^i \text{ is a delivery stop} \\ 0 & \text{Otherwise} \end{cases}$$

Note that  $p_j^i(k) = 1 - d_j^i(k)$ . The third column of the  $S_j(k)$  matrix represents the external travel time function, where  $t_j^i$  is the total travel time between points  $i-1$  and  $i$ . This measure of travel time between any two points of the network is external to the process, and we assume that it is available to the dispatcher at any time. For simplicity, we assume that the average delay associated to a pick-up or a delivery operation at nodes is approximately constant, and is added to segment travel times. Thus,  $t_j^i$  is computed as the network travel time on segment  $(i-1, i)$  plus the transfer operation delay at node  $i$ .

For simulation purposes, we assume that vehicles move at constant speed, and therefore their position can be estimated at any moment. The size of the sequence matrix in (7) is  $(w_j + 3) \times 4$ , comprising one row for the initial condition,  $w_j$  rows for the previously scheduled stops, and two final rows with the information (pickup and delivery locations) of the last call. Finally, the last column  $label_j^i$  keeps the passenger identifier.

Thus, the vehicle load behavior is obtained using the following state space model:

$$E\{L(k+1)/k\} = \hat{L}(k+1) = A_L L(k) + B_L(S(k)) \quad (8)$$

where the load vector, can be decoupled into the following components:

$$L_j(k) = \begin{bmatrix} L_j^0(k) & L_j^1(k) & \cdots & L_j^{w_j}(k) \end{bmatrix}_{(w_j+1) \times 1}^T \quad (9)$$

As mentioned above,  $L_j^i(k)$  represents the estimated vehicle  $j$  load, when it leaves stop  $i$ , estimated at time  $k$ , and  $w_j$  is the number of stops belonging to a desired vehicle sequence. The corresponding matrices of the state space model, defined in equation (8) are computed as follows

$$B_L(S(k))_{(w_j \times 3) \times 1} = B_L^2 \cdot (S(k) \cdot B_L^1)$$

$$A_L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)} \quad ; \quad B_L^1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}_{1 \times 4}$$

and

$$B_L^2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

The vehicle sequence matrix  $S(k)$  and the expected load vector  $\hat{L}(k+1)$ , could change their dimension dynamically. Therefore, the matrix dimensions of  $A_L, B_L^1, B_L^2$  are variable and depend on the vehicle sequence matrix, obtaining an adaptive state space model.  $B_L^1$  is designed to remove the two last columns of the sequence vector, which are not necessary for representing load changes from step to step.

The vehicle departure time behavior is obtained by using the same methodology. Analytically

$$E\{T(k+1)/k\} = \hat{T}(k+1) = A_T \cdot T(k) + B_T(S(k)) \quad (10)$$

The departure time vector is decoupled as follows

$$T(k) = \begin{bmatrix} T_j^0(k) & T_j^1(k) & \cdots & T_j^{w_j}(k) \end{bmatrix}_{(w_j+1) \times 1}^T \quad (11)$$

where  $T_j^i(k)$  represents the vehicle  $j$  departure time at stop  $i$  estimated at time  $k$ , and  $w_j$  is the number of stops belonging to the desired vehicle sequence. The corresponding matrices of the state space model defined in equation (10) are:

$$B_T(S(k))_{(w_j+3) \times 1} = B_T^1 \cdot (S(k) \cdot B_T^2)$$

$$A_T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)} ; B_T^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{4 \times 1}$$

$$B_T^1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

Like the load state space model, the matrices  $A_T, B_T^1, B_T^2$  could change their dimensions dynamically, depending on the vehicle sequence matrix. The sequence feature regarding travel time between stops is required for computing departure time from stops.

For both models, the label is required to identify the matching between a specific pick up and delivery in order to check the feasibility of the sequence in terms of precedence. These last definitions complete the adaptive state space model.

### 3.3. Objective Function

In the last section, the problem-specific state space formulation is analytically developed. Here, the concept of cost function is added in order to have a performance measure for deciding the optimal predicted vehicle routes by the controller. In this case, we consider both total expected waiting and travel time for passengers. The idle travel time (vehicles moving around without passengers) is also included in the formulation, as explained next.

The major issue in the definition of the objective function is to define a reasonable horizon for prediction  $N$ , which depends on the studied problem, and also on the intensity of the unknown events entering the system in real time. In cases where the decision is taken at instant  $k$ , but considering a predictive horizon greater than one, the decision maker (controller) adds the predictive feature into the formulation, since decisions made in  $k+1$  will depend on possible events (new service requests) occurring in future instants ( $k+2, k+3, \dots$ etc). Thus, the central dispatcher (controller) computes the decisions for the entire control horizon  $N$ , i.e.,  $\{S(k), \dots, S(k+N-1)\}$ , and applies just the next step sequence  $S(k)$ , based on receding horizon control. The routing decisions will depend on how well the system predicts the impact of rerouting passengers due to unknown insertions. The estimation of the parameters of the model

requires data obtained from the performance of the system for a time period  $H_e$  measured backward since the occurrence of event  $k$  (online model). This parameter  $H_e$  is defined arbitrarily by the controller in order to perform a robust and representative estimation of the stochastic model embedded in the definition of the dynamic routing decision. Another way to deal with the estimation procedure is to use historical information referred to a representative period similar to the period in which the controller is taking dispatch decisions (offline model).

The objective function for a generic prediction horizon  $N$  can be written as follows

$$\underset{\{S(k), S(k+1), \dots, S(k+N-1)\}}{\text{Min}} \quad J = \sum_{t=1}^N \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h^{\Delta T(k+t)} \cdot \left( (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2), h} \right) \quad (12)$$

$$C_j(k+t) \Big|_{S_j(k+t-2), h} = \sum_{i=1}^{w_j(k+t)} \left\{ \underbrace{\left[ \hat{L}_j^{i-1}(k+t) + 1 \right] \left( \hat{T}_j^i(k+t) - \hat{T}_j^{i-1}(k+t) \right)}_{\text{J travel time}} + \underbrace{r_j^i(k+t) \alpha \left( \hat{T}_j^i(k+t) - T_j^0(k+t) \right)}_{\text{J waiting time}} \right\} \Big|_{S_j(k+t-2), h} \quad (13)$$

where  $k+t$  is the instant at which the  $t^{\text{th}}$  request enters the system, measured from time interval  $k$ .  $h_{\max}(k+t)$  is the number of probable requests at instant  $k+t$ ,  $p_h^{\Delta T(k+t)}$  is the probability of occurrence of the  $h^{\text{th}}$  request type (associated to a specific pair of zones, as discussed ahead in this paper) during time interval  $\Delta T(k+t)$ , noting that  $\Delta T(k+t)$  specifies the time interval to which time step  $k+t$  belongs.  $C_j(k+t) \Big|_{S_j(k+t-2), h}$  in (13) is the cost function of vehicle  $j$  at instant  $k+t$ , which depends on the previous sequence at  $k+t-1$ ,  $S_j(k+t-2)$  and a new potential request  $h$  with probability  $p_h^{\Delta T(k+t)}$ .  $w_j(k+t)$  is the number of stops estimated for vehicle  $j$  at instant  $k+t$ . Thus,  $S_j(k+t-2), h$  is the new sequence provided that  $h$  occurs.

It is important to mention here that fixed time steps are not directly applicable to the proposed model. In fact, for this problem a variable time step is required, and a good estimate of such a parameter is crucial for developing a consistent and powerful predictive model. In fact, the algorithm could seem not to work properly mainly because

of using a wrong stepsize. By taking a deep look into the model, we realize that the stepsize should represent the expected time at which a new request would enter the system from the current decision time. It is clear that this parameter varies over time, and strongly depends on the system conditions at each period. This problem must not be underestimated as pointed out by George and Powell (2005), who develop and discuss many interesting methods to incorporate a good estimation of such a parameter (like Kalman Filter and others).

We realize that none of these methods properly replicate the MDPDP conditions, considering that in addition to represent a good estimation of the time between calls, what we really want to calibrate is a parameter for optimizing the system performance function over time, in order to get the optimal routing strategy including future information. For doing that, a sensitivity analysis was conducted from simulated data in order to find the stepsize value that minimizes the cost function for more than one step ahead. It is very important to highlight the fact that these variables are continuous and non-optimal behavior could occur if they are not properly adjusted by sensitivity analysis. For the two-step ahead application (see section 4), this parameter is called  $\tau$  in our notation, and as discussed above, physically it represents the expected time for a predicted request to happen. However, what  $\tau$  really represents is the best instant for inserting the future expected call in order to optimize the routing scheme. In general, these parameters are tunable for each step ahead of prediction.

As mentioned before, the cost function  $C_j(k+t)$  as shown in (13), can be split into two pieces: a waiting time and a travel time component. Both of them are written as function of the load and departure time and they are computed as the departure time between consecutive stops times the vehicle load (represented by the number of passengers plus the vehicle driver). For the sake of flexibility and economic consistency, the waiting cost component is weighted by a coefficient  $\alpha$ . Analytically,  $\hat{L}_j^{i-1}(k+t)$  denotes the expected load over segment from stop  $i-1$  to  $i$ , the difference  $\hat{T}_j^i(k+t) - \hat{T}_j^{i-1}(k+t)$  measures the expected vehicle travel time on segment  $(i-1, i)$ , including the transfer delay at node  $i$ , and the difference  $\hat{T}_j^i(k+t) - T_j^0(k+t)$  measures the vehicle expected travel time to reach stop  $i$  from its current position plus the expected transfer delay at node  $i$ .  $r_j^i(k+t)$  is a binary variable corresponding to a waiting time factor defined as

$$r_j^i(k+t) = \begin{cases} 1 & \text{if stop } i \text{ belonging to } S_j(k+t-1) \text{ is a pick-up} \\ 0 & \text{Otherwise} \end{cases}$$

Note that in the first component of the cost function expression in (13), the expected travel time is weighted by  $\hat{L}_j^{i-1}(k+t)+1$ . In such a computation, the expected load captures the user cost associated to travel time, while the added *one* roughly incorporates

a *proxy* for the operational cost through the total time traveled by vehicles, even though some of them do not carry any passenger on certain segments of their routes.

The problem in (12) is to minimize  $J$  with respect to the sequence set  $\{S(k), S(k+1), \dots, S(k+N-1)\}$  to then consider the optimal sequence  $S^*(k)$  for the imminent decision for dynamic routing of the fleet.

The probabilities of occurrence of each scenario are parameters in the objective function, and they are computed based on either real-time data, historical data, or a combination of both. In this particular application, we use a simple way to compute these probabilities from historical data (off line implementation). For doing that, let us define the call mass center as the geographical location of the most likely call to occur during a specific time period, and within a specific area. As described in the problem formulation, what we really need is the probability that the expected new request will occur between two specific zones (pick-up and delivery) within certain time interval.

As previously defined, the new call has a probability  $p_h^{\Delta T(k+t)}$  of occurring between a specific pair of zones  $h$  at time step  $k+t$  (belonging to a time interval  $\Delta T(k+t)$ ). In order to apply this methodology, the zone of study has to be split into smaller sub-areas (clusters). How to choose the zoning will depend upon the demand intensity associated to each specific problem. In the application section 4, the probability that a new call will appear for a specific pair of clusters  $h \subseteq H$  within a time interval  $\Delta T(k+t)$  is computed using the following expression:

$$p_h^{\Delta T(k+t)} = \frac{N_h^{\Delta T(k+t)}}{\sum_{g \in H} N_g^{\Delta T(k+t)}} \quad (14)$$

where  $N_h^{\Delta T(k+t)}$  is the total number of travel requests belonging to a specific origin-destination pair of clusters  $h$  over a set of pairs  $H$ , within a specific time interval  $\Delta T(k+t)$ . Note that  $\sum_{h \in H} p_h^{\Delta T(k+t)} = 1$  as expected.

Next, the two particular cases computed in the simulation example in section 4 are highlighted. A myopic strategy (one step ahead) is developed, to be compared with a two-step ahead predictive approach, which is the simplest algorithm including future information from the system, in order to show the importance of considering a predictive component when taking routing decisions in a dynamic system like MDPDP.

#### *One-step ahead algorithm*

In this case, the cost function for a given vehicle  $j$  depends upon the load as well as departure time (both described in the previous section). The one step ahead strategy

means that the prediction horizon is  $N = 1$ , and  $h_{\max}(k+1) = 1$  since the new requirement is one and known, and therefore its probability is equal to 1, obtaining the following expression for the objective function

$$\begin{aligned}
J &= \sum_{t=1}^1 \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h^{\Delta T(k+t)}(k+t) \cdot (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2),h} \\
&= \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+1)=1} p_h^{\Delta T(k+1)}(k+1) \cdot (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),h} \\
&= \sum_{j=1}^F \overbrace{p_1^{\Delta T(k+1)}(k+1)}^{=1} \cdot (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1} = \sum_{j=1}^F (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1}
\end{aligned} \tag{15}$$

The difference  $(C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1}$  means that the cost is evaluated considering the control action in the previous instant, represented by  $S_j(k-1)$ . The optimization problem is written as follows:

$$\underset{S(k)}{\text{Min}} J = \sum_{j=1}^F (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1),1} \tag{16}$$

$$\begin{aligned}
&C_j(k+1) \Big|_{S_j(k-1),h} = \\
&\sum_{i=1}^{w_j(k+1)} \left\{ \underbrace{\left[ \hat{L}_j^{i-1}(k+1) + 1 \right] \left( \hat{T}_j^i(k+1) - \hat{T}_j^{i-1}(k+1) \right)}_{J \text{ travel time}} + \underbrace{r_j^i(k+1) \alpha \left( \hat{T}_j^i(k+1) - T_j^0(k+1) \right)}_{J \text{ waiting time}} \right\} \Big|_{S_j(k-1),h}
\end{aligned} \tag{17}$$

Expression (16) can be computed for each feasible vehicle sequence  $S(k)$  at time  $k$  by using an explicit enumeration optimization method. The optimal sequence matrix  $S^*(k)$  is obtained by comparing the cost in (17) among all feasible sequences, since  $C_j(k)$  turns out to be known and constant in (16), for all vehicles. Analytically,

$$S^*(k) = \arg \min \left\{ J = \sum_{j=1}^F (C_j(k+1) - C_j(k)) \Big|_{S_j(k-1)} \right\} \tag{18}$$

Conceptually,  $J$  represents the insertion cost when the system accepts a new call, computed in real time and considering the entire vehicle fleet. Note that there are many

possible alternatives to insert the new request. Thus, the vehicle sequence finally chosen by the controller is obtained by solving (18).

### 3.3.1 Two-step ahead algorithm

The two step prediction ahead cost function is slightly different from the previous one, in the sense that now a computation of a closed expression for  $J_j$  is not straightforward as before, since we do not know with certainty the position of the call that will enter the system two steps ahead. However, in the formulation we postulate that the decision for the imminent assignment must depend on the potential future insertions. A simple probabilistic approach is used in order to incorporate the most likely position of the next potential call into the cost function expression (see equation (12)). A distribution for the time interval between successive calls is also assumed in order to compute time interval probabilities as explained in the section 3.1.

The two step ahead hybrid predictive controller selects the vehicle's sequence that minimizes the general two step ahead cost function. This cost function represents the potential insertion cost when a possible new call will appear near to the spatial vehicle trajectory into a specific interval time given by a probability  $p_h^{\Delta T(k+2)}$ .

The corresponding expression for the cost function in this case is as follows:

$$\begin{aligned}
J &= \sum_{t=1}^2 \sum_{j=1}^F \sum_{h=1}^{h_{\max}(k+t)} p_h^{\Delta T(k+t)}(k+t) \cdot (C_j(k+t) - C_j(k+t-1)) \Big|_{S_j(k+t-2),h} = \\
& \sum_{j=1}^F \left[ C_j(k+1) \Big|_{S_j(k-1),1} - C(k) + \sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot \overbrace{C_j(k+1) \Big|_{S_j(k-1),1}}^{\text{Independent of h}} \right] = \\
& \sum_{j=1}^F \left[ \sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot C_j(k+2) \Big|_{S_j(k),h} - \overbrace{C(k)}^{\text{known constant}} \right]
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
C_j(k+2) \Big|_{S_j(k),h} &= \\
& \sum_{i=1}^{w_j(k+2)} \left\{ \underbrace{\left[ \hat{L}_j^{i-1}(k+2) + 1 \right] \left( \hat{T}_j^i(k+2) - \hat{T}_j^{i-1}(k+2) \right)}_{J \text{ travel time}} + \underbrace{r_j^i(k+2) \alpha \left( \hat{T}_j^i(k+2) - T_j^0(k+2) \right)}_{J \text{ waiting time}} \right\} \Big|_{S_j(k),h}
\end{aligned} \tag{20}$$



The optimization problem for both one and two step ahead, is solved by using explicit enumeration that considers all feasible insertion solutions whenever a call request enters the system. In order to have a manageable problem, we restrict the solution to keep the order of the previous sequence for all vehicle, and test the insertion of the new call on those routes without changing their original order (no swapping). We have to highlight that this paper is focused more on the development of a formal predictive control framework to model the problem than on the solution algorithms. Nevertheless, we are currently developing better solution methods that will allow us to relax such a constraint (refer to Section 5 for a discussion on this issue).

For illustrative purposes of the proposed methodology (see Figure 3), let us concentrate on the two step-ahead prediction case (no swapping) for an example of two origin-destination pairs, in which the strategy would be to evaluate the following chain of scenarios. At instant  $k-1$ , vehicles follow certain sequence vector  $S(k-1)$  associated with a total cost  $C(k)$ . Whenever a new service request enters the system, there are several feasible sets of sequences  $S(k)$  to be evaluated by the controller (each alternative inserting the new pick-up and delivery in feasible segments of the sequence of a specific vehicle). At one-step ahead, one call is considered (instant  $k$  with probability equal 1). At two-step ahead, we fix two potential calls appearing in the next time step  $k+1$ , with probabilities  $p_1(k+2)$  and  $p_2(k+2)$  respectively, in order to incorporate the dynamic nature of the problem, and consequently to have good estimations of both travel and waiting times for the cost function decision. Finally, two potential cases are evaluated for all possible scenarios, containing two new sequential insertions each (the known new call that comes up and the potential call that appears two steps ahead). If  $w_j$  is the number of stops in vehicle  $j$  sequence including the new request at instant  $k$ , the number of possible sequences for that vehicle one step ahead, containing those combinations not satisfying the vehicle capacity constraints, is equal to  $(w_j - 1)w_j/2$ . For the whole fleet,

this number increases to  $\sum_{j=1}^F (w_j - 1)w_j/2$ . In case of evaluating the two step ahead options, again not fulfilling the vehicle capacity constraints, the number of combinations to be evaluated for one vehicle is  $(w_j + 1)(w_j + 2)/2$ . Therefore, for the entire fleet, the number of options to be considered is the product of the combinations one step ahead and those arising two steps ahead, which turns out to be

$$\frac{1}{4} \left( \sum_{j=1}^F \sum_{i \neq j}^F (w_j - 1)(w_i - 1)w_j w_i + \sum_{j=1}^F (w_j - 1)w_j (w_j + 1)(w_j + 2) \right).$$

All these previous computations are assuming a no-swapping operation, which means, considering that the new requests will be inserted into the previous sequence without changing the original order of vehicle tasks.

Graphically,

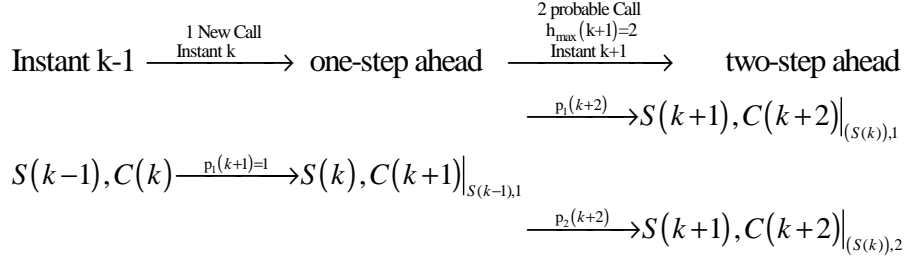


Figure 3. Possible combinations of sequences at future.

In the next section, a strict simulated experiment is performed and reported in detail, but here we present a simple example for a single vehicle to graphically show the predictive power of our approach to improve performance of dynamic dispatch decisions in real-time routed transit services. The idea is to show the potential benefits of the two-step ahead over the one-step ahead algorithm, which can be generalized to more complex cases as shown in the next section. In this example, the vehicle sequence for three instants ( $k-1$ ,  $k$  and  $k+1$ ) is depicted in Figure 4, considering that decisions were taken by the central dispatcher using both proposed algorithms. The original vehicle sequence starts from the initial vehicle position (0.8,0.2) at instant  $k-1$  9:00 AM, then one passenger is picked up ( $1^+$ ) and next, is delivered ( $1^-$ ) as shown in figures 4 a) and b).

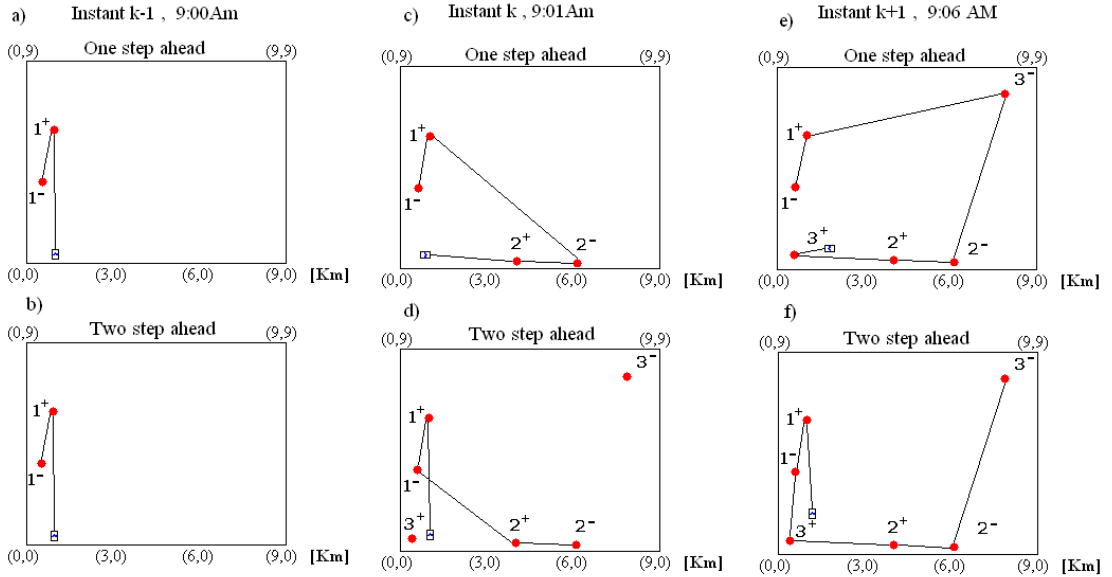


Figure 4. Single vehicle sequences at  $k-1$ ,  $k$  and  $k+1$  time steps

When the vehicle is located at (0.79,0.53) at instant  $k$  (9:01 AM), a new call enters the system with origin ( $2^+$ ) and destination ( $2^-$ ). The optimal assignment by the central dispatcher when running the two algorithms is shown in Figure 4, c) and d). Also, at instant  $k+1$  (9:06 AM) a new request for service appears with origin ( $3^+$ ) and destination

(3). In Table 1 next, the expected cost function values at each instant for the two algorithms is reported:

Table 1. One-step ahead versus two-step ahead expected performance function

	$\hat{J}(k-1)$	$\hat{J}(k)$	$\hat{J}(k+1)$
<b>One step ahead</b>	45.7163	103.6254	216.4958
<b>Two step ahead</b>	45.7163	107.9561	203.9334
<b>Savings [min]</b>	0	-4.3307	12.5624

Note that the predictive control algorithm shows a slightly higher cost for the first simulation step than that obtained from the one-step ahead method (103.6 vs. 107.96), which could be interpreted as preferring the one step ahead option over the two step ahead one for the time  $k$  decision. However, this apparent disadvantage of the two-step ahead method is compensated when an additional call enters the system in  $k+1$ . (See Figures 4e) and 4f)). In  $k+1$ , from table 1, the  $\hat{J}(k+1)$  cost shows considerable (216.5 v/s 203.9), since the system considers the effect of the potential call request 3 (predictive feature). The better the guess of where the new call will appear, the better the performance of running the predictive algorithm against the one step ahead counterpart.

In summary, the Hybrid Predictive Control general structure is applied to the specific problem of rerouting transit vehicles in real time, under stochastic and unknown demand generated dynamically. The method described here includes discrete controller decisions and input/output discrete and continuous variables, resulting in a new hybrid form of conventional techniques. In this paper, we develop a detailed modeling approach to solve the problem. In addition, unknown parameters have been calibrated for estimating expected vehicle travel times in real time, assuming random demand generated dynamically (following a real case pattern), and a mixed integer non linear optimization algorithm for optimized decision at the dispatching module. The framework is developed for application to a real-time large scale transit fleet routing problem, but has applicability in other contexts as well.

In the next section, a simulation experiment is carried out in order to test the methodology and measure how important is to incorporate prediction within the dispatching rules for this kind of service through a reasonable cost function expression.

## 4. Simulation Test

### 4.1 Experiment description

A discrete-event system simulation for a three-hour period is conducted in order to evaluate the performance of the proposed dispatch control algorithm for a dynamic vehicle routing system. The scheme considers a fleet of nine transit vehicles, each with

capacity for four passengers. Dispatch decisions are taken in real-time by the controller. Service requests are unknown, however the average system pattern is supposed to be known from historical data, obtained from the average demand measured over a week before or so.

The simulation scenario is not real; however the demand patterns follow a heterogeneous distribution inspired on real data from the Origin-Destination Survey in Santiago, Chile, 2001. We consider an urban service area of approximately 81 km<sup>2</sup>. Vehicles are assumed to travel straight between stops at an average speed of 20 km/hr over the region. The simulation was performed over three time intervals in a representative labor day during the morning peak hour, i.e.  $T_p = (7:00-7:59, 8:00-8:59, 9:00-9:59)$ , and the demand distribution was assumed to follow various patterns over the studied period as discussed next.

The objective of the experiments was to test the performance of the predictive algorithm under different conditions and modeling assumptions. One major factor in the definition of the expected occurrence probability of future service requests is the spatial (and temporal) disaggregation of the total area (and time period)

Following the methodology, we generate historical data assuming that 90% of the inter vehicle trips occur in 6 pair of sectors ( $H=6$ ), and therefore, we consider this subset of origin destination pairs for the cost function computation. Next, the different arrival rates per geographic pair-interval were obtained, from which the corresponding occurrence probabilities were generated. For simplicity, we assumed that the probabilities do not change within each time interval, but they do from one interval to the next. For the sake of clarity and simplicity, intrazonal trips were not considered in our numerical experiments.

The first example we tested 9 homogeneous zones, where historical data show that trips occur in the 6 most relevant interzonal origin destination pairs, as shown in Figure 5. The probability associated to each trip pattern is in the table underneath each figure. For example, regarding the schedule between 7:00 and 7:59 A.M., the chance that a trip from zone 1 to zone 5 occurs is 0.2 among all trips occurring everywhere else within that time interval. The probability of other trips between zones not considered among the six most important pairs, are assumed to be negligible.

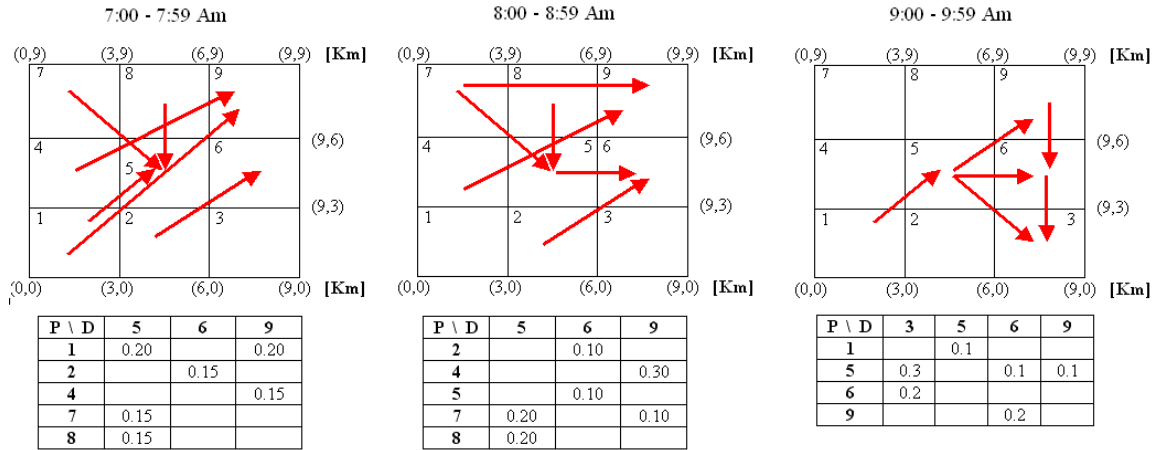


Figure 5.- Origin-destination trip patterns (9 zones division).  
P \ D: Pickup and delivery zones

A second experiment that we conducted, was to aggregate the spatial area into four zones instead of nine as in the previous example. Figure 6 shows a similar scheme for this case as the one above.

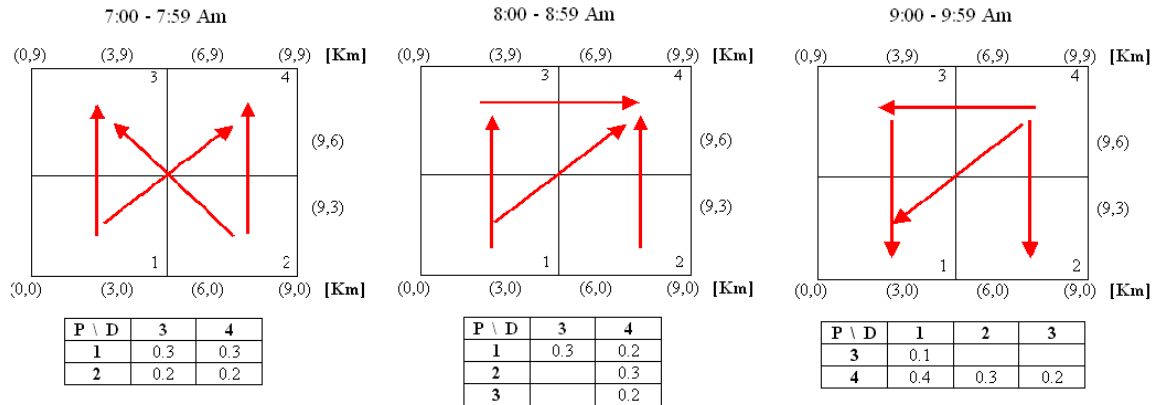


Figure 6.- Origin-destination trip patterns (4 zones division) .  
P \ D: Pickup and delivery zones

In terms of demand, 90 calls were generated over the whole simulation period of three hours according to a spatial and temporal distribution with the same behavior as the historical pattern considered. For the temporal pattern, we assume a negative exponential distribution for time intervals between calls, with rates of 0.5 [call/minute], 1 [call/minute] and 0.5 [call/minute] for the first, second and third hour of simulation respectively. In terms of spatial distribution, pick-up and delivery points were generated randomly within each corresponding zone in order to replicate the trip pattern and probabilities shown in Figures 5 and 6, depending on the experiment. Arbitrarily,

vehicles were initially located at the mass center of zones. This assumption is not affecting the final statistics of the tests, since a reasonable warm up period was considered before start looking at the system in detail to avoid boundary distortions. 30 replications of each experiment were conducted to obtain global statistics as shown in section 4.2 next. With regard to the cost function,  $\alpha = 1$  was used, which means that travel time is as important as waiting time into the cost function expression.

The one step and two step ahead algorithms are then evaluated and compared at both cases. Explicit enumeration algorithms (no swapping option) were implemented in Matlab version 6.5.1 release 13 in a Pentium IV processor.

As introduced in the previous section, one relevant fine tuning parameter is the predicted in between call request time  $\tau$ , which is relevant when evaluating the performance function of the two step ahead algorithm. We find the optimal value of such a parameter by conducting a sensitivity analysis around the observed interarrival times from the historical data report. Figure 7 show the mean value of the objective function for different  $\tau$  values. Ten replications were used in order to obtain the optimum value  $\tau = 5$  for 9 zones, and  $\tau = 2$  for 4 zones partition respectively.

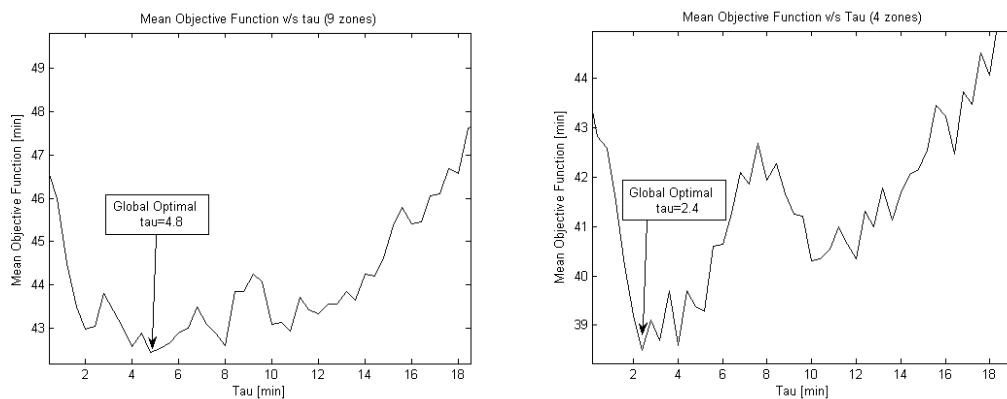


Figure 7.- Sensitivity analysis for  $\tau$  (9 and 4 zones partition).

## 4.2 Analysis of results

In Tables 2 and 3 the mean and standard deviation values of total travel and waiting times for the 30 replications are reported. These values capture system performance in terms of user level of service, when running both algorithms (one step ahead versus two step ahead) for both experiments, 9 zones (6 probabilities), and 4 zones (4 probabilities) respectively. Additionally, in Table 4, the average total time spent by a vehicle in the system is also reported as a proxy of the average operational cost. A warm up period of 20 minutes at both sides (the start and end of the simulation period) was considered to measure system performance under steady state conditions.

Table 2. One-step ahead and two-step ahead performance comparison. 9 zones, 6 probabilities.

	waiting time (min)		travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
<b>One step ahead</b>	20.6479	2.27	25.4290	1.02	46.0769	2.98
<b>Two step ahead</b>	18.3375	1.75	24.9952	1.16	43.3327	2.61
<b>Savings</b>	2.3104		0.0171		0.0593	
<b>Improv. (%)</b>	<b>11.12%</b>		<b>1.71%</b>		<b>5.93%</b>	

Table 3. One-step ahead and two-step ahead performance comparison. 4 zones, 4 probabilities.

	waiting time (min)		travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
<b>One step ahead</b>	20.2571	2.32	25.3887	0.98	45.6458	2.96
<b>Two step ahead</b>	17.1415	5.03	23.3106	6.54	40.4520	11.47
<b>Savings</b>	3.1156		2.0781		5.1937	
<b>Improv. (%)</b>	<b>15.38%</b>		<b>8.19%</b>		<b>11.38%</b>	

Table 4. One-step ahead and two-step ahead operational cost per vehicle comparison.

Vehicle travel time (min/veh)	4 zones, 4 probabilities		9 zones, 6 probabilities	
	Mean	Std	Mean	Std
<b>One step ahead</b>	141.5668	13.5577	152.7011	11.2001
<b>Two step ahead</b>	141.4056	9.6822	153.4361	6.7921

Tables 2 and 3 show the performance of the two algorithms over the whole three hours period (excluding the warm up interval). From the results shown in the tables above, we appreciate that the predictive (two step ahead) algorithm systematically performs better than the one step ahead algorithm, supporting our initial guess regarding the necessity of adding predictive power into the real time routing algorithms applied to this kind of systems.

The most important savings from the predictive approach comes from the waiting time component (15.38% in the best case). From this result, we could infer that most prediction benefits are due to avoiding extra waiting at future pickup points on scheduled vehicle sequences, which is quite manageable by the dispatcher once he can decided based upon future potential requests.

From the example, improvements in travel time from predictive algorithms are lower than those obtained for waiting time (8.19% in the best case), and also there is no observable improvement in operational cost at all (it remains practically constant). We guess that the magnitude of the travel time benefits as well as savings in operational cost could be improved by considering further adjustments of the cost function formulation. Actually, the current cost function version does not take into account the real weight of the operational cost compared with customer level of service in terms of waiting and travel time as part of the specification (see section 5 for discussion of this topic as part of planned further research).

Moreover, it seems that the most aggregated zoning scenario (Table 3) results in a more efficient routing scheme from the two step ahead algorithm than the solution obtained from one step ahead case, when compared against the 9 zone scheme (Table 2). One of the reasons to explain this difference is that the more aggregated the modeling area is, the more accurate the prediction of probabilities results. Thus, it is easier to find vehicle routes following well known trip patterns (as those shown in Figures 5 and 6), resulting in better prediction of potential rerouting along those paths.

For a more detailed representation of how the system dynamic is performing, Figure 8 shows the effective waiting time, travel time and total time spent by each passenger during his(her) trip, for one specific replication (taken among the 30 repetitions summarized above), of the more aggregated experiment (4 zones aggregation).

In Figure 8, performance patterns (in terms of waiting and travel time) are shown, passenger by passenger. This is a reasonably good replication in terms of predictive algorithm performance (26.45% and 6.64% savings in waiting and travel times respectively). By looking at Figure 8 a), it is clear that, except for one or two cases, most passengers considerably diminish their effective waiting time. Figure 8 b) on the other hand, shows a good performance of the one step ahead algorithm in term of travel time, resulting in low improvements from the two step method for most passengers. The two step algorithm seems to stabilize the system, mainly in terms of waiting time values very close to the mean for almost all customers.

Moreover, note that when the intensity of calls increase (approximately when customer 30 is served), the two step ahead algorithm (mainly on waiting time, Figure 8a) consistently starts performing better than the one step method for every customer being served. The same phenomenon is observed for travel time savings, but less significant (Figure 8b). One conclusion we could guess from this tendency is that future information becomes more important in taking proper real-time routing decisions as the problem size increases (in terms of call intensity and probably in terms of fleet size as well). Thus, the next step to follow is the development of much more powerful algorithms in order to be able to run bigger problems in short computation time (in fact, we are currently testing Genetic algorithms for solving our model as explained in the next section).



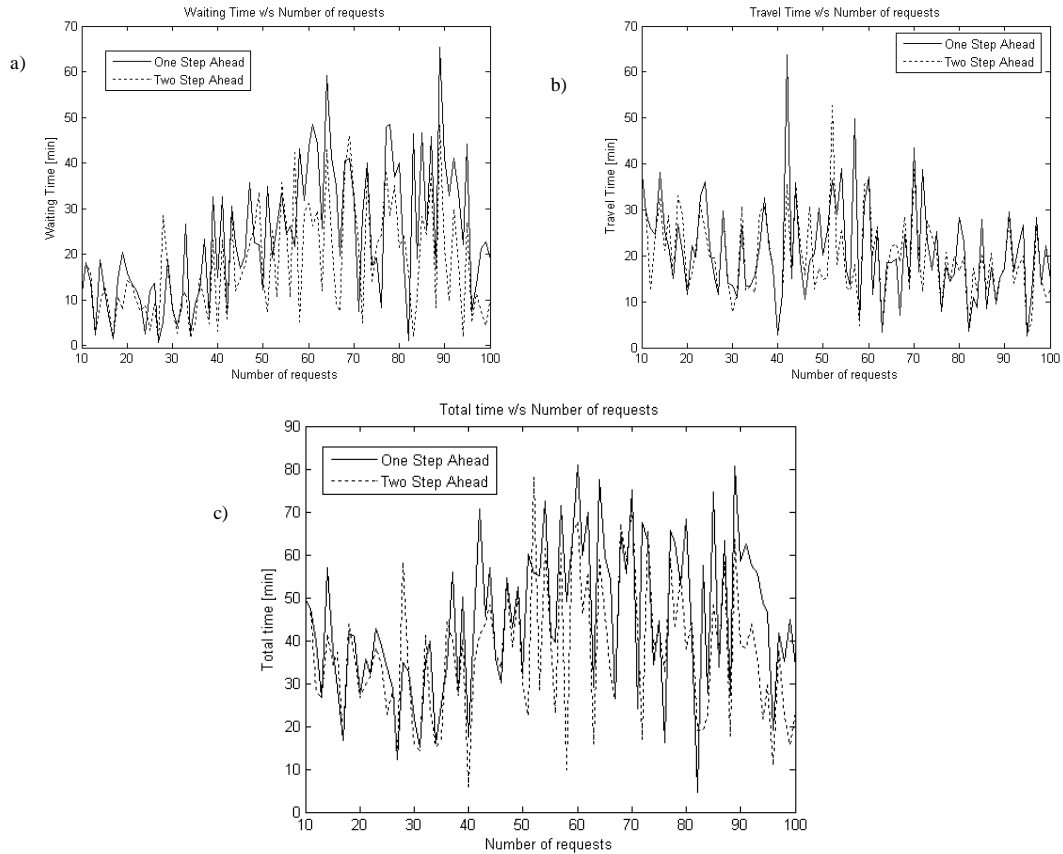


Figure 8.- Waiting time, travel time and total time spent by each passenger for one selected replication (4 zones aggregation)

## 5. SUMMARY, CONCLUSIONS AND FURTHER RESEARCH

This paper presents an analytical way of modeling the impact of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems based on a hybrid adaptive predictive control scheme to optimize the vehicle dispatch and routing decisions by including future information with regard to unknown demand. The major contribution of this research relies on the development of an adaptive dynamic state space model, including two well used descriptive variables (vehicle load and departure time) and a cost function for dispatching oriented to improve the client level of service provided by the system. By using prediction (two step ahead method) within the optimization scheme, a considerable improvement with respect to myopic rules (one step ahead method) was empirically found, showing the benefit of start introducing such schemes in real-time routing and scheduling of flexible transit systems.

This research developed an innovative approach for optimizing real time dispatch decisions of a flexible vehicle routing system under stochastic demand generated dynamically over time, by internalizing future information into both the model and

objective function. In short, this formulation allows us to find the optimal system behavior from current dispatch decisions considering uncertain future demand. The methodology is based upon real-time update of probabilities, in order to predict the real cost of future insertions into current vehicle routes.

Moreover, we recognize that the problem considered here has a hybrid nature. Thus, it has to be formulated through discrete (sequences and load) as well as continuous (departure time) variables that need to be adaptive along the system operation time and must be predictive for improving the system behavior by considering potential rerouting in the near future. For this reason, the problem was solved under a hybrid adaptive predictive control scheme. We could have predicted towards a longer horizon, however since this is a first experience, a two steps ahead prediction seems to be enough to prove the benefits of the approach so far. In addition, explicit enumeration must be replaced by a more efficient solution approach in further applications in order to incorporate a longer prediction horizon into the decisions.

Experimental results show a considerable reduction of customer waiting times along with a positive (although small) improvement in travel time savings. Benefits of the predictive approach considerably raise when the urban area is compacted into four zones instead of the original partitioning corresponding to nine zones. This benefit of the aggregation most probably came from the decrease in uncertainty from less probabilities being involved in the computations.

In further applications, several elements of the approach have to be improved. First, we expect to count on a stricter analytical expression for the state space model, particularly the  $A$  and  $B$  matrices as dependent of the vehicle sequence. In more sophisticated applications, additional components could enrich the state space models as well as the cost function formulation. We have though in adding the network speed in order to capture real traffic effects, such as recurrent and non recurrent congestion.

Second, and connected with the previous issue, a more complete rigorous expression for the dispatch decision cost function must be used. That should explore the inclusion of time windows (hard and soft), and a better consideration of operational costs. A sensitivity analysis with regard to both parameters  $\alpha$  and  $\tau$  is planned to be also investigated, when counting with more sophisticated algorithms to solve the two step ahead problem. We claim that it is possible to improve the estimation of tuning variables, such as number of probable calls, future step time prediction ( $\tau$ ) which is unknown), prediction horizon ( $N$ ), service policy, search over different feasible solutions structures, etc.

Note that this potential generalization of the methodology does not invalidate the results obtained here, since the ultimate goal of this first step on our research is to propose a

predictive control based scheme for modeling the real time routed transit and test if our premises are fulfilled based on the results of a first set of experiments

Third, the solution algorithm under this stricter scenario has to be more flexible than explicit enumeration. Genetic Algorithms (GA) and Dynamic Programming (DP) are now being investigated by the authors in order to solve more complex instances. In fact, so far we obtained promissory results from the use of GA to deal with the optimization. Since GA is much more efficient than explicit enumeration, we are now able to predict to more than two steps ahead. For the same reason, we have been able to allow partial swapping of the original vehicle sequences to search into a bigger feasible set, and get solutions closer to the optimum.

And fourth, the way in which the system information is internalized through probabilities can also be improved through a more realistic scheme. That needs a better understanding of the historical data (Fuzzy clustering). We also propose to combine historical data (offline) with online information in a more elaborated model able to capture imminent events that could affect the system performance.

### **Acknowledgments**

This research was partially financed by Fondecyt, Chile, grants 1040698 and 1030700, and the Millennium Nucleus "Complex Engineering Systems". The authors also would like to acknowledge the helpful comments of Riju Lavanya as well as those from anonymous referees.

### **References**

- W. Bell, W., L.M. Dalberto, M.L. Fisher, A.J Greenfield, R. Jaikumar, P. Media, R.G. Mack and P.J. Prutzman, "Improving the Distribution of Industrial Gases with an Online Computerized Routing and Scheduling Optimizer", *Interfaces* 13, 4 – 23, (1983).
- A. Bemporad, F. Borrelli, M. Morari, "Model predictive control based on linear programming. The explicit solution". *IEEE Transactions Automatic Control*, 47 (12), 1974-1985, (2002).
- A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics and constraints". *Automatica*, 35, 407-427, (1999).
- D. Bertsekas and J. N. Tsitsiklis, "Neuro-Dynamic Programming", *Athena Scientific*, (1996).
- D. Bertsimas and G. Van Ryzin, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane", *Operations Research* 39, 601-615 (1991).

- D. Bertsimas and G. Van Ryzin, “Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane with Multiple Capacitated Vehicles”, *Operations Research* 41, 60-76 (1993).
- D. Bertsimas and G. Van Ryzin, “Stochastic and Dynamic Vehicle Routing with General Demand and Interarrival Time Distributions”, *Applied Probability* 25, 947-978 (1993).
- D. Bertsimas, “A vehicle routing problem with stochastic demand”, *Operations Research*, 40, 574-585, (1992).
- D. Bertsimas and L.H. Howell. “Further Results on the Probabilistic Traveling Salesman Problem”. *European Journal of Operational Research* 65, 68-95 (1993).
- D. Bertsimas, P. Jaillet and A. Odoni, “A priori optimization”, *Operations Research*, 38 (6), 1019-1033, (1990).
- D. Bertsimas and D. Simchi-Levi, “A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty”, *Operations Research* 44, 286-304 (1996).
- N. Bianchessi and G. Righini, “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery”, *Computers & Operations Research* (Article in Press) (2006).
- C.E. Cortés and R. Jayakrishnan, “Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle Pickup and Delivery problems”, at The fifth Triennial symposium on transportation analysis, TRISTAN V. 13-18 June. Le Gosier, Guadalupe (2004).
- C.E. Cortés, High-Coverage Point-to-Point Transit (*HCPPT*): A new design concept and simulation-evaluation of operational schemes for future technological deployment, *Ph.D. Dissertation*, University of California at Irvine, U.S.A. (2003).
- J. Desrosiers, F. Soumis, and Y. Dumas, “A Dynamic Programming Solution of a Large-Scale Single-Vehicle Dial-a-Ride with Time Windows”, *American Journal of Mathematical and Management Sciences* 6, 301-325 (1986).
- R. Dial, “Autonomous Dial a Ride Transit – Introductory Overview”, *Transportation Research - Part C* 3, 261-275 (1995).
- J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, “Metaheuristics for Hard Optimization Methods and Case Studies”, Springer-Verlag, 2006.
- M.L. Fisher and R. Jaikumar, “A Generalized Assignment Heuristic for Solving the VRP”, *Networks* 11, 109 (1982).

- C. Floudos, *Non linear and mixed integer optimization*, Oxford University Press (1995).
- M. Gendreau, F. Guertin, J. Potvin, and E. Taillard, "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching", *Transportation Science* 33, 381-390 (1999).
- M. Gendreau, G. Laporte, and R. Séguin, "An Exact algorithm for the Vehicle Routing Problem with stochastic Customers and Demands", *Transportation Science* 29, 143-155 (1995).
- M. Gendreau, G. Laporte, and R. Séguin. "Stochastic Vehicle Routing", *European Journal of Operational Research*, 88, 3-12 (1996).
- A. George and W. Powell, "Adaptive Stepsizes for Recursive Estimation with Applications in Approximate Dynamic Programming", (2005) <http://www.castlelab.princeton.edu/>.
- G. Godfrey and W.B. Powell, "An Adaptive Dynamic Programming Algorithm for Stochastic Resource Allocation Problems I: Single Period Travel Times", *Transportation Science* 36, 21-39 (2002).
- A. Haghani, S. Jung, "A dynamic vehicle routing problem with time-dependent travel times", *Computers & Operations Research* 32, 2959-2986 (2005).
- S. Ichoua, M. Gendreau and J.Y. Potvin, "Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching", accepted for publication in *Transportation Science* (2005).
- P. Jaillet, "Probabilistic Traveling Salesman Problems", Ph.D. Dissertation, Massachusetts Institute of Technology, (1985).
- P. Jaillet, "A Priori Solution of the Traveler Salesman Problem in which a Random Subset of the Customers are visited", *Operations Research* 36, 929-936 (1988).
- J. Jaw, A. Odoni, H. Psaraftis and N. Wilson, "A heuristic algorithm for the multivehicle many-to-many advance-request dial-a-ride problem," Working paper MITUMTA-82-3, M.I.T., Cambridge, M (1986).
- W. Jih, J. Yun-jen, "Dynamic Vehicle routing using hybrid genetic algorithms", *Proceeding of the IEEE International Conference on Robotics & Automation, Detroit, Michigan, May 1999*, 453-458 (1999).

A.J. Kleywegt and J.D.Papastavrou, "[The Dynamic and Stochastic Knapsack Problem](#)", *Operations Research*, 46, 17-35, (1998).

A.J. Kleywegt and J.D.Papastavrou, "The dynamic and stochastic knapsack problem with random sized items", *Operations Research* 49(1), 26-41 (2001).

G. Laporte, F.V. Louveaux and H. Mercure, "A Priori Optimization of the Probabilistic Traveling Salesman Problem", *Operations Research* 42, 543-549 (1994).

A. Larsen, "The Dynamic Vehicle Routing Problem", Ph.D. Thesis, Technical University of Denmark, (2000).

A. Le Bouthillier and Crainic, "A cooperative parallel meta-heuristic for the vehicle routing problem with time windows", *Computers & Operations Research*, Vol. 32, pp. 1685-1708 (2005).

F. Li, B. Golden, E. Wasil, "Very large-scale vehicle routing: new test problems, algorithms, and results", *Computers & Operations Research* 32, 1165-1179 (2005).

O. Madsen, H. Raven and J. Rygaard, "A heuristics algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives", *Annals of Operations Research* 60, pp. 193-208. (1995).

C. Malandraki and M. S. Daskin, "Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms", *Transportation Science* 26, 185-200 (1992).

A.S. Minkoff, "A Markov decision model and decomposition heuristic for dynamic vehicle dispatching", *Operations Research* 41(1), 77-90 (1993).

R. Montemanni, L. M. Gambardella, A. E. Rizzoli and A. Donati, "Ant colony system for a dynamic vehicle routing problem", *Journal of Combinatorial Optimization*, Issue: Volume 10, Number 4, December, Pages: 327 – 343 (2005).

M. Osman, M. Abo-Sinna and A. Mousa, "An effective genetic algorithm approach to multiobjective routing problems (MORPs)", *Applied Mathematics and Computation* 163, 769-781 (2005).

J.D. Papastavrou, S. Rajagopalan, and AJ Kleywegt, "The Dynamic and Stochastic Knapsack Problem with Deadlines", *Management Science* 42, 1706-1718 (1996).

- W.B. Powell, P. Jaillet and A. Odoni, "Stochastic and dynamic networks and routing". in: *Network routing. A handbook in operations research and management science Vol. 8*. Elsevier (eds), 141-296, The Netherlands, (1995).
- W. B. Powell, "A comparative review of alternative algorithms for the dynamic vehicle allocation problem", B.L. Golden and A.A. Assad editors, *Vehicle routing methods and studies*, (1988).
- H. Psaraftis, "Dynamic vehicle routing problems", B.L. Golden and A.A. Assad editors, *Vehicle routing methods and studies*, 223-248 (1988).
- H. Psaraftis, "A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem", *Transportation Science* 14(2), pp. 130-154 (1980).
- S. Roy, J.M. Rousseau, G. Lapalme and J. Ferland, "Routing and Scheduling of Transportation Services for the Disabled" *Summary Report. Technical Report CRT-473A*, Centre de recherche sur les transports, Université of Montréal (1984).
- M. Savelsbergh and M. Sol, "The general Pickup and Delivery problem", *Transportation Science*, Vol. 29, No. 1, pp. 17-29, (1995).
- N. Secomandi, "Exact and Heuristics Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands", Ph.D. Dissertation, College of Business Administration, University of Houston (1998).
- N. Secomandi, "Comparing Neuro-Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands", *Computers and Operations Research* 27, 1201-1225 (2000).
- D. Skrllec, M. Filipec and S. Krajcar, "A heuristic modification of genetic algorithm used for solving the single depot capacitated vehicle routing problem", *Proceedings of Intelligent Information Systems, IIS '97*, 184-188 (1997).
- M. Spivey and W.B. Powell, "The Dynamic Assignment Problem," *Transportation Science* 38(4), 399-419 (2004).
- M.R. Swihart and J.D. Papastavrou, "A Stochastic and Dynamic Model for the Single-Vehicle Pick-Up and Delivery Problem", *European Journal of Operational Research* 114, 447-464, (1999).
- C. Tarantilis, G. Ioannou and G. Prastacos, "Advanced vehicle routing algorithms for complex operations management problems", *Journal of Food Engineering* 70, 455-471 (2005).

C. Tarantilis, "Solving the vehicle routing problem with adaptive memory programming methodology", *Computers & Operations Research* 32, 2309-2327 (2005).

Thomas, B.W. and C.C. White III, "Anticipatory Route Selection", *Transportation Science* 38(4), 473-487 (2004)

A. Tighe, F. Smith and G. Lyons, "Priority based solver for a Real-Time Dynamic Vehicle Routing", *IEEE International Conference on Systems, Man and Cybernetics*, pgs: 6237-6242 (2004).

P. Toht, D. Vigo, "The granular Tabu search and its application to the vehicle-routing problem", *INFORMS Journal on Computing*, Volume: 15. Fall, Number: 4. Pgs: 0333-0346, (2003).

H. Topaloglu and W. Powell, "A Distributed Decision-Making Structure for Dynamic Resource Allocation Using Non Linear Functional Approximations", *Operations Research* 53(2), 281-297 (2005).

C.D.J. Waters, "Vehicle Routing Problems with Uncertainty and Omitted Customers", *Journal of the Operational Research Society* 40, 1099-1108 (1989).

N. Wilson and N. Colvin, "Computer control of Rochester dial-a-ride system", *Report R77-31*, Department of Civil Engineering, M.I.T., Cambridge, MA. (1977)

N. Wilson and H. Weissberg, "Advanced dial-a-ride algorithms research project: Final report", *Report R76-20*, Department of Civil Engineering, M.I.T., Cambridge, MA. (1976).

W.H. Yang, K. Mathur and R.H. Ballou, "Stochastic Vehicle Routing Problem with Restocking", *Transportation Science* 34(1), 99-112 (2000).

K.Q. Zhu, "A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows", *15th IEEE International Conference on Tools with Artificial Intelligence, November 3-5, 2003*, 176-183 (2003).



## **ANEXO 6. CONTROL PREDICTIVO HIBRIDO PARA UN SISTEMA PERSONALIZADO DE TRANSPORTE PÚBLICO PUERTA A PUERTA PROGRAMADO EN TIEMPO REAL**

**Cristián E. Cortés**

Departamento de Ingeniería Civil  
Universidad de Chile, Santiago, Chile.  
Fono: (56-2) 678 4380, Fax: (56-2) 689 4206  
**e-mail : [ccortes@ing.uchile.cl](mailto:ccortes@ing.uchile.cl)**

**Doris Sáez H.**

Departamento de Ingeniería Eléctrica  
Universidad de Chile, Santiago, Chile  
Fono: (56-2) 678 4207, Fax: (56-2) 695 3881  
**e-mail: [dsaez@ing.uchile.cl](mailto:dsaez@ing.uchile.cl)**

**Felipe Murcia M.**

Departamento de Ingeniería Industrial  
Universidad de Chile, Santiago, Chile.  
Fono: (56-2) 678 4026, Fax: (56-2) 689 7895  
**e-mail: [fmurcia@ing.uchile.cl](mailto:fmurcia@ing.uchile.cl)**

**Alfredo Núñez V.**

Departamento de Ingeniería Eléctrica  
Universidad de Chile, Santiago, Chile  
Fono: (56-2) 678 4207, Fax: (56-2) 695 3881  
**e-mail: [alfnunez@ing.uchile.cl](mailto:alfnunez@ing.uchile.cl)**

### **R E S U M E N**

En este artículo se sintetiza el diseño de un modelo de programación de itinerarios de vehículos pertenecientes a un sistema de transporte de pasajeros con demanda dinámica, puerta a puerta y operando sobre un área urbana. El objetivo central de este trabajo es la formulación de un modelo de programación de itinerarios de vehículos que se adapte a las condiciones del sistema y que sea capaz de predecir las condiciones futuras de acuerdo con datos históricos y respondiendo a la demanda que se genera en tiempo real. Se utilizó una estrategia de control del tipo control predictivo adaptivo híbrido, que se caracteriza por trabajar con variables tanto continuas como enteras. En primer lugar, se formula el problema en variables de estado, las cuales son carga esperada y tiempo esperado de llegada a cada parada. De esta formulación, se desarrolla modelos de carga y tiempo de llegada con capacidad de predicción en el instante posterior (un paso) y en dos instantes futuros (dos pasos), de acuerdo con funciones de costo *ad hoc*, considerando tanto usuarios como operadores. Se muestra con un ejemplo, los beneficios en ahorro en tiempos de espera y viaje globales por efecto de usar una estrategia predictiva a más de un paso.

## 1. INTRODUCCIÓN

Uno de los problemas más estudiados en logística es el problema de recoger y dejar pasajeros, caracterizado por satisfacer un conjunto de requerimientos de servicio a partir de una flota de vehículos de tamaño fijo, y donde cada vehículo inicialmente se localiza en terminales o bodegas predeterminadas (Desrosiers *et al.*, 1995; Savelsberg y Sol, 1995). El problema puede generalizarse al caso dinámico, en el cual los requerimientos de servicio no son conocidos a priori, y por lo tanto, las decisiones de ruteo en el futuro podrían afectar a los usuarios cuya asignación se hace en tiempo presente (Psaraftis, 1988).

En este trabajo, se aborda el problema del ruteo de vehículos de transporte de pasajeros con servicio puerta a puerta, el cual presenta complejidad para la toma de decisiones de despacho cuando la demanda aparece en tiempo real. Estas decisiones tenderían a afectar el rendimiento del sistema percibido por los usuarios, lo que debiera reflejarse en baja demanda y deficiente productividad. Uno de los tópicos más relevantes para mejorar la eficiencia de estos sistemas es una definición apropiada de la función de costos de decisión, que debiera incluir tiempo de viaje y espera de los usuarios, así como costos de operación de los vehículos. Sin embargo, cuando el problema es dinámico, un efecto adicional debiera ser considerado dentro de la especificación analítica de tales funciones de costo, cual es el costo adicional asociado con potenciales reruteos futuros provocados por la demanda incierta, en las decisiones tomadas en tiempo real. Este problema fue tratado heurísticamente por Cortés y Jayakrishnan (2004) quienes formulan conceptualmente el problema dentro del marco de la teoría de control predictivo.

En la literatura especializada, es común que el tiempo de viaje entre paradas se asuma constante para efectos de computar el costo detrás de las reglas de programación dinámica y ruteo de vehículos (por ejemplo, Jaw *et al.*, 1986; Psaraftis, 1986). En esta investigación, proponemos un esquema para estimar el costo real de una inserción basado en información futura, con el objeto de tomar mejores decisiones dinámicas de inserción, en base a las condiciones de los vehículos y del sistema en todo instante. Notar que este análisis se sustenta en satisfacer las necesidades de servicio de clientes desconocidos que entrarán al sistema en el futuro, provocando el menor impacto posible (en términos de costo) sobre el operador y los usuarios que ya están en el sistema al momento de tomar la decisión. Sistemas de control de rutas e itinerarios que permitan tomar decisiones de despacho en tiempo real se hacen cada vez más atractivos principalmente debido al desarrollo de la tecnología en temas tan diversos como tecnologías de navegación, sistemas de posicionamiento satelital y sistemas de identificación automática de vehículos.

Aquí, el problema se formula bajo un enfoque de control híbrido, ya que se introducen variables tanto continuas como discretas, y se resuelve para un horizonte de predicción de dos pasos, el cual se compara con los resultados a un paso. Para esto, el sistema es definido en términos de variables de estado, cuales son la carga y el tiempo de llegada

esperado en cada parada programada de los vehículos, se formulan las funciones de costo a uno y dos pasos y se resuelve usando enumeración explícita, considerando las restricciones físicas típicas de precedencia y capacidad de los vehículos. En la siguiente sección se hace referencia a la literatura en ruteo dinámico de vehículos y a los conceptos y formulaciones básicas de control predictivo. Luego se formula nuestro problema (en términos de variables de estado y funciones de costo), para terminar con una prueba empírica que muestra la necesidad de incorporar enfoques predictivos de este tipo en la toma de decisiones de sistemas dinámicos.

## **2. REVISIÓN DE LA LITERATURA: RUTEO DE VEHÍCULOS Y CONTROL PREDICTIVO**

En esta sección, nos parece interesante revisar la literatura en temas de ruteo dinámico de vehículos, y por otra parte identificar los elementos de la teoría de control predictivo que fuesen aplicables al problema de ruteo de vehículos en tiempo real bajo nuestro enfoque. Psaraftis (1988), definió el problema de ruteo dinámico de vehículos de pasajeros como la programación de un conjunto de rutas para todos los vehículos, las cuales cambian dinámicamente sobre el tiempo según los requerimientos de demanda. Se destacan las complicaciones adicionales del programa dinámico sobre el estático, enfatizando detalles tales como lo insensato que parece definir ventanas de tiempo duras en problemas dinámicos. Minkoff (1993) propuso un modelo de decisión markoviano y una heurística de descomposición para el despacho dinámico de vehículos. Adicionalmente, Bertsimas y Simchi-Levi (1996), propusieron una nueva generación de investigación en ruteo de vehículos basado en algoritmos de optimización robusta, con estocacidad.

Una muy buena fuente de información para efectos de nuestra investigación, resulta ser la literatura respecto del problema de ruteo estocástico de vehículos. Se destaca el trabajo de Bertsimas *et al.* (1990) enfocado a situaciones donde el tamaño de la demanda es desconocida. Este problema es resuelto creando un conjunto de rutas que minimizan el costo esperado antes de que la demanda ocurra, asignando dicha demanda automáticamente a una ruta y a una posición sobre la ruta, de acuerdo con una programación *a-priori* (por soluciones *a-priori* se entiende que el planificador determina una o más rutas basadas en información probabilística de requerimientos futuros de servicio, demanda de clientes, tiempos de viaje, etc.). Los costos esperados incluyen el costo de viaje esperado entre los clientes sobre las rutas y los costos de penalización en los que se incurre si se necesitan rutas adicionales, cuando la demanda excede la capacidad. Adicionalmente, Bertsekas y Tsitsiklis (1996) proponen una un enfoque de programación neuro dinámica que también permite resolver el problema de despacho dinámico de vehículos. Esta metodología permite a los sistemas “aprender a tomar buenas decisiones por observación de su propio comportamiento (simulación) y usar mecanismos incorporados para mejorar sus acciones a través de mecanismos de refuerzo (esquemas iterativos usados para mejorar la calidad de la función de costo).

Una investigación más reciente realizada por Malucelli (1999) describe un nuevo sistema de transporte flexible. Su sistema considera líneas de ruta fijas convencionales combinadas con líneas basadas en itinerarios flexibles y programados. Zhu y Ong (2000), proponen un método reactivo para resolver el problema de ruteo dinámico de vehículos. Ichoua (2000), propone una metodología para caracterizar los aspectos de diversión o divergencia de vehículos en tiempo real usando avances tecnológicos recientes en sistemas de comunicación que permiten la utilización de información en tiempo real para el ruteo dinámico de vehículos y programación.

Respecto del control predictivo con variables enteras (control híbrido), Bemporad y Morari (1999) presentan un esquema de control predictivo para sistemas híbridos incluyendo restricciones operacionales que también fueron resueltas usando programación cuadrática entera mixta. Bemporad *et al* (2002) desarrollan un trabajo sobre control predictivo basado en modelos resuelto por una técnica de programación entera denominada enumeración explícita. Ellos proponen un algoritmo tipo *Branch and Bound* con gran eficiencia al aplicarse al problema de control predictivo híbrido. Finalmente, Borrelli *et al.* (2003) presentan un algoritmo eficiente para resolver el problema de control óptimo en tiempo finito basado en sistemas híbridos lineales de tiempo discreto con un criterio de desempeño cuadrático. Todos los trabajos en control predictivo referenciados arriba muestran mayoritariamente aplicaciones vía simulación para problemas de control típicos. Una contribución del presente artículo es formular el problema de transporte real descrito como un problema de control predictivo híbrido y resolverlo.

### 3. DESCRIPCIÓN DEL PROBLEMA

Suponga que se tiene un conjunto de vehículos  $VF$  de tamaño de flota  $F$ . En algún instante de tiempo se asume que cada vehículo  $j \in VF$  ha sido asignado a una serie de tareas que incluyen recogidas y entregas de pasajeros, las que pueden ser representadas por una secuencia  $S_j = \{1, 2, \dots, i, \dots, w_j\}$ , en la cual el  $i$ -ésimo elemento de la secuencia representa una parada específica a lo largo de la ruta del vehículo  $j$ . La posición del vehículo  $j$  en el instante  $k$  se denota por  $v_j$  y este corresponde al primer elemento de la secuencia  $S_j$ .

Los requerimientos de servicio entran al sistema dinámicamente y las decisiones tienen que ser tomadas en tiempo real basadas en algún tipo de algoritmo de inserción perteneciente al controlador. La decisión final está dada por la inserción de nuevos requerimientos dentro de la mejor secuencia  $S_j$ , (aquella con el mínimo costo incremental de inserción) sujeto a las restricciones de factibilidad. En este trabajo se desarrolla un esquema de control predictivo adaptivo para modelar tal problema. La

formulación se basa en la siguiente premisa: los tiempos esperados de viaje y espera en que todos los usuarios ya programados podrían incurrir durante su viaje podrían verse fuertemente afectados por futuras reasignaciones de los vehículos al programar llamadas futuras mientras van entrando dinámicamente al sistema. La idea es introducir dentro de las funciones de costo de asignación vehículo-pasajero una componente de predicción estocástica de forma de estimar de mejor manera y más realista los tiempos de viaje y espera que realmente experimentarían los individuos a lo largo de su viaje, de donde se debieran obtener mejores reglas de despacho en el sentido de minimizar el costo total del sistema.

El esquema propuesto es predictivo en el sentido de tomar decisiones de menor impacto futuro, y es adaptivo en la forma de generar reglas que se adapten dependiendo de cómo varíen las condiciones del sistema en el tiempo. Dado que las decisiones lógicas de asignación involucran acciones discretas, conceptualmente este problema puede ser considerado como una forma de un sistema de control híbrido adaptivo predictivo cuasi óptimo (HAPC en la literatura especializada). Para describir la formulación, asumimos que se trabaja sobre un área de influencia  $A$ , dentro de una red de servicio de longitud  $D$ , -en unidades de distancia [UD]-. También contamos con una flota de vehículos de tamaño  $F$  moviéndose dentro del área de acuerdo con reglas de ruteo y programación determinadas por el algoritmo despachador. La demanda por servicio es desconocida y se genera en tiempo real (cuya media es  $\mu$  en llamadas por unidad de tiempo [llamada/UT] ). El controlador predictivo toma las decisiones de ruteo en tiempo real basado en la información del sistema (proceso), y de los valores esperados tanto para el tiempo de viaje como para otros atributos de la flota de vehículos (modelo). Dicho controlador predictivo toma las decisiones de asignación de pasajero a las rutas de los vehículos siempre que un nuevo requerimiento entra al sistema. En otras palabras, el mecanismo de control es activado siempre que una nueva llamada ingrese. Por lo tanto, en lugar de definir un lapso de tiempo fijo, en este esquema de control se formula el problema en términos de un lapso de tiempo variable, el cual representa el intervalo de tiempo entre los requerimientos  $k-1$  y  $k$ .

El estado del sistema en el instante  $k$  (por ejemplo, cuando un requerimiento entra al sistema) está determinado por los atributos de cada vehículo, denominados  $M_j(i-1, i)$  para el vehículo  $j$  sobre el segmento  $(i-1, i)$  de la secuencia de la ruta. Estos atributos incluyen todas las características de los vehículos cuando parten desde la  $i$ -ésima parada de su secuencia. En la formulación propuesta el vector de atributos está compuesto por el tiempo de reloj de llegada  $T_j^i(k)$  y por la carga del vehículo  $L_j^i(k)$ , ambos para cada parada  $i$  dentro de la ruta del vehículo  $j$  en el instante  $k$ . En un contexto más general, otras variables podrían ser añadidas para obtener una visión más completa del estado de los vehículos, tales como el tiempo acumulado de viaje experimentado por todos los pasajeros a bordo después de la parada  $i$  y una posible medida de disponibilidad de espacio en otros vehículos para potenciales pasajeros en la proximidad del segmento

$(i-1, i)$  (Cortés y Jayakrishnan, 2004). El controlador  $u(k)$  puede ser visualizado como las decisiones de despacho del módulo de ruteo representadas por el conjunto de secuencias asignadas a cada vehículo en el instante  $k$ . Analíticamente,

$$u(k) = \{S_j\}_k = \{S_j(k)\} \quad \forall j: 1, \dots, F \quad (1)$$

Notar que las secuencias  $S_j(k)$  se mantienen fijas durante el intervalo de tiempo completo  $(k, k+1)$ . El vector de medida de proceso  $y$ , puede ser computado midiendo los tiempos de viaje para los segmentos observados  $T_j^{(i-1, i)}(k)$ , ocurridos durante el intervalo de tiempo  $(k, k+1)$  para todos los vehículos y para todos los pares de paradas  $(i-1, i)$ . Cabe anotar que cada pasajero que se incorpora al sistema, genera dos puntos de referencia, cuyas coordenadas geográficas definen el lugar exacto de recogida y de entrega de dichos pasajeros. La decisión de despacho dinámico depende de la función de costo que se utilice (ver sección 4), la cual está expresada en términos del tiempo de viaje del segmento  $T_j^{(i-1, i)}(k)$  como su principal componente. Adicionalmente, la formulación permite la medición del tiempo de espera y el tiempo de viaje del segmento medido, como un indicador decisivo de la acción de control.

El valor esperado de los atributos de cada vehículo (carga y tiempo de llegada en cada parada), -calculados para un horizonte de predicción  $N$ -, se entregan para cada instante  $k$  usando un modelo en variables de estado. Este modelo se usa para predecir la salida del sistema futuro  $y(k+l)$  con  $l=1, \dots, N$ , basado en el pasado y en los valores actuales de las variables de estado y las acciones de control futuras  $u(k+l-1)$ . Los valores predichos  $y(k+l)$  para  $l=1 \dots N$  dependen de los valores conocidos en el instante  $k$  (entradas y salidas anteriores) y la señal de control  $u(k+l-1)$ . Analíticamente, los tiempos de viaje por trayecto se computan directamente como  $T_j^{(i-1, i)}(k) = T_j^i(k) - T_j^{i-1}(k)$ . Se entiende que los pares de paradas  $(i-1, i)$ , son adyacentes en la secuencia actual  $S_j(k)$ . Entonces, por ejemplo, cuando  $l=1$ , la variable  $x(k)$  representa a los atributos (carga y tiempo de llegada) que un vehículo tiene entre un par de puntos (paradas) en un instante de tiempo determinado,

$$x(k) = \{M_j(i-1, i)\}_k \quad \forall j: 1, \dots, F \quad (2)$$

$$y(k) = \{T_j^{(i-1, i)}(k)\} \quad \forall j: 1, \dots, F \quad (3)$$

donde

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k+1) &= g(x(k+1)) \end{aligned} \quad (4)$$

Después de evaluar una cierta función de costo (ver sección 5), el controlador decide la localización óptima de requerimientos  $k+1$ , generando el vector final de secuencias

$S^*(k)$ . En los casos donde  $l > 1$  (el horizonte de predicción es mas grande que el lapso de tiempo siguiente al momento actual), el controlador adiciona el elemento predictivo dentro de la formulación, ya que las decisiones que fueron tomadas en  $k+1$  podrán depender de los posibles eventos a ocurrir en instantes futuros. El proceso de decisión de la secuencia de cada vehiculo, requiere datos obtenidos del desempeño del periodo para un horizonte  $H_e$  medido hacia atrás desde la ocurrencia del evento  $k$ . Un elemento central para poder establecer la dinámica del problema es la formulación en variables de estado del problema específico de ruteo dinámico de vehículos descrito aquí, tal como se desarrolla en la siguiente sección. En la sección 5, se explica el mecanismo adaptivo basado en la actualización de los parámetros del modelo dados los datos observados (procedimiento de calibración).

#### 4. FORMULACIÓN EN VARIABLES DE ESTADO PARA EL PROBLEMA DE RUTEO DINÁMICO DE VEHÍCULOS

Esta investigación considera los modelos en variables de estado híbridos, específicamente los asociados a la formulación *MLD (Mixed Logical Dynamical systems)*. Por lo tanto, lo que se decide realizar es adaptar el problema a un esquema de control similar al de los modelos híbridos en variables de estado para expresar la carga y los tiempos de arribo a cada parada, explicitando la primera ecuación en expresión (4), de la forma

$$\begin{bmatrix} L(k+1) \\ T(k+1) \end{bmatrix} = \begin{bmatrix} f_L(L(k), S(k)) \\ f_T(T(k), S(k)) \end{bmatrix} \quad (5)$$

##### 4.1. Modelo en variables de estado de la carga del vehículo

El comportamiento de la carga del vehículo se obtiene usando el siguiente modelo

$$\mathbf{L}(k+1) = \mathbf{A}_L \mathbf{L}(k) + \mathbf{B}_L(\mathbf{S}(k)) \quad (6)$$

La variable manipulada  $\mathbf{S}(k)$  esta representada por expresión (7), donde la primera y segunda columna representan la recogida y la entrega en la parada  $i$  respectivamente.

$$\mathbf{S}(k) = \begin{bmatrix} p^i & d^i & t^i & label^i \\ p^{i+1} & d^{i+1} & t^{i+1} & label^{i+1} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{(w_j+3) \times 4} \quad (7)$$

donde

$$p^i = \begin{cases} 1 & \text{si } label^i \text{ es una parada tipo } pick-up \\ 0 & \text{Caso contrario} \end{cases} \quad \text{y}$$

$$d^i = \begin{cases} 1 & \text{si } label^i \text{ es una parada tipo } delivery \\ 0 & \text{Caso contrario} \end{cases}.$$

Nótese que  $p^i = 1 - d^i$ . Se asume que cada requerimiento de servicio se asocia a un solo pasajero, modelación que fácilmente se puede extender a múltiples pasajeros en cada lugar de recogida. La tercera columna representa la función externa de tiempo de viaje (asumida exógena y siempre conocida por el controlador en este trabajo), donde  $t^i$  es el tiempo de viaje de la red entre los puntos  $i-1$  y  $i$ . Finalmente, la última columna  $label^i$  indica la identificación del pasajero. El vector de carga se puede escribir como

$$\mathbf{L}(k) = [L_j^0(k) \quad L_j^1(k) \quad \dots \quad L_j^{w_j}(k)]_{(w_j+1) \times 1}^T \quad (8)$$

Donde  $L_j^i(k)$  representa la carga del vehículo  $j$ , cuando dicho vehículo abandona la parada  $i$  en el periodo de decisión  $k$ , y  $w_j$  es el número de paradas pertenecientes a una secuencia de un vehículo elegido. Las matrices del modelo en variables de estado, definidas en la ecuación (6) son las siguientes:

$$\mathbf{B}_L(\mathbf{S}(k)) = \mathbf{B}_L^2 \cdot (\mathbf{S}(k) \cdot \mathbf{B}_L^1)$$

$$\mathbf{A}_L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)} \quad ; \quad \mathbf{B}_L^1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}_{4 \times 1} \quad ; \quad \mathbf{B}_L^2 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

Las matrices auxiliares  $\mathbf{A}_L, \mathbf{B}_L^1, \mathbf{B}_L^2$  son parámetros adaptivos del modelo en variables de estado para la carga requeridos para ajustar el vector de carga a la dimensión de la secuencia.

## 4.2 Modelo en variables de estado del tiempo de llegada

Este modelo es definido por el comportamiento del tiempo de llegada del vehículo a sus paradas, obtenido usando una metodología similar, representada en la siguiente expresión

$$\mathbf{T}(k+1) = \mathbf{A}_T \cdot \mathbf{T}(k) + \mathbf{B}_T(\mathbf{S}(k)) \quad (9)$$

La variable manipulada es de nuevo la secuencia  $\mathbf{S}(k)$ , y el vector de tiempo de llegada es:



$$\mathbf{T}(k) = \begin{bmatrix} T_j^0(k) & T_j^1(k) & \cdots & T_j^{w_j}(k) \end{bmatrix}_{(w_j+1) \times 1}^T \quad (10)$$

donde  $T_j^i(k)$  representa el tiempo de llegada del vehículo  $j$  a la parada  $i$  medida en el intervalo  $k$ . Así mismo,  $w_j$  es el número de paradas en una secuencia de un vehículo dado. Las matrices del modelo en variables de estado definidas en la ecuación (9) son en este caso las siguientes:

$$\mathbf{B}_T(\mathbf{S}(k)) = \mathbf{B}_T^2(\mathbf{S}(k)\mathbf{B}_T^1)$$

$$\mathbf{A}_T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}_{(w_j+3) \times (w_j+1)}$$

$$; \mathbf{B}_T^2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}_{(w_j+3) \times (w_j+3)}$$

$$; \mathbf{B}_T^1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{(4) \times 1}$$

Similarmente al modelo de carga, la matrices  $\mathbf{A}_T, \mathbf{B}_T^1, \mathbf{B}_T^2$  son parámetros adaptivos del modelo en variables de estado para el tiempo de arribo. La dimensión de estas matrices también depende de la dimensión de la secuencia.

## 5. ALGORITMOS DE CONTROL HÍBRIDO ADAPTIVO PREDICTIVO (HAPC) PARA EL PROBLEMA DE PROGRAMACIÓN Y RUTEO DINÁMICO DE VEHÍCULOS

En esta sección se explicitan los algoritmos con los cuales el controlador toma las decisiones. La función de costos es la métrica de desempeño con la cual el controlador toma cada decisión de ruteo. El controlador debe asignar cada llamada -que aparece aleatoriamente en el tiempo y el espacio-, a la ruta del vehículo más conveniente (aquel que minimice la función de costos para todo el sistema). Uno podría plantear el problema genérico a  $n$  pasos, sin embargo en este trabajo lo plantearemos a uno y dos pasos como horizonte de evaluación, y se resolverá con enumeración explícita para comprobar empíricamente las ventajas de introducir el elemento predictivo en la asignación pasajero-vehículo.

### 5.1 Esquema de control a un paso

El esquema de control a un paso tiene una función de desempeño económico dada por la función de costos de un vehículo en particular midiendo una nueva inserción, expresada como un costo incremental; la asignación se evalúa en base a dicha función determinando la mejor secuencia que podría tener cada vehículo, dentro de todas las secuencias factibles (en términos de precedencia y capacidad de los vehículos), dado que se debe satisfacer el requerimiento de la nueva llamada que entra al sistema. La función de costos del vehículo elegido será la siguiente:

$$J_j = \left\{ \Delta C_j \right\}_{S_j(k)} = \{C_j(k+1) - C_j(k)\} \quad (11)$$

La expresión (11) debe ser computada para cada secuencia posible  $S_j(k)$  en el instante  $k$ . La matriz de secuencia óptima  $S_j^*(k)$  se obtiene comparando los costos en (11) entre todas las secuencias factibles. Analíticamente,  $S_j^*(k) = \arg \min_{S_j(k)} \{J_j(S_j(k))\} \forall S_j(k)$ .

Conceptualmente,  $J_j$  representa el costo de inserción computado en tiempo real dentro de una secuencia particular cuando el sistema acepta una nueva llamada (en tiempo  $k+1$ ). Finalmente, la llamada se asigna al vehículo con menor costo, es decir,  $j^* = \arg \min_j \{J_j(S_j^*(k))\} \forall j: 1..F$ . Se utiliza la siguiente expresión para la función de costos:

$$C_j(k+1) = \sum_{i=1}^{w_j} \left\{ \underbrace{[L_j^i(k+1) + 1][T_j^i(k+1) - T_j^{i-1}(k+1)]}_{\text{TiempoViaje}} + \underbrace{r_j^i(k+1) \cdot \alpha \cdot T_j^i(k+1)}_{\text{TiempoEspera}} \right\} \quad (12)$$

donde  $L_j^l(k+1)$  y  $T_j^l(k+1)$  son el tiempo de llegada y la carga pertenecientes a la secuencia del vehículo en un instante dado  $(k+1)$  para cada parada  $l$ , respectivamente. Adicionalmente,

$$r_j^i = \begin{cases} 1 & \text{si el elemento } i\text{ésimo de la secuencia del vehiculo } j \text{ es un pick up} \\ 0 & \text{Caso contrario} \end{cases}$$

La función de costos  $C_j(k+1)$  en (12), se puede separar en una componente de tiempo de espera y otra de tiempo de viaje. Ambas están escritas en función del tiempo de llegada y de la carga del vehículo (representada por el número de pasajeros más el conductor de cada vehículo con el fin de incorporar de alguna forma el costo del operador por mover vehículos vacíos). La componente de costo de espera es ponderada por un coeficiente  $\alpha$ . Usualmente  $\alpha > 1$ , de donde se asume que los pasajeros están dispuestos a pagar más por disminuir el tiempo de espera que el tiempo de viaje.

## 5.2 Esquema de control a dos pasos

La función de costos asociada al esquema de control predictivo a dos pasos requiere que el cálculo de una expresión para  $J_j$  dependa de la incertidumbre de la posición de la llamada que podría entrar en el futuro al sistema. En la formulación planteada en este trabajo, la decisión de asignar dicha llamada que aparecerá en el futuro (predecir la llamada futura) se basa en los datos históricos del sistema. Para especificar la función de costos, se utiliza una expresión para la probabilidad de la llamada potencial que recibirá

el sistema en el instante de tiempo futuro, de acuerdo a su ubicación espacial dentro de la red urbana.

Complementariamente, se define un centro de masa para las llamadas en un área específica como la coordenada geográfica de mayor concentración de llamadas a ocurrir durante un periodo de tiempo específico. Para realizar la predicción a dos pasos, se requiere de una probabilidad asociada a que un nuevo requerimiento (como par origen-destino) ocurra dentro de un par de áreas geográficamente cercanas al trayecto del vehículo, en un intervalo de tiempo conocido. Entonces, la nueva entrada de llamadas tiene una probabilidad  $p_h^{\Delta T(k+2)}$ , donde  $h$  denota el par específico de zonas y  $\Delta T(k+2)$  representa el intervalo de tiempo que contiene al instante  $(k+2)$ , y que dependerá de cuan fino se quiera discretizar el tiempo total de modelación considerando la calidad de la información histórica. El tiempo entre llamadas se asume siguiendo una exponencial negativa. Con el fin de aplicar esta metodología, la zona de estudio es dividida en áreas más pequeñas, tomando en cuenta cada problema específico. El controlador predictivo a dos pasos selecciona la secuencia de aquel vehículo que minimiza la función de costos global. Esta función de costos representa el costo de la inserción potencial de una posible nueva llamada que aparece cerca de la trayectoria espacial en un intervalo de tiempo dado, y es función de la probabilidad  $p_h^{\Delta T(k+2)}$  de ocurrencia de la siguiente llamada entre un par de zonas  $h$ , en un intervalo de tiempo  $\Delta T(k+2)$  y que cambia dinámicamente dependiendo de la intensidad de la demanda en un periodo particular de tiempo asociada a cada par de zonas  $h$ . El criterio de asignación de un requerimiento a un vehículo está determinado por la secuencia óptima en cada instante de tiempo. La asignación de la llamada en  $(k+1)$  en este caso se determina de resolver el siguiente problema de optimización:

$$\text{Min } J(S(k)) = \sum_{j=1}^F C_j(k+2) \Big|_{S_j(k)}$$

de donde se obtiene  $S^*(k) = (S_1^*(k), S_2^*(k), S_3^*(k), \dots, S_F^*(k))$  y la llamada será asignada al vehículo que corresponda según lo establecido en  $S^*(k)$ . La expresión que representa al costo de un vehículo dado en el instante futuro  $(k+2)$  es la siguiente:

$$C_j(k+2) = \sum_{h \in H} \sum_{i=1}^{w_j} p_h^{\Delta T(k+2)} \{ [L_j^{ih}(k+2) + 1] [T_j^{ih}(k+2) - T_j^{(i-1)h}(k+2)] + r_j^{ih}(k+2) \cdot \alpha \cdot T_j^{ih}(k+2) \} \quad (13)$$

donde  $L_j^{ih}(k+2)$  es la carga del vehículo  $j$  en el instante  $(k+2)$  para cada parada  $i$  (recogida o entrega) dado que la llamada potencial ocurre en el par  $h$ .  $T_j^{ih}(k+2)$  es el tiempo de arribo esperado del vehículo  $j$  en el instante  $(k+2)$  a cada parada  $i$  (recogida o entrega) dado que la llamada potencial ocurre en el par  $h$ .

Se puede demostrar que la optimización en este caso es independiente del paso intermedio  $k+1$ , sin embargo las variables en (13) dependen implícitamente de la decisión que se haya tomado en el paso intermedio, totalmente definidas a partir del vector  $S(k)$ . Además, y tal como se mencionó anteriormente, la probabilidad que una nueva llamada aparezca en un par específico  $h \subseteq H$  dentro del intervalo de tiempo  $\Delta T(k+2)$  se computa como

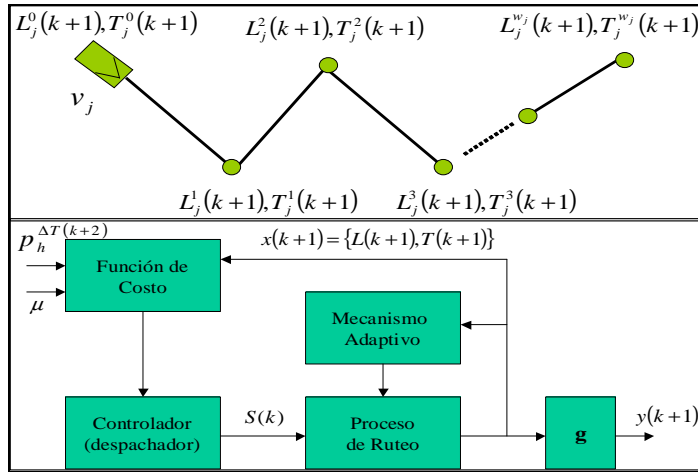
$$p_h^{\Delta T(k+2)} = \frac{N_h^{\Delta T(k+2)}}{\sum_{g \in H} N_g^{\Delta T(k+2)}} \quad (14)$$

donde  $N_h^{\Delta T(k+2)}$  es el número total de requerimientos de viaje observados entre un par de zonas  $h$ , en un intervalo específico de tiempo  $\Delta T(k+2)$ . Notar que  $\sum_{h \in H} p_h^{\Delta T(k+2)} = 1$ .

Gráficamente el sistema trabaja tal como se ilustra en la figura 1. El proceso de ruteo ha sido representado por las variables de estado carga y tiempo del vehículo para un instante futuro. Del bloque **g** se obtiene las variables observadas (tiempos de viaje y espera finales), y que servirán para determinar la calidad de la modelación, y la necesidad de adaptar los parámetros del modelo. El controlador genera la mejor secuencia entre todas las posibles y finalmente el mecanismo adaptivo corresponde al mecanismo de variación (actualización) de las matrices **A, B** con el propósito de actualizar los vectores de carga y tiempo de arribo a la dimensión de la secuencia.

## 6. PRUEBAS EXPERIMENTALES

En esta sección se sintetiza una primera implementación del método vía simulación a partir de datos históricos de viaje en la ciudad de Santiago de Chile, obtenidos de la encuesta origen destino realizada el año 2001. Con este experimento, queremos visualizar las ventajas de utilizar control predictivo a dos pasos versus decisiones en tiempo real a un paso. Se eligió realizar la simulación para el periodo de punta mañana (de 7:00 a 9:00).



**Figura 1 a) Secuencia de un vehículo con variables de estado, b) Esquema de control híbrido adaptivo predictivo (HAPC) aplicado al problema de ruteo dinámico de vehículos**

Se asume que la distribución de demanda por el servicio de transporte propuesto (servicio puerta a puerta similar al radiotaxi), sigue la misma distribución de viajes motorizados observada de la encuesta. Se observa que el 90% de estos viajes ocurren en seis pares de comunas de Santiago ( $H=6$ ). Las probabilidades promedio observadas en el período de análisis (se asumen constantes durante todo el período) son entonces, las que se observan en tabla 1. La flota de vehículos considerada es de tamaño 10 (cada uno con capacidad para 6 pasajeros). Los vehículos inician su recorrido en centros de masa preasignados, promediando una velocidad comercial de 20 kilómetros por hora, siguiendo las secuencias generadas en tiempo real de acuerdo a la distribución de requerimientos de servicio.

**Tabla 1. Probabilidades de ocurrencia de 1 llamada por zona.**

Origen	Destino	Probabilidad
Puente Alto	Puente Alto	0.3048
Vitacura	Santiago	0.1940
La Florida	La Florida	0.1859
Las Condes	Las Condes	0.1399
Maipú	Santiago	0.0914
Providencia	Providencia	0.0840

Fuente: SECTRA Santiago de Chile, 2005

Se generaron 50 repeticiones de 60 llamadas aleatorias (siguiendo el patrón histórico) durante el período de dos horas, obteniéndose los tiempos de viaje y espera de servicio promedio (y desviación estándar) para 40 llamadas en cada repetición que se resumen en tabla 2 (considerando un período efectivo de régimen entre la llamada 10 y la 50 aproximadamente para evitar distorsiones por los límites temporales de la prueba).

**Tabla 2. Tiempos de espera y viaje registrados bajo dos estrategias de control.**

Estrategia de Control	Tiempos Promedios (min)			Desviacion Estandar (min)		
	Espera	Viaje	Total	Espera	Viaje	Total
Un paso	30.96	21.909	52.869	12.108	11.346	23.454
Dos pasos	21.491	14.99	36.481	10.058	8.916	18.974

De la tabla 2 se aprecian los considerables beneficios (ahorros de cerca de 16 minutos en el tiempo total considerando viaje y espera, en promedio) de utilizar un esquema predictivo por sobre una asignación en tiempo real sin predicción futura (un paso), incluso considerando un esquema de optimización muy razonable.

## **7. COMENTARIOS Y CONCLUSIONES**

En este trabajo se muestra una forma innovadora de modelar demoras provenientes de reruteos de vehículos provocados por la necesidad de servir demanda incierta en el contexto de un problema de recoger y dejar pasajeros con decisiones de programación en tiempo real por demanda dinámica. Se muestra que si el problema es modelado como un problema de control del tipo adaptivo predictivo híbrido, se logra obtener considerables beneficios a partir de las decisiones de programación en tiempo presente tomando en cuenta la entrada al sistema de demanda futura incierta.

Considerando lo promisorio de los resultados, en la actualidad se está desarrollando esquemas de programación para horizontes de evaluación superiores a dos pasos. Adicionalmente, se está trabajando en mejorar tanto las especificaciones de las funciones de costo, así como los algoritmos de solución utilizados (mejores opciones que enumeración explícita). De hecho, a esta altura se tienen resultados a partir de la incorporación de algoritmos genéticos en la optimización de las metodologías, tanto a uno como a dos pasos (Núñez *et. al*, 2005). En aplicaciones aún más sofisticadas, se espera contar con expresiones analíticas para los modelos en variables de estado, particularmente las matrices A y B como dependientes de la secuencia de los vehículos, y una forma funcional más rigurosa para la función de costos de decisión (Núñez *et. al*, 2005). Finalmente, la forma de internalizar la información histórica vía probabilidades también puede ser mejorada a partir de un mejor entendimiento de la data histórica. Se está estudiando la forma de usar data histórica (tal como la que se usa en este trabajo) en combinación con información histórica reciente generada y procesada en tiempo real en un horizonte previo a la toma de decisiones.

## **AGRADECIMIENTOS**

Esta investigación ha sido parcialmente financiada por los proyectos FONDECYT 1030700 y 1040698, y por el Núcleo Milenio, en su proyecto “Sistemas Complejos de Ingeniería”.

## REFERENCIAS

Bemporad A., y M. Morari (1999). Control of systems integrating logic, dynamics and constraints, **Automatica**, 35, 407-427.

Bemporad, A., F. Borrelli y M. Morari (2002). Model predictive control based on linear programming. The explicit solution, **IEEE Transactions Automatic Control**, 47 (12), 1974-1985.

Bertsekas D., y J. N. Tsitsiklis (1996), **Neuro-Dynamic Programming**, Athena Scientific.

Bertsimas, D., y D. Simchi-Levi (1996), The New Generation of Vehicle Routing Research: Robust Algorithms Addressing Uncertainty, **Operations Research** 44, 286-304.

Bertsimas, D., P. Jaillet and A. Odoni (1990). A priori optimization **Operations Research**, 38, 6, 1019-1033.

Borrelli, F., M. Baotic, A. Bemporad and M. Morari (2003), An Efficient Algorithm for Computing the State Feedback Optimal Control Law for Discrete Time Hybrid Systems, **Proceedings of 2003 American Control Conference**, Denver, Colorado, USA.

Cortés C.E., y R. Jayakrishnan (2004). Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems. **Proceeding of the Triennial Symposium on Transportation Analysis (TRISTAN) V**, Guadeloupe, French West Indies.

Desrosiers, J., Y. Dumas, M. Salomon y F. Soumis (1995). Time Constrained Routing and Scheduling, en **Handbooks in Operations Research / Management Science**, Volume 8, Network Routing, pp. 35-139, Elsevier Science Publishers B.V.

Ichoua, S., M. Gendreau y J. Potvin (2000). Diversion Issues in real-time vehicle dispatching, **Transportation Science**, 34 (4), 426-435.

Jaw, J., A. Odoni, H. Psaraftis y N. Wilson. (1986), A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, **Transportation Research B**, 20(3), 243-257.

Malucelli, F., M. Nonato y S. Pallottino (1999). Demand Adaptive Systems: some proposals on flexible transit, **Operational Research in Industry**, T.A. Ciriani, et al., Editors, McMillan Press: London, 157-182.

Minkoff A.S. (1993). A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching. **Operations Research**, 41(1), 77-90.

Núñez, A., D. Sáez y C.E. Cortés (2005). Problema de ruteo dinámico de una flota de vehículos con un enfoque de control predictivo híbrido basado en algoritmos genéticos, **Actas del XII Congreso Chileno de Ingeniería de Transporte**, Valparaíso, Chile.

Psaraftis, H. (1988). Dynamic vehicle routing problems, en B.L. Golden and A.A. Assad editors, **Vehicle routing methods and studies**, 223-248.

Psaraftis, H. (1986) Scheduling large-scale advance-request dial-a-ride systems, **American Journal of Math. And Mgmt. Sciences**, 6, (3/4), 327-367.

Savelsbergh, M., y M. Sol (1995), The general pickup and delivery problem, **Transportation Science** 29(1), 17-29.

Zhu, K.Q., y K.L Ong (2000). A Reactive Method for Real Time Dynamic Vehicle Routing Problems, **Proceedings of the 12th IEEE International Conference on Tools for Artificial Intelligence, ICTAI 2000**, Vancouver, Canada.