Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 35 (2008) 3412 – 3438

computers & operations research

www.elsevier.com/locate/cor

Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering

Doris Sáez[a,*], Cristián E. Cortés[b], Alfredo Núñez[a]

[a] *Electrical Engineering Department , Universidad de Chile, Av. Tupper 2007, Santiago, Chile*
[b] *Civil Engineering Department, Universidad de Chile, Blanco Encalada 2002, Santiago, Chile*

Available online 9 February 2007

Abstract

In this paper, we develop a family of solution algorithms based upon computational intelligence for solving the dynamic multi-vehicle pick-up 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 dispatching 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

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 researchers, 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

* Corresponding author. Tel.: +56 2 9784207; fax: +56 2 6720162.
E-mail addresses: dsaez@ing.uchile.cl (D. Sáez), ccortes@ing.uchile.cl (C.E. Cortés).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2007.01.025

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 (HPC) framework for the dynamic PDP formulated by the authors in a previous work.

Over the last few 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 Gendreau et al. [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 the 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 DPDP, 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] postulated 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] describe 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. 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 efficient solution algorithms (based on genetic algorithm heuristic techniques, GA) 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, GA 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 Tighe et al. [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. Skrlec 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, Zhu [32] describes specialized GA 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, Haghani and Jung [34] applied a GA optimization method for the multi-vehicle dynamic VRP with time-dependent travel time and soft time windows. This method 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 metaheuristics inspired by the behavior of real ant colonies (ant colony optimization) 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 optimization) have been applied only on myopic dynamic VRP formulations without considering future demand scenarios for improving current dispatch decisions. In this paper, we 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, a formal analytical formulation of the HAPC state space model for DPDP is presented. 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 HAPC-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 Cortés et al. [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 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

$$x(k+1) = f(x(k), u(k)),$$
$$y(k) = g(x(k)), \tag{1}$$

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 HPC design minimizes the following generic objective function [40]:

$$\min_{\{u(k),u(k+1),\ldots,u(k+N-1)\}} J(u(k),\ldots,u(k+N-1),\hat{x}(k+1),\ldots,\hat{x}(k+N),\hat{y}(k+1),\ldots,\hat{y}(k+N)), \tag{2}$$

where J is an 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 $\{u(k),\ldots,u(k+N-1)\}$ 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 (τ), 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 it 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 $S(k)=\{S_1(k),\ldots,S_j(k),\ldots,S_F(k)\}$ 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 is 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,

$$S_j(k) = \begin{bmatrix} z_j^0(k) & 1-z_j^0(k) & P_j^0(k) & id_0(k) & \mu(id_0(k)) \\ z_j^1(k) & 1-z_j^1(k) & P_j^1(k) & id_1(k) & \mu(id_1(k)) \\ z_j^2(k) & 1-z_j^2(k) & P_j^2(k) & id_2(k) & \mu(id_2(k)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_j^{w_j(k)}(k) & 1-z_j^{w_j(k)}(k) & P_j^{w_j(k)}(k) & id_{w_j(k)}(k) & \mu(id_{w_j(k)}(k)) \end{bmatrix}, \tag{3}$$

$$T_j(k) = \begin{bmatrix} T_j^0(k) \\ T_j^1(k) \\ T_j^2(k) \\ \vdots \\ T_j^{w_j(k)}(k) \end{bmatrix}, \quad L_j(k) = \begin{bmatrix} L_j^0(k) \\ L_j^1(k) \\ L_j^2(k) \\ \vdots \\ L_j^{w_j(k)}(k) \end{bmatrix}, \tag{4}$$

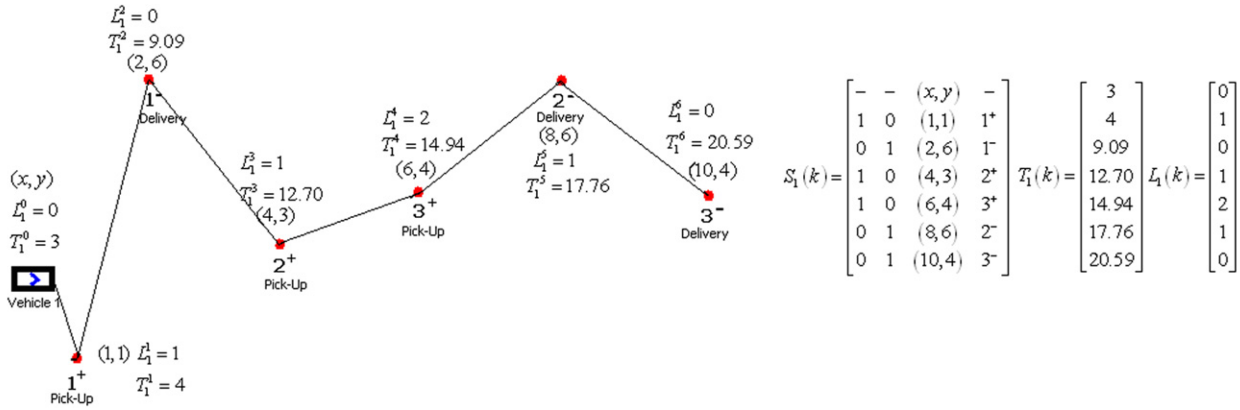

Fig. 1. Example of vehicle 1 sequence and associated state space variables.

where

$$z_j^i(k) = \begin{cases} 1 & \text{if stop } i \text{ defined at } k \text{ is a pick-up,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

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 represent 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, $z_j^{w_j(k)}(k) = 0, \forall j : 1 \ldots F$.

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 Eqs. (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 Eqs. (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 $L_j^i(k) \leqslant L_{\max}$.

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 was 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.

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

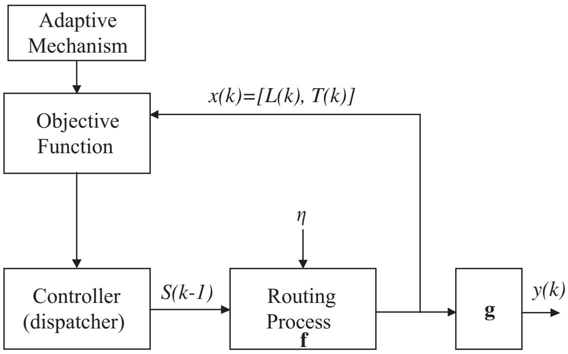

Fig. 2. Overall block diagram of a hybrid predictive approach for DPDP.

(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 (η) through a fuzzy zoning (see Section 3).

The authors propose the following scheme:

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

$$y(k) = T(k) = [T_1(k), \ldots, T_j(k), \ldots, T_F(k)], \tag{6}$$

where

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

$$L(k) = [L_1(k), \ldots, L_j(k), \ldots, L_F(k)],$$

$$S(k) = [S_1(k), \ldots, S_j(k), \ldots, S_F(k)]. \tag{7}$$

The functions f_L and f_T in Eqs. (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 $G = (g_{ir})_{\substack{i=1\ldots n \\ r=1\ldots n}}$ 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 rth 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 $(f_1, f_2, \ldots, f_n)$ using the binary factors of activation g_{ir}. Analytically,

$$P_j^i(k) = g_{i1} f_1 + g_{i2} f_2 + \cdots + g_{ii} f_i + \cdots + g_{in} f_n, \tag{8}$$

where

$$g_{ir} = \begin{cases} 0, & f_r \text{ is not stop } i, \\ 1, & f_r \text{ is stop } i. \end{cases} \tag{9}$$

Therefore, the stop position vector $P_j(k)$, excluding the initial condition $P_j^0(k)$, can be written as follows:

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1(n-1)} & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2(n-1)} & g_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1} & g_{(n-1)2} & \cdots & \cdots & g_{(n-1)(n-1)} & g_{(n-1)n} \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{n(n-1)} & g_{nn} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G \cdot f. \tag{10}$$

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:

$$g_{i1} + g_{i2} + \cdots + g_{in} = 1, \quad \forall i = 1, \ldots, n, \tag{11}$$

$$g_{1r} + g_{2r} + \cdots + g_{nr} = 1, \quad \forall r = 1, \ldots, n. \tag{12}$$

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):

$$\hat{L}_j(k+1) = \left[L_j^0(k) \;\; \cdots \;\; L_j^0(k) + \sum_{m=1}^{i} \left(\sum_{r \in P} \mu(id_r(k)) g_{mr} - \sum_{r \in D} \mu(id_r(k)) g_{mr} \right) \;\; \cdots \;\; \cdots \;\; 0 \right]^{\mathrm{T}}, \tag{13}$$

where $\mu(id_r(k))$ equals the 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,

$$L_j^0(k) + \sum_{m=1}^{i} \left(\sum_{r \in P} \mu(id_r(k)) g_{mr} - \sum_{r \in D} \mu(id_r(k)) g_{mr} \right) \leqslant L_{\max}, \quad i = 2, \ldots, n-1. \tag{14}$$

In addition, and to complete the state space model, the departure time vector can be expressed as function of the matrix G. In short,

$$\hat{T}(k+1) = \left[T^0(k) \;\; T^0(k) + G^1 Q(k) G^{2^{\mathrm{T}}} \;\; \cdots \;\; T^0(k) + \sum_{r=1}^{i-1} G^r Q(k) G^{r+1^{\mathrm{T}}} \;\; \cdots \;\; T^0(k) + \sum_{r=1}^{n-1} G^r Q(k) G^{r+1^{\mathrm{T}}} \right]^{\mathrm{T}} \tag{15}$$

with G^r denotes the rth 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, an expansion and reduction matrix size technique is developed (adaptive behavior) to capture the dynamic effect caused by the real operation. The idea is to either increase or reduce the stop position vector shown in Eq. (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 Cortés [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

$$P_j(k) = \begin{bmatrix} P_j^1(k) \\ P_j^2(k) \\ \vdots \\ \vdots \\ P_j^{n-1}(k) \\ P_j^n(k) \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & \cdots & g_{1(n-1)} & g_{1n} \\ g_{21} & g_{22} & \cdots & \cdots & g_{2(n-1)} & g_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{(n-1)1} & g_{(n-1)2} & \cdots & \cdots & g_{(n-1)(n-1)} & g_{(n-1)n} \\ g_{n1} & g_{n2} & \cdots & \cdots & g_{n(n-1)} & g_{nn} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = G \cdot f, \tag{16}$$

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 in previous assigned sequences; by keeping the order of previously scheduled stops (only insertions on previous segments are allowed). As previous sequences hold, $(f_1, f_2, \ldots, f_{n-2})$, 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). Analytically,

$$P_i(k) = \begin{cases} g_{11} f_1 + g_{1,n-1} f_{n-1} = (x_1, y_1) & \text{if } i = 1, \\ g_{21} f_1 + g_{22} f_2 + g_{2,n-1} f_{n-1} + g_{2,n} f_n = (x_2, y_2) & \text{if } i = 2, \\ g_{i,i-2} f_{i-2} + g_{i,i-1} f_{i-1} + g_{i,i} f_i + g_{i,n-1} f_{n-1} + g_{i,m} f_n = (x_i, y_i) & \text{if } i = 3, \ldots, (n-2), \\ g_{n-1,n-3} f_{n-3} + g_{n-1,n-2} f_{n-2} + g_{n-1,n-1} f_{n-1} & \\ \quad + g_{n-1,n} f_n = (x_{n-1}, y_{n-1}) & \text{if } i = n-1, \\ g_{n,n-2} f_{n-2} + g_{n,n} f_n = (x_n, y_n) & \text{if } i = n, \end{cases} \tag{17}$$

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 pick-up stop request or the new delivery stop, and so on.

Eq. (17) can also be written in the form of general expression (16), obtaining the following sparse G matrix (optimization decision matrix):

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

This analytical problem formulation allows us to generalize the N-step ahead optimization criteria defined in the next section and to evaluate different non-linear 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 Section 6.

Once the state space variables are analytically defined in Section 2.1 (Eqs. (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 ($k+2, k+3, \dots,$ 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 objective function for a generic prediction horizon N, can be written as follows:

$$\underset{\{S(k),S(k+1),\dots,S(k+N-1)\}}{\mathrm{Min}} J=\sum_{t=1}^{N}\sum_{j=1}^{F}\sum_{h=1}^{h_{\max}(k+t)} p_h^{\Delta T(k+t)}\cdot((C_j(k+t)-C_j(k+t-1))|_{S_j(k+t-2),h}), \tag{18}$$

$$C_j(k+t)|_{S_j(k+t-2),h} = \sum_{i=1}^{w_j(k+t)} \left\{ \underbrace{[\hat{L}_j^{i-1}(k+t)+1](\hat{T}_j^i(k+t)-\hat{T}_j^{i-1}(k+t))}_{J \text{ travel time}} + \underbrace{z_j^i(k+t-1)\alpha(\hat{T}_j^i(k+t)-T_j^0(k+t))}_{J \text{ waiting time}} \right\} \Bigg|_{S_j(k+t-2),h}, \tag{19}$$

where $k+t$ is the instant at which the tth 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 hth request type (associated with a specific pair of zones, as discussed before 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. $C_j(k+t)|_{S_j(k+t-2),h}$ 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 α.

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,

$$\begin{aligned} C_j(k+t)|_{S_j(k+t-2),h} &= \sum_{i=1}^{w_j(k+t)} \left[1 + L_j^0(k+t-1) + \sum_{m=1}^{i} \left(\sum_{r\in P} \mu(id_r(k+t-1))g_{mr} - \sum_{r\in D} \mu(id_r(k+t-1))g_{mr} \right) \right] \\ &\times [G^{(i-1)}Q(k+t-1)G^{i^{\mathrm{T}}}] + \alpha \cdot z_j^i(k+t-1) \left(\sum_{r=1}^{i-1} G^r Q(k+t-1) G^{r+1^{\mathrm{T}}} \right). \end{aligned} \tag{20}$$

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),

$$J = \sum_{j=1}^{F} (C_j(k+1) - \overbrace{C_j(k)}^{\text{known constant}})|_{S_j(k-1)}. \tag{21}$$

The difference $(C_j(k+1) - C_j(k))|_{S_j(k-1),1}$ 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 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

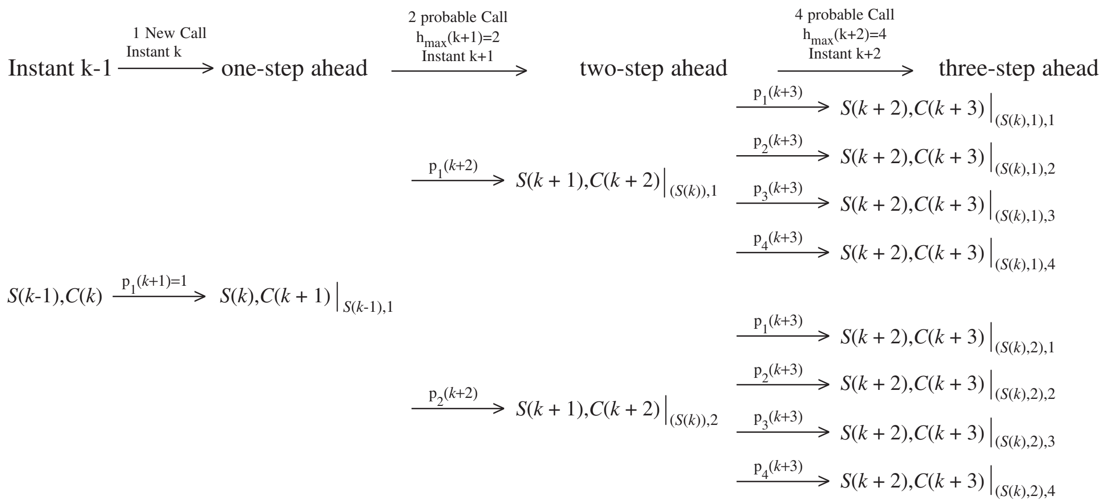

Fig. 3. Potential combinations of sequences at future.

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:

$$J = \sum_{j=1}^{F} \left[\sum_{h=1}^{h_{\max}(k+2)} p_h^{\Delta T(k+2)}(k+2) \cdot C_j(k+2)|_{S_j(k),h} - \overbrace{C(k)}^{\text{known constant}} \right]. \tag{22}$$

Another interesting case, is the three-step ahead objective function, again computed from the generic expression (17), as follows:

$$J = \sum_{j=1}^{F} \left[\sum_{h_2=1}^{h_{\max}(k+2)} p_{h_2}(k+2) \cdot \left(\sum_{h_3=1}^{h_{\max}(k+3)} p_{h_3}(k+3) \cdot C_j(k+3)|_{S_j(k),h_2,h_3} \right) - \overbrace{C(k)}^{\text{known constant}} \right]. \tag{23}$$

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.

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 a typical classic zoning approach 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:

$$S(c) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^m \|x_k - v_i\|^2, \tag{24}$$

where n is the data number, c is the number of clusters, u_{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 FCM algorithm are found in Babuska [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 below:

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 to which 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.

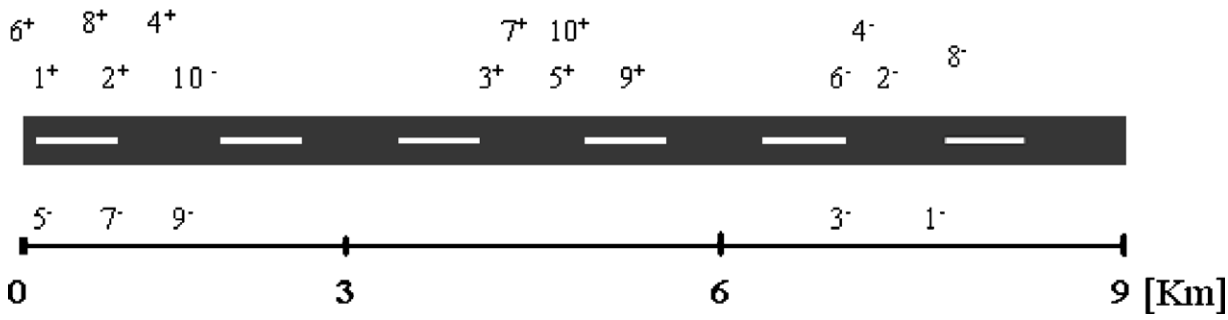

Fig. 4. Single vehicle requests in a certain period of time.

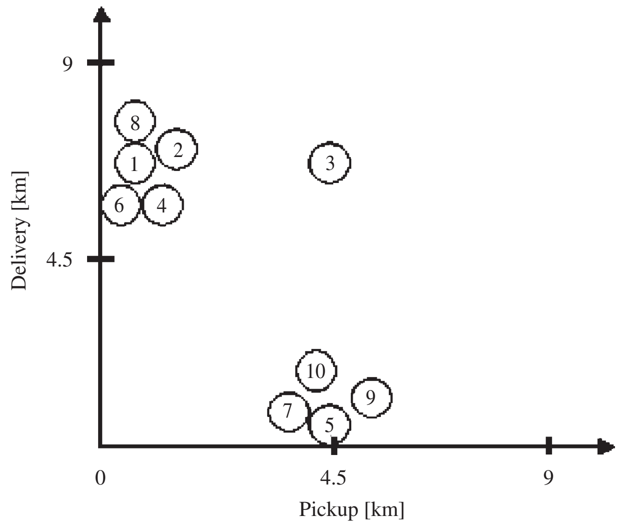

Fig. 5. Pick-up–delivery coordinates of historical demand over a certain time period.

For example, the FCM model performs quite well for jumbled up trip patterns, in which representative zones could be spatially overlapped. Next, a one-dimensional example is shown to illustrate the application of the method in the context of the DPDP.

3.2. 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 3.1. 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 10 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. 5 shows a two-dimensional 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.

From the historical data shown in Fig. 5, the FCM 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 two and three fuzzy clusters, respectively. As explained in Section 3.1, the cluster centers are obtained and denoted by “x” marks in the figure.

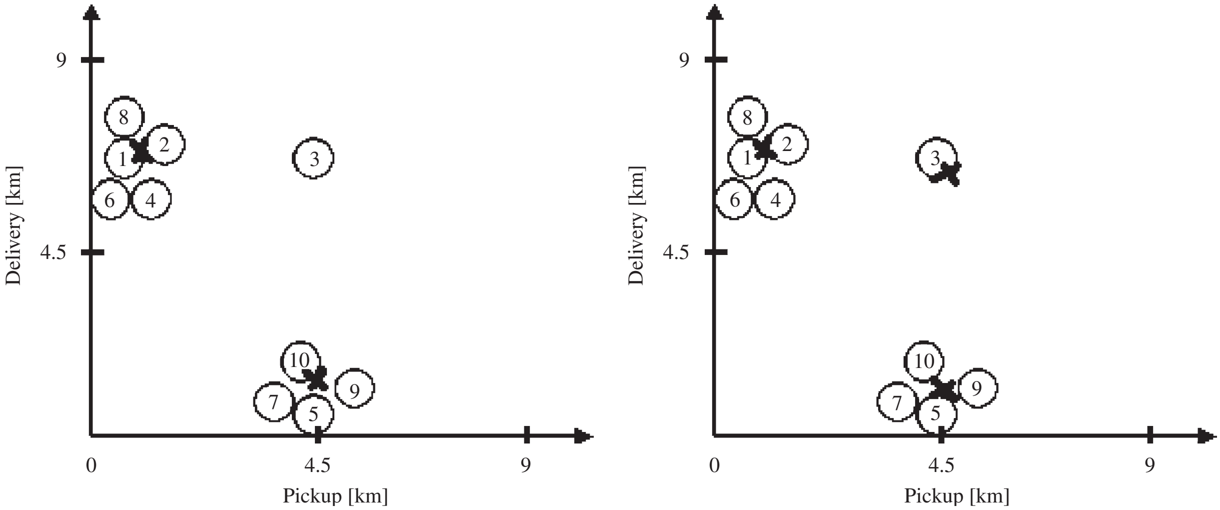

Fig. 6. Cluster centers for two and three clusters selected.

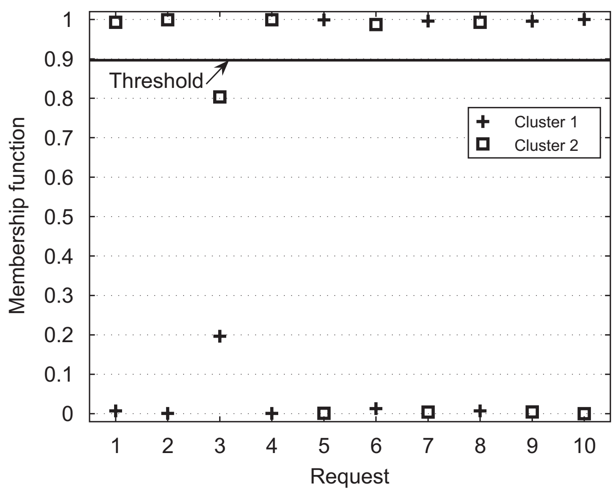

Fig. 7. Membership degree of historical demand over a certain time period for two clusters.

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 two clusters instead of three, 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 10 call requests for two 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.

Finally, and using the FCM procedure, the probabilities associated with trip patterns are shown in Table 1 for two fuzzy clusters.

Table 1
Probabilities for the trip patterns using two fuzzy clusters

Trip pattern	Pick-up 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 HPC has shown to be very useful for the control of hybrid systems and authors like Bemporad and Morari [40], Bemporad et al. and Thomas et al. [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 to solve 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, 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 GA 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.

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

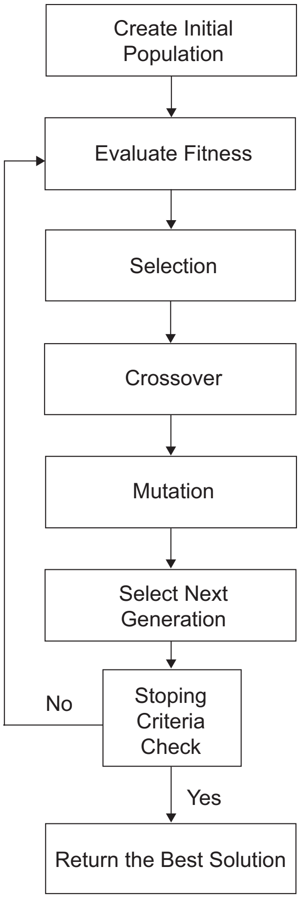

Fig. 8. GA flowchart.

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 pick-up 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 followingvector $P_j(k-1)$, as defined in Section 2.1, associated with the sequence at the previous instant $k-1$ $(S_j(k-1))$:

$$P_j(k-1) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \end{bmatrix}}_{f}, \tag{25}$$

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.

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

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 seven individuals for vehicle j is presented by considering the previous sequence (given by expression (25)) and the new call request:

$$\text{Population} \Leftrightarrow \begin{pmatrix} \text{Individual 1} \\ \text{Individual 2} \\ \text{Individual 3} \\ \text{Individual 4} \\ \text{Individual 5} \\ \text{Individual 6} \\ \text{Individual 7} \end{pmatrix} \Leftrightarrow \begin{pmatrix} (j,1,4) \\ (j,1,6) \\ (j,5,6) \\ (j,3,5) \\ (j,4,6) \\ (j,1,6) \\ (j,2,4) \end{pmatrix} \Leftrightarrow \begin{pmatrix} j, \boxed{3^+} \to 1^+ \to 2^+ \to \boxed{3^-} \to 1^- \to 2^- \\ j, \boxed{3^+} \to 1^+ \to 2^+ \to 1^- \to 2^- \to \boxed{3^-} \\ j, 1^+ \to 2^+ \to 1^- \to 2^- \to \boxed{3^+} \to \boxed{3^-} \\ j, 1^+ \to 2^+ \to \boxed{3^+} \to 1^- \to \boxed{3^-} \to 2^- \\ j, 1^+ \to 2^+ \to 1^- \to \boxed{3^+} \to 2^- \to \boxed{3^-} \\ j, \boxed{3^+} \to 1^+ \to 2^+ \to 1^- \to 2^- \to \boxed{3^-} \\ j, 1^+ \to \boxed{3^+} \to 2^+ \to \boxed{3^-} \to 1^- \to 2^- \end{pmatrix}. \tag{27}$$

For example, the individual $(j, 1, 4)$ in terms of $P_j(k)$ can be written as

$$\text{Individual 1} \Leftrightarrow P_j(k) = \begin{bmatrix} P_j^1 \\ P_j^2 \\ P_j^3 \\ P_j^4 \\ P_j^5 \\ P_j^6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{G} \cdot \underbrace{\begin{bmatrix} b(1^+) \\ b(2^+) \\ b(1^-) \\ b(2^-) \\ b(3^+) \\ b(3^-) \end{bmatrix}}_{f}. \tag{28}$$

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), $(j, 1, 6)$.

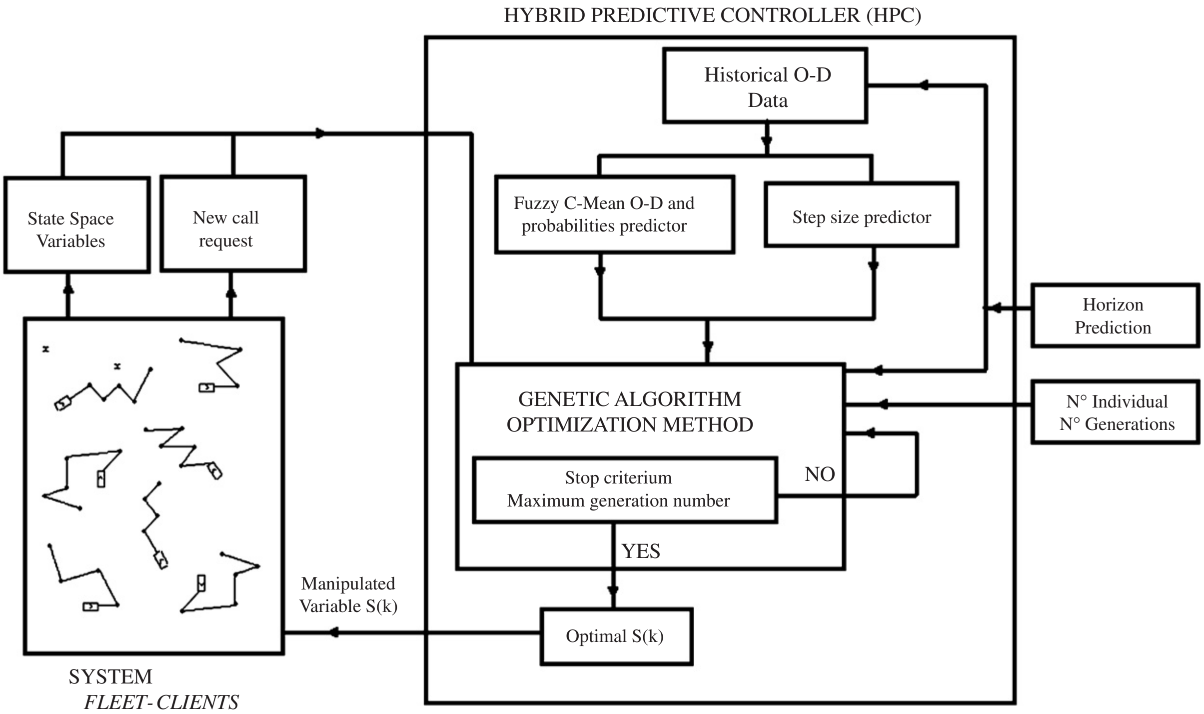

Fig. 9. Overall block diagram of an HAPC for DPDP.

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 EE method. The crossover operator is not applied here since the *no-swapping* constraint has to be satisfied.

Fig. 9 presents the proposed HAPC 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.

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 EE, mainly in computation time saving.

4.3. Example

In this section, illustrative tests using 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 four vehicles and an objective function of two-step ahead with six potential calls are considered. Vehicles cover an urban service area of around 81 km^2, 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

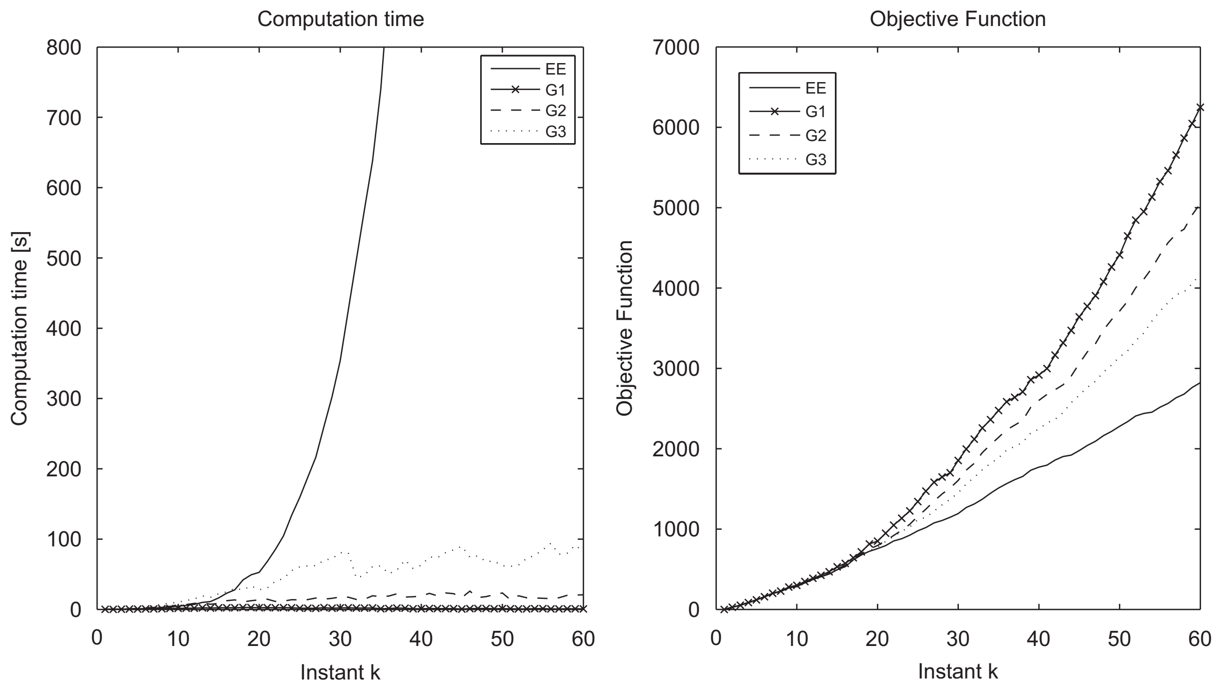

Fig. 10. Computation time and objective function evolution.

Table 2
Objective function and computation time mean

Control strategy Test 1	Objective function mean	Computation time mean
Explicit enumeration EE	1297.4	1536.7
Genetic algorithms G1	2288.2	1.4
Genetic algorithms G2	1945.8	13.9
Genetic algorithms G3	1694.6	49.7

considers five individuals and five 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.

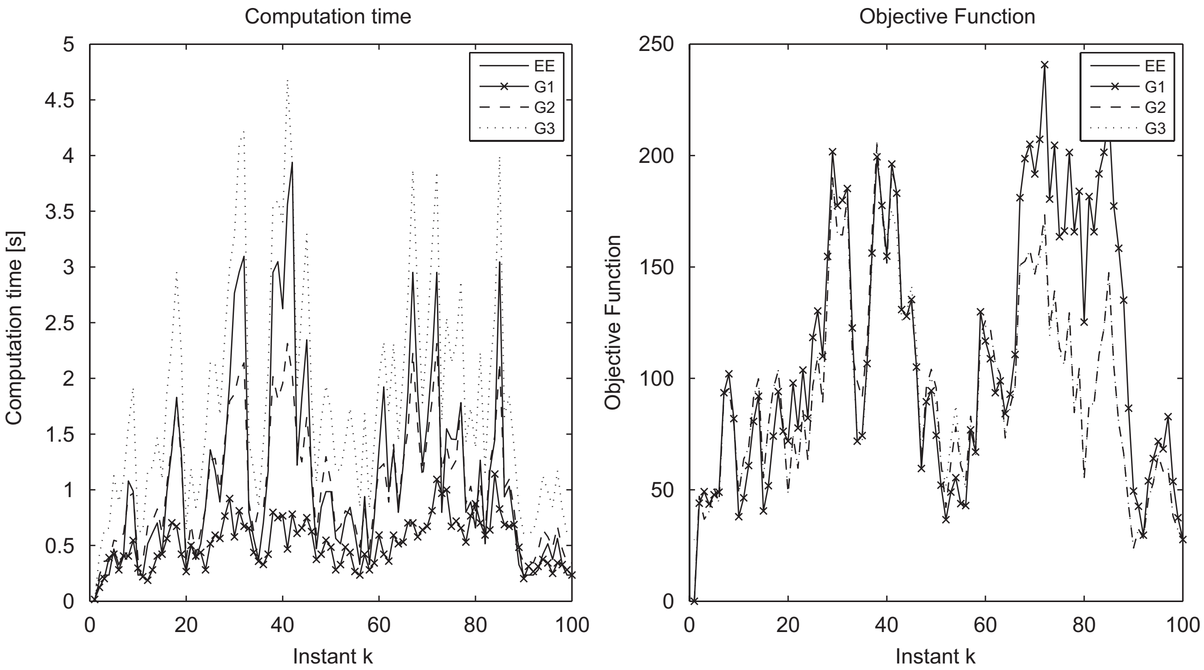

Fig. 11. Computation time and objective function evolution.

Table 3
Objective function and computation time mean

Control strategy Test 2	Objective function mean	Computation time mean
Explicit enumeration EE	94.5	1.1
Genetic algorithms G1	110.9	0.5
Genetic algorithms G2	95.4	1.1
Genetic algorithms G3	94.5	1.8

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.

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.

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 problems).

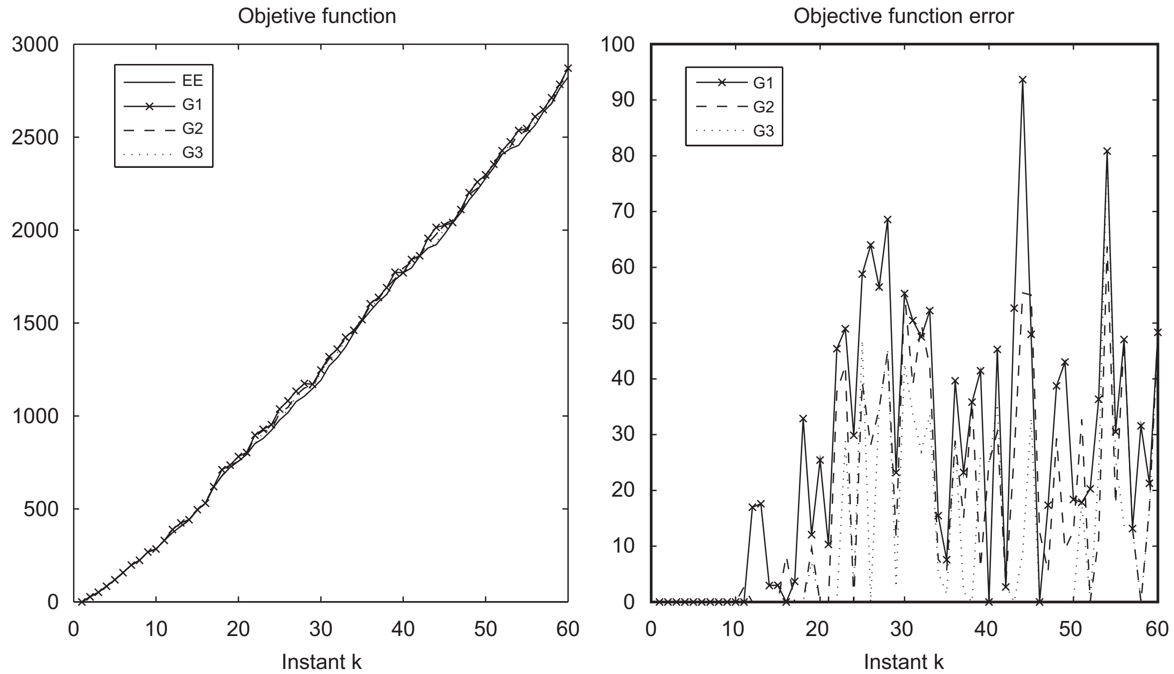

Fig. 12. Computation time and objective function evolution.

Table 4
Objective function and error mean

Control strategy Test 3	Objective function mean	Error mean
Explicit eumeration EE	1297.4	–
Genetic algorithms G1	1324.0	26.6
Genetic algorithms G2	1315.1	17.7
Genetic algorithms G3	1309.3	11.9

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 GA 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^2. 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 working day.

The historical data generated via simulation follows the trips patterns shown in Fig. 13 with arrows.

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,

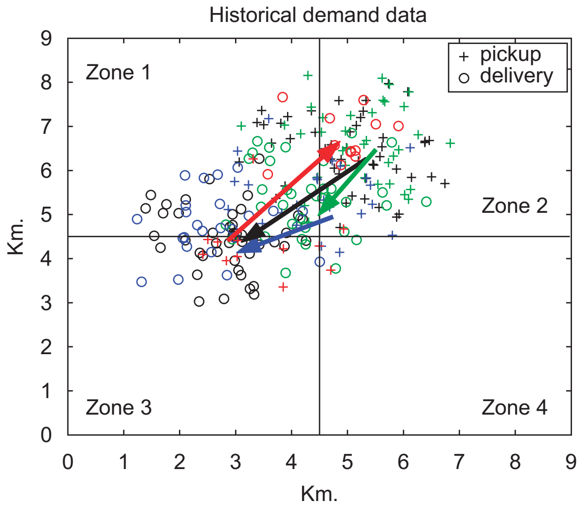

Fig. 13. Origin–destination trip patterns.

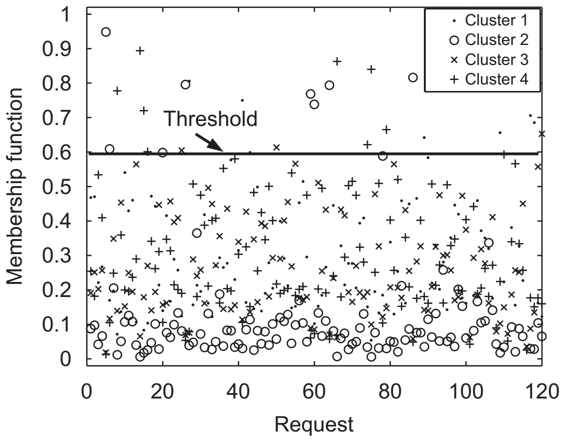

Fig. 14. Membership degree for call requests.

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). Fifty 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 application*

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 EE results were considered for benchmarking.

Fig. 14 shows an application of the procedure described in Section 3.1. In fact, four fuzzy clusters are obtained (Step 1), next their membership degrees are depicted (Step 2). Each call is associated with 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

Table 5
Pick-up and delivery coordinates and probabilities: fuzzy zoning

X pick-up	*Y* pick-up	*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
Pick-up and delivery coordinates and probabilities: classic zoning

X pick-up	*Y* pick-up	*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

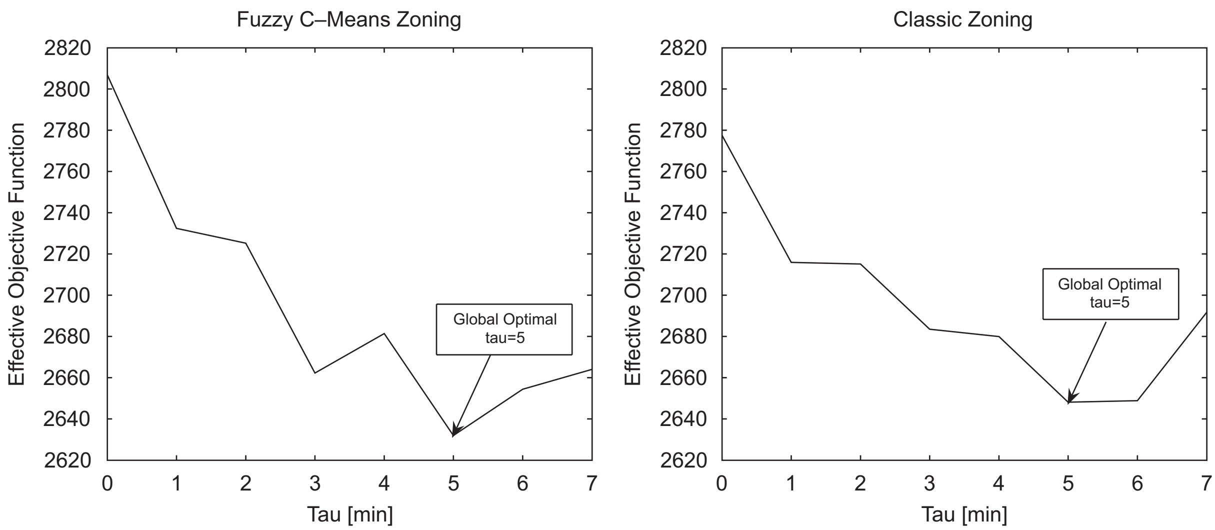

Fig. 15. Sensitivity analysis for τ (classic and fuzzy zonings).

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 pick-up and delivery points of relevant trip patterns and the corresponding probabilities. On the other hand, Table 6 shows the classic zoning based upon four origin–destination pairs.

One fine-tuning parameter is the predicted interval between successive calls, τ, 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 τ values for both classic and fuzzy zonings. Ten replications for each considered τ value were used in order to obtain optimal values. For both zoning methods, the resulting optimal $\tau = 5$.

Using the obtained optimal values of τ, 50 replications of the two-step ahead algorithm based on EE were conducted in order to compare the performance of both zoning methods. Table 7 presents the mean and standard deviations of the

Table 7
Passenger costs

Two-step ahead algorithm	Waiting time (min)		Travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
Classic zoning	6.1437	0.87	10.2358	0.71	16.3795	1.44
Fuzzy zoning	5.9370	0.72	10.1629	0.76	16.0999	1.36
Savings	0.2067		0.0729		0.2796	
Improv. (%)	3.36%		0.71%		1.71%	

Table 8
Vehicle and passenger operational costs

Two-step ahead algorithm	Operational costs (min)		Total effective costs (min)	
	Mean	Std	Mean	Std
Classic zoning	117.9	8.81	2699.4	122.84
Fuzzy zoning	115.7	8.12	2651.1	112.86
Savings	2.2618		48.3163	
Improv. (%)	1.92%		1.79%	

Table 9
Performance comparison for one-, two- and three-step ahead problems

	Waiting time (min)		Travel time (min)		Total time (min)	
	Mean	Std	Mean	Std	Mean	Std
One-step ahead	6.969	0.82	10.877	0.89	17.847	1.46
Two-step ahead	5.921	0.67	10.238	0.79	16.159	1.42
Three-step ahead	5.415	0.53	10.687	0.65	16.102	1.35
Savings 2 step	1.048		0.639		1.688	
Improv. (%)	15.04%		5.87%		9.45%	
Savings 3 step	1.554		0.190		1.745	
Improv. (%)	22.30%		1.75%		9.78%	

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%).

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.

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 problems 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.

Table 10
Vehicle and passenger operational costs

	Operational costs (min)		Effective total costs (min)	
	Mean	Std	Mean	Std
One-step ahead	105.04	9.76	2730.0	127.832
Two-step ahead	105.87	11.68	2568.7	114.516
Three-step ahead	110.86	11.18	2608.0	112.444
Savings 2 step	−0.84		161.27	
Improv. (%)	−0.79%		5.90%	
Savings 3 step	−5.82		122.05	
Improv. (%)	−5.54%		4.47%	

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.

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. There are three major contributions of this paper. 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, 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 dispatching 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 zoning method based on fuzzy clustering 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 a 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 α and τ is planned to be also investigated, for two- and three-step ahead problems. We claim that it is possible to improve the estimation of tuning variables, such as number of probable calls, future step time prediction (τ) 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, Sol M. The general pickup and delivery problem. Transportation Science 1995;29(1):17–29.
[2] Gendreau M, Laporte G, Séguin R. Stochastic vehicle routing. European Journal of Operational Research 1996;88:3–12.
[3] Gendreau M, Laporte G, Séguin R. An exact algorithm for the vehicle routing problem with stochastic customers and demands. Transportation Science 1995;29:143–55.
[4] Yang W, Mathur K, Ballou R. Stochastic vehicle routing problem with restocking. Transportation Science 2000;34(1):99–112.
[5] Secomandi N. Exact and heuristics dynamic programming algorithms for the vehicle routing problem with stochastic demands. PhD 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 & Operations Research 2000;27:1201–25.
[7] Waters C. Vehicle routing problems with uncertainty and omitted customers. Journal of the Operational Research Society 1989;40:1099–108.
[8] Bertsimas D. A vehicle routing problem with stochastic demand. Operations Research 1992;40:574–85.
[9] Bertsimas D, Jaillet P, Odoni A. A priori optimization. Operations Research 1990;38(6):1019–33.
[10] Psaraftis H. Dynamic vehicle routing problems. In: Golden BL, Assad AA, editors. Vehicle routing methods and studies. 1998. p. 223–48.
[11] Psaraftis H. A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem. Transportation Science 1980;14(2):130–54.
[12] Gendreau M, Guertin F, Potvin J, Taillard E. Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science 1999;33:381–90.
[13] Kleywegt A, Papastavrou J. The dynamic and stochastic knapsack problem. Operations Research 1998;46:17–35.
[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 1995;60:193–208.
[15] Dial R. Autonomous dial a ride transit—introductory overview. Transportation Research, Part C 1995;3:261–75.
[16] Spivey M, Powell WB. The dynamic assignment problem. Transportation Science 2004;38(4):399–419.
[17] Ichoua S, Gendreau M, Potvin JY. Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science 2006;40(2):211–25.
[18] Powell WB. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. In: Golden BL, Assad AA, editors. Vehicle routing methods and studies. 1988.
[19] Godfrey G, Powell WB. An adaptive dynamic programming algorithm for stochastic resource allocation problems I: single period travel times. Transportation Science 2002;36:21–39.
[20] Topaloglu H, Powell WB. A distributed decision-making structure for dynamic resource allocation using non-linear functional approximations. Operations Research 2005;53(2):281–97.
[21] Cortés CE, Jayakrishnan R. Analytical modeling of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems. Proceedings of the triennial symposium on transportation analysis (TRISTAN) V, Guadeloupe, French West Indies; 2004.
[22] Cortés CE, Sáez D, Núñez A, Muñoz-Carpintero D. Hybrid adaptive predictive control for a dynamic pick-up and delivery problem, Transportation Science 2007, under review.
[23] Le Bouthillier A, Crainic T. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. Computers & Operations Research 2005;32:1685–708.
[24] Tarantilis C, Ioannou G, Prastacos G. Advanced vehicle routing algorithms for complex operations management problems. Journal of Food Engineering 2005;70:455–71.
[25] Tarantilis C. Solving the vehicle routing problem with adaptive memory programming methodology. Computers & Operations Research 2005;32:2309–27.

[26] Li F, Golden B, Wasil E. Very large-scale vehicle routing: new test problems, algorithms and results. Computers & Operations Research 2005;32:1165–79.

[27] Bianchessi N, Righini G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research 2007;34(2):578–94.

[28] Toht P, Vigo D. The granular Tabu search and its application to the vehicle-routing problem. INFORMS Journal on Computing 2003;15(4): 0333–46.

[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. 6237–42.

[30] Skrlec D, Filipec M, Krajcar S. A heuristic modification of genetic algorithm used for solving the single depot capacited vehicle routing problem. In: Proceedings of intelligent information systems, IIS '97. 1997. p. 184–8.

[31] Filipec M, Skrlec D, Slavko K. An efficient implementation of genetic algorithms for constrained vehicle routing problem. In: 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. In: 15th IEEE international conference on tools with artificial intelligence. November 3–5 2003. p. 176–83.

[33] Tong Z, Ning L, Debao S. Genetic algorithm for vehicle routing problem with time window with uncertain vehicle number. Proceedings of the fifth world congress on intelligent control and automation, June 15–19, Hangzhou, PR China; 2004. p. 2846–49.

[34] Haghani A, Jung S. A dynamic vehicle routing problem with time-dependent travel times. Computers & Operations Research 2005;32: 2959–86.

[35] Jih W, Yun-Jen J. Dynamic vehicle routing using hybrid genetic algorithms. Proceedings of the IEEE international conference on robotics & automation, Detroit, MI; 1999. p. 453–8.

[36] Osman M, Abo-Sinna M, Mousa A. An effective genetic algorithm approach to multiobjective routing problems (MORPs). Applied Mathematics and Computation 2005;163:769–81.

[37] Montemanni R, Gambardella L, Rizzoli A, Donati A. Ant colony system for a dynamic vehicle routing problem. Journal of Combinatorial Optimization 2005;10(4):327–43.

[38] Dréo J, Pétrowski A, Siarry P, Taillard E. Metaheuristics for hard optimization methods and case studies. Berlin: Springer; 2006.

[39] Dorigo M, Stützle T. Ant colony optimization. Cambridge, MA: The MIT Press; 2004.

[40] Bemporad A, Morari M. Control of systems integrating logic, dynamics and constraints. Automatica 1999;35:407–27.

[41] Ceylan H, Bell M. Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing. Transportation Research B 2004;38:329–42.

[42] Cortés CE. High coverage point-to-point transit (HCPPT): a new design concept and simulation–evaluation of operational schemes. PhD dissertation, Civil Engineering Department, University of California at Irvine. 2003.

[43] Bezdek J. Fuzzy mathematics in pattern classification. PhD thesis, Applied Mathematics Center, Cornell University, Ithaca; 1973.

[44] Babuska R. Fuzzy modeling for control. Dordrecht: 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 2002;47:863–9.

[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, MA, USA; 2004.

[47] Floudas C. Non-linear and mixed integer optimization. Oxford: Oxford University Press; 1995.

[48] Man K, Tang K, Kwong S. Genetic algorithms, concepts and designs. Berlin: Springer; 1998.