Multiobjective model predictive control for dynamic pickup and delivery problems

Alfredo Núñez [a,*], Cristián E. Cortés [b], Doris Sáez [c], Bart De Schutter [d], Michel Gendreau [e]

[a] Section of Road and Railway Engineering, Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands
[b] Civil Engineering Department, Universidad de Chile, Blanco Encalada 2002, Santiago, Chile
[c] Electrical Engineering Department, Universidad de Chile, Avenida Tupper 2007, Santiago, Chile
[d] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
[e] CIRRELT and MAGI, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montreal, Quebec, Canada

ARTICLE INFO

ABSTRACT

A multiobjective-model-based predictive control approach is proposed to solve a dynamic pickup and delivery problem in the context of a potential dial-a-ride service implementation. A dynamic objective function including two relevant dimensions, user and operator costs, is considered. Because these two components typically have opposing goals, the problem is formulated and solved using multiobjective model predictive control to provide the dispatcher with a more transparent tool for his/her decision-making process. An illustrative experiment is presented to demonstrate the potential benefits in terms of the operator cost and quality of service perceived by the users.

1. Introduction

The dynamic pickup and delivery problem (DPDP) considers a set of online requests for service for passengers traveling from an origin (pickup) to a destination (delivery) served by a fleet of vehicles initially located at several depots (Desrosiers, Soumis, & Dumas, 1986; Savelsbergh & Sol, 1995). The final output of such a problem is a set of routes for the fleet that dynamically change over time and must be determined in real time. Progress in communication and information technologies has allowed researchers to formulate such dynamic problems and to develop efficient algorithms of high computational complexity to solve these problems. The DPDP has been intensely studied in the last few decades (Cordeau & Laporte, 2007; Psaraftis, 1980, 1988) and corresponds to the embedded problem behind the operation of most dial-a-ride services. With regard to real applications, Madsen, Raven, and Rygaard (1995) adopted insertion heuristics from Jaw, Odoni, Psaraftis, and Wilson (1986) and solved a real-life problem for moving elderly and handicapped people in Copenhagen. Dial (1995) proposed a distributed system for the many-to-few dial-a-ride transit operation Autonomous Dial-a-Ride Transit (ADART), which is currently implemented in Corpus Christi,

TX, USA. A complete review of DPDPs can be found in Berbeglia, Cordeau, and Laporte (2010), where general issues and solution strategies are described. These authors conclude that it is necessary to develop additional studies on policy analysis associated with dynamic many-to-many pickup and delivery problems.

A well-defined DPDP should be based on an objective function that includes the prediction of future demands and traffic conditions in current routing decisions. Regarding dynamic routing formulations that consider the prediction of future events in real-time routing and dispatch decisions, the works of Branke, Middendorf, Noeth, and Dessourky (2005), Ichoua, Gendreau, and Potvin (2006), Mitrovic-Minic and Laporte (2004), Mitrovic-Minic, Krishnamurti, and Laporte (2004), Powell, Bouzaïene-Ayari, and Simão (2007, Chap. 5), Topaloglu and Powell (2005) can be mentioned. In previous studies (Cortés, Núñez, & Sáez, 2008; Cortés, Sáez, Núñez, & Muñoz, 2009; Sáez, Cortés, & Núñez, 2008), an analytical formulation for the DPDP as a model-based predictive control (MPC) problem using state-space models was proposed. In the previously mentioned MCP schemes, the dynamic feature of the problem appears as the system to be controlled considers future requests that are not known in advance; instead, the availability of historical information is assumed, from which future scenarios with certain probabilities of occurrence are created. Different authors treat the dynamism of the routing decisions in DPDPs differently; most of the methods in the literature are developed to be problem dependent (Ichoua et al.,

* Corresponding author. Tel.: +31 15 2789355; fax: +31 15 2783443.
E-mail address: a.a.nunezvicencio@tudelft.nl (A. Núñez).

2006; Powell et al., 2007, Chap. 5). However, in those studies, the trade-off between users' levels of service and the associated additional operational costs was completely unknown to the dispatcher. Moreover, issues regarding users' levels of service, such as delayed users (experiencing long travel or waiting times), were not considered.

In real-life implementations of DPDPs, the quality of service is critical. Paquette, Cordeau, and Laporte (2009) concluded that most dial-a-ride studies are focused on the minimization of operational costs and those additional studies on user policies must be performed. It is then reasonable that the objective function properly quantifies the impact on the users' levels of service as affected by real-time routing decisions and the effect on the associated additional operational costs. These two dimensions represent opposite objectives. On the one hand, the users want to obtain good service, implying more direct trips, which results in lower vehicle occupancy rates and higher operational costs to satisfy the same demand for a fixed fleet. More efficient routing policies from the operator's standpoint will reflect higher occupation rates, longer routes, and thus longer waiting and travel times for users. Thus, both components in the objective function must be properly balanced to make appropriate planning and fleet-dispatching decisions. The method of achieving such a balance has not yet been clarified in the literature; it depends on who makes the decisions and in what context. In this work, to guide the decision maker, the use of a multiobjective-MPC (MO-MPC) approach to solve the pickup and delivery problem is proposed. Whenever a request appears, a set of Pareto-optimal solutions are presented to the dispatcher, who must express his/her preferences (criteria in a progressive way manner (interactively), seeking the best compromise solution from the dynamic Pareto set. The final performance of the system will be related to the dispatcher and the criterion used to select the re-routing decisions. Because a set of Pareto-optimal solutions is available, the dispatcher will have additional flexibility to change the criterion on-line based on new, different circumstances, including the impact of the communications (Dotoli, Fanti, Mangini, Stecco, & Ukovich, 2010), driver behavior (Ma & Jansson, 2013), and traffic predictions using insufficient data (Chang, Chueh, & Yang, 2011), among many other real-life situations, and to select the Pareto solution that better addresses those new conditions.

Multiobjective optimization has been applied to a large number of static problems. Farina, Deb, and Amato (2004) presented several dynamic multiobjective problems found in the literature, noting the lack of methods that allow for adequate testing. The use of multiobjective optimization is not new in vehicle routing problems (VRPs; Garcia-Najera & Bullinaria, 2011; Osman, Abo-Sinna, & Mousa 2005; Paquete & Stützle, 2009). For a static VRP, Yang, Mathur, and Ballou (2000) also realized the different goals pursued by users and operators regarding their costs. Tan, Cheong, and Goh (2007) considered a multiobjective stochastic VRP with limited capacity; the authors proposed an evolutionary algorithm that incorporates two local search heuristics to determine a near-optimal solution using a fitness function. The authors demonstrated that the algorithm is capable of finding useful trade-offs and robust solutions. For a comprehensive review of multiobjective VRPs, the interested reader is referred to Jozefowiez, Semet, and Talbi (2008), who classified the different problems according to their objectives and the multiobjective algorithm used to solve them. Most of the multiobjective applications in VRPs in the literature are evaluated in static scenarios; therefore, one of the aims of this paper is to contribute to the analysis of using multiobjective optimization in dynamic and stochastic environments. In a dynamic context, multiobjective optimization can be applied in the framework of multiobjective optimal control. Many examples using multiobjective optimization in control have appeared in various fields, such as the parameter tuning of PID controllers, assignment of eigenvalues by the multiobjective optimization of feedback matrices, robust control, supervisory control, fault tolerant control, multiloop control systems, and within the framework of MPC (Gambier & Badreddin, 2007; Gambier & Jipp, 2011). For the case of multiobjective optimization in MPC, the methods can be classified into two groups.

- The most common methods are those based on (a priori) transformations into scalar objective. Those methods are overly rigid in the sense that changes in the preference of the decision maker cannot be easily considered. Among those methods, some formulations based on prioritizations (Hu, Zhu, Lei, Platt, & Dorrell, 2013; Kerrigan, Bemporad, Mignone, Morari, & Maciejowski, 2000; Kerrigan & Maciejowski, 2003; Li, Li, Rajamani, & Wang, 2011) and some based on a goal-attainment method (Zambrano & Camacho, 2002) can be highlighted; the most often used in the literature of MPC is the weighted-sum strategy.

- The second family of solutions is based on the generation and selection of Pareto-optimal points, which enables the decision maker to obtain solutions that are never explored under a mono-objective predictive control scheme, where only one solution (either optimal or near-optimal through heuristics) is obtained. This variety of options makes routing decisions more transparent and aligned with the service provider goals. The additional information (from the Pareto-optimal set) is a crucial support for the decision maker, who seeks reasonable options for service policies for users and operators. For further details, the book by Haimes, Tarvainen, Shima, and Thadatnil (1990) describes the tools necessary to understand, explain and design complex, large-scale systems characterized by multiple decision makers, multiple non-commensurate objectives, dynamic phenomenon, and overlapping information.

In the present paper, a method of the last type described above is proposed to solve a DPDP and to implement a solution scheme for the operation of a dial-a-ride service. The MO-MPC approach together with a properly well-defined objective function allows the dispatcher to make more educated dispatch and routing decisions in a transparent manner. The multiobjective feature provides more flexibility to the dispatcher when making decisions, although the problem to be solved becomes more difficult, highlighting the fact that the generation of a set of solutions instead of only one solution, as in a mono-objective formulation, is needed. In addition to the dynamic feature, a speed distribution associated with the modeled area, which is dependent on both time and space, was included. One important contribution of the present approach is the manner in which the waiting and re-routing times were modeled; an expression based on weights that are variable and that depend on previous wait times and impatientness due to rerouting actions is proposed.

All of the aforementioned features of this formulation generate a highly non-linear problem in the objective function and in the operational constraints; the dynamic feature and uncertainty regarding future demands are reasons to address the problem through a heuristic method instead of an exact solution method to provide a solution set (pseudo Pareto front) to the dispatcher, who must make adequate, real-time routing decisions. An efficient algorithm based on genetic algorithms (GA) is proposed in this context and is validated through several simulation experiments under different dispatching criteria.

The remainder of this paper is organized as follows. In the next section, the MO-MPC approach is presented. The DPDP, including the model, objective functions and MO-MPC statement, are then discussed. Next, the simulation results are presented and analyzed. Finally, conclusions and future work are highlighted.

2. Multiobjective-model-based predictive control

2.1. Model-based predictive control

MPC involves a family of controllers whose three main objectives are (1) the use of a predictive model over a prediction horizon, (2) computation of a sequence of future control actions through the optimization of an objective function, considering operation constraints and the desired behavior of the system, and (3) use of the receding horizon strategy, i.e., the first action in the obtained control sequence is applied to the system, and then, the optimization process is repeated in the next sampling instant (Camacho & Bordons, 1999). Consider, for example, the process modeled by the following non-linear, discrete-time system:

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

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the input vector, and $k \in \mathbb{N}$ denotes the time step. For this process, the following MPC problem is solved:

$$\min_{U_k} \lambda^T \cdot J(U_k, x_k)$$

subject to
$$x(k+\ell+1) = f(x(k+\ell), u(k+\ell)), \quad \ell = 0, 1, ..., N-1$$
$$x(k) = x_k,$$
$$x(k+\ell) \in \mathbf{X}, \quad \ell = 1, 2, ..., N$$
$$u(k+\ell) \in \mathbf{U}, \quad \ell = 0, 1, ..., N-1, \tag{2}$$

where $U_k = [u(k)^T, ..., u(k+N-1)^T]^T$ is the sequence of future control actions, $J(U_k, x_k) = [J_1(U_k, x_k), ..., J_M(U_k, x_k)]^T$ are the M objective functions to be minimized, $\lambda = [\lambda^1, ..., \lambda^M]^T$ is the weighting factor vector, N is the prediction horizon, and $x(k+\ell)$ is the ℓ-steps-ahead predicted state from the initial state x_k. The state and inputs are constrained to $\mathbf{X}$ and $\mathbf{U}$, respectively. The objective functions in MPC could be conflicting, i.e., a solution that optimizes one objective may not optimize the others. Next, to consider the trade-off between those opposite objectives, the multiobjective MPC framework is presented in the next section.

2.2. Multiobjective-model-based predictive control

MO-MPC is a generalization of MPC, where instead of minimizing a single-objective function, additional performance indices are considered (Bemporad & Muñoz, 2009; Gambier, 2008; Wojsznis, Mehta, Wojsznis, Thiele, & Blevins, 2007). In MO-MPC, if the process that is modeled by (1) has conflicts, i.e., a solution that optimizes one objective may not optimize others, the following multiobjective problem is solved:

$$\min_{U_k} J(U_k, x_k)$$

subject to
$$x(k+\ell+1) = f(x(k+\ell), u(k+\ell)), \quad \ell = 0, 1, ..., N-1$$
$$x(k) = x_k,$$
$$x(k+\ell) \in \mathbf{X}, \quad \ell = 1, 2, ..., N$$
$$u(k+\ell) \in \mathbf{U}, \quad \ell = 0, 1, ..., N-1, \tag{3}$$

where $U_k = [u^T(k), ..., u^T(k+N-1)]^T$ is the sequence of future control actions, $J(U_k, x_k) = [J_1(U_k, x_k), ..., J_M(U_k, x_k)]^T$ is a vector-valued function with the M objectives to be minimized, N is the prediction horizon, and $x(k+\ell)$ is the ℓ-steps-ahead predicted state from the initial state x_k. The state and inputs are constrained to $\mathbf{X}$ and $\mathbf{U}$, respectively. The solution of the MO-MPC problem is a set of control action sequences called the Pareto-optimal set.

Next, Pareto optimality is defined. Consider a feasible control sequence $U_k^P = [u_P^T(k), ..., u_P^T(k+N-1)]^T$. The sequence U_k^P is said to be Pareto optimal if and only if there does not exist another

feasible control action sequence U_k such that

(1) $J_i(U_k, x_k) \leq J_i(U_k^P, x_k)$, for $i = 1, ..., M$.

(2) $J_i(U_k, x_k) < J_i(U_k^P, x_k)$, for at least one $i \in \{1, ..., M\}$.

The Pareto-optimal set P_S contains all Pareto-optimal solutions. The set of all objective function values corresponding to the Pareto-optimal solutions is known as the Pareto front $P_F = \{[J_1(U_k, x_k), ..., J_M(U_k, x_k)]^T : U_k \in P_S\}$. If the manipulated variable is discrete and the feasible input set is finite, the size of P_S is also finite.

Among the algorithms used to solve these problems, conventional methods based on decomposition and weighting can be considered (Haimes et al., 1990). In addition, there is currently an increased interest in evolutionary multiobjective optimization algorithms, and many researchers are working on more efficient algorithms (for example Durillo et al. (2010), to mention one recent study that includes a systematic comparison of different methods).

From the set of the optimal control solutions, only the first component $u(k)$ of one of those solutions must be applied to the system; therefore, at every instant, the controller (the dispatcher in the context of a dial-a-ride system) must use a criterion to determine the control sequence that best suits the current objectives. In this paper, that decision is obtained after the Pareto-optimal set is determined. It is not possible to then choose some weighting factor a priori and subsequently solve a single-objective optimization problem. The idea is to provide the dispatcher with a more transparent tool for the decision-making process (see Fig. 1).

In the context of a dial-a-ride system, the MO-MPC is dynamic, meaning that decisions related to a service policy are made as the system progresses; for example, the dispatcher could minimize the operational costs J_2, keeping a minimum acceptable level of service for users (through J_1) when deciding on a vehicle-user assignment. Nevertheless, this tool could be implemented as a reference to support the dispatcher's decision and offers the flexibility of deciding which criterion is more adequate. MO-MPC is well suited for this type of problem because it helps the dispatcher to choose a solution to be applied considering the trade-off between Pareto-optimal solutions.

Once the Pareto front MO-MPC is found, there are many ways to choose one solution from the Pareto-optimal set (Gambier, 2008; Marler & Arora, 2004). In this paper, the criteria-based weighted average and that based on the ε-constraint method are
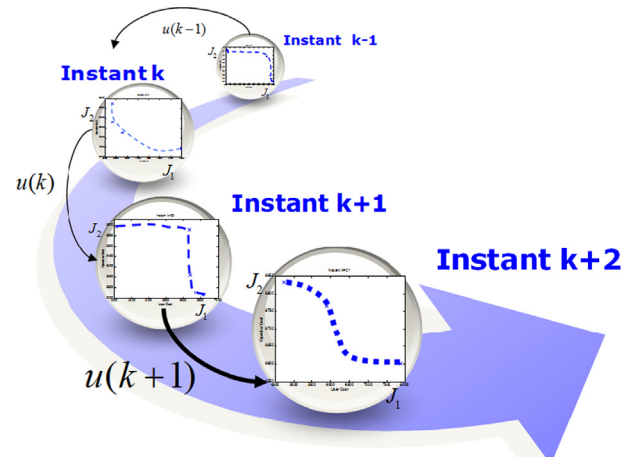

Fig. 1. In MO-MPC, at every instant, a Pareto front is shown to the dispatcher, who decides the control action to be applied.

used (Exadaktylos & Taylor, 2010; Haimes et al., 1990). In the next section, the implementation of MO-MPC used to control a dial-a-ride system is presented.

3. Dynamic pickup and delivery problem

3.1. Process description

Dial-a-ride systems are transit services that provide a share-ride, door-to-door service with flexible routes and schedules. The quality of service of a dial-a-ride service is supposed to be in between that of transit and that of taxis. The typical specifications are the users' pickup and delivery destinations and the desired pickup or delivery times. Assume that all requests are known only after the dispatcher receives the associated call and that all users want to be served as soon as possible. Thus, even if explicitly hard time windows are not included, to provide a good service, a user-oriented objective function is proposed to address the problem of undesired assignments to clients while keeping the service as regular (stable) as possible.

The service demand η_k comprises the information of the request and is characterized by two positions, pickup p_k and delivery d_k; the instant of the call occurrence t_k; a label r_k, which identifies the passenger who is calling; and the number of passengers waiting there Ω_k. The expected minimum arrival time tr_k is the best possible time to serve the passenger considering a straight journey from the origin to the destination (similar to a taxi service) and considering a waiting time obtained with the closest available vehicle (in terms of capacity) to pick up that passenger.

Assume a fixed and known fleet size F over an urban area A. The specific characteristics of a request are known only after the associated call is received by the dispatcher. A selected vehicle is then rerouted to insert the new request into the predefined route of the vehicle while the vehicle remains in motion. The assignment of the vehicle and the insertion position of the new request into the previous sequence of tasks associated with such a vehicle are control actions that are dynamically decided by the dispatcher (controller) based on multiple objective functions, which depend on the variables related to the state of the vehicles.

The modeling approach uses discrete time; the steps are activated when a relevant event k occurs, that is, when the dispatcher receives a call asking for service. Next, at any event k, each vehicle j is assigned to complete a sequence of tasks, including several pickup and delivery points. Only one of those vehicles will serve the last new request. The set of sequences is given by $u(k) = S(k) = [S_1(k)^T, ..., S_j(k)^T, ..., S_F(k)^T]^T$ and corresponds to the control (manipulated) variable, where the sequence of stops assigned to vehicle j at instant k is given by $S_j(k) = \begin{bmatrix} s_j^0(k) & s_j^1(k) & \cdots & s_j^i(k) & \cdots & s_j^{w_j(k)}(k) \end{bmatrix}^T$. The vector $s_j^i(k)$ contains the information about the ith stop of vehicle j along its route, and $w_j(k)$ is the number of planned stops. A stop is defined by either a pickup or delivery location. The initial condition $s_j^0(k)$ corresponds to the last point visited by the vehicle. In particular, the sequence of stops assigned to vehicle j at instant k, $S_j(k)$, is given by

$$S_j(k) = \begin{bmatrix} s_j^0(k) \\ s_j^1(k) \\ \vdots \\ s_j^{w_j(k)}(k) \end{bmatrix} = \begin{bmatrix} r_j^0(k) & P_j^0(k) & z_j^0(k) & \Omega_j^0(k) \\ r_j^1(k) & P_j^1(k) & z_j^1(k) & \Omega_j^1(k) \\ \vdots & \vdots & \vdots & \vdots \\ r_j^{w_j(k)}(k) & P_j^{w_j(k)}(k) & z_j^{w_j(k)}(k) & \Omega_j^{w_j(k)}(k) \end{bmatrix}, \quad (4)$$

where $r_j^i(k)$ identifies the passenger who is making the call (label), $P_j^i(k)$ is the geographic position in spatial coordinates of stop i assigned to vehicle j, $z_j^i(k)$ equals 1 if the stop i is a pickup and

equals 0 if it is a delivery, and $\Omega_j^i(k)$ is the number of passengers associated with request $r_j^i(k)$. The vehicle follows the sequence in order until completing the list of tasks assigned. The optimization procedure considers all of the necessary constraints, such as first assigning the pickup and later the delivery for a specific set of passengers in the same vehicle without violating its capacity. These constraints can be written as logical conditions as follows:

Constraint 1. Constraint of precedence. The delivery of a passenger cannot occur before his/her pickup. If a sequence contains the same label twice, then the first task is the pickup and the second is the delivery. Thus, if $r_j^{i_1}(k) = r_j^{i_2}(k)$, then $z_j^{i_1}(k) = 1$ and $z_j^{i_2}(k) = 0$. If a sequence contains only one given label, then the task is to deliver the passenger. Thus, if $\forall i_2 \leq w_j(k), i_2 \neq i_1, r_j^{i_1}(k) \neq r_j^{i_2}(k)$, then $z_j^{i_1}(k) = 0$. Therefore, the final node of every sequence will be a delivery. In brief, $z_j^{w_j(k)}(k) = 0, \quad \forall j : 1, ..., F$.

Constraint 2. Each customer has only one pickup (origin) and only one delivery (destination). In this formulation, no transfer points are allowed. Therefore, the pickup or delivery location $p_j^i(k)$ for each customer will be visited only once.

Constraint 3. Consistency. Once a group of passengers boards a specific vehicle, they must be delivered to the destination by the same vehicle (no transfers are considered in this scheme).

Constraint 4. Capacity load constraint. A vehicle will not be able to carry more passengers than its maximum load, which is $L_j^i(k) \leq L_{max}$.

Constraint 5. No swapping constraint. The order of the tasks in the sequence obtained in the previous time step $k-1$ will be kept in the next instants. Therefore, for a new request, a good pickup and delivery pair within the previous sequence is calculated to make the optimization problem more tractable by reducing the solution space.

Fig. 2 presents an example of a sequence. Users labeled as "$r_1 = 1$", "$r_2 = 2$" and "$r_3 = 3$" are assigned to vehicle j. The sequence assigned considers to pick up user "1" (coordinate 1^+), pick up user "3" (coordinate 3^+), then to delivery user "1" (coordinate 1^-) and so on. In the figure, users "1" and "3" will experience longer travel times due to rerouting. A different situation occurs with user "2", whose pickup occurs just before delivery. However, the sequence could be improved for user "2" if the first stop of the vehicle sequence is the pickup of user "2" and their subsequent delivery. The controller must then decide which sequence is better to maintain a desired user policy and a minimum operational cost.

In this work, a base-modeling approach by Cortés et al. (2008) is considered, where two sources of stochasticity are included: the first regarding the unknown future demand entering the system in real-time and the second coming from the network traffic conditions. The traffic conditions are modeled using a commercial distribution of speeds associated with the vehicles. This distribution considers two dimensions: spatial and temporal. The distribution of speeds is assumed to be unknown (denoted by $v(t, p, \varphi(t))$) and depends on a stochastic source $\varphi(t)$ (representing the traffic conditions of the network), the time and the current
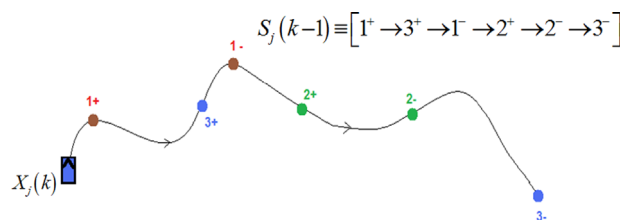
$$S_j(k-1) \equiv \begin{bmatrix} 1^+ \rightarrow 3^+ \rightarrow 1^- \rightarrow 2^+ \rightarrow 2^- \rightarrow 3^- \end{bmatrix}$$

Fig. 2. Representation of a sequence of vehicle j and its stops.

position p. Moreover, a conceptual network is assumed, where the trajectories are defined as a collection of straight lines that join two consecutive stops. In addition, a speed distribution represented by a speed model $\hat{v}(t, p)$ for the urban zone is assumed to be known, which could be obtained from historical speed data. The premise is that $\hat{v}(t, p)$ is a good approximation of the reality of the scenario in aggregate terms because this function depends on both the time and current vehicle position.

Regarding the future demand, trip patterns are extracted from historical data using a fuzzy clustering zoning method (see Section 3.3). To apply the MO-MPC approach, in the next section, a synthesis of the dynamic model of Cortés et al. (2008) used to represent the routing process is presented.

3.2. Process model

For vehicle j, the state space variables are the position $\hat{X}_j(k)$, estimated departure time vector $\hat{T}_j(k)$ and estimated vehicle load vector $\hat{L}_j(k)$.

Then, $x_j(k) = [\hat{X}_j(k)^T, \hat{T}_j(k)^T, \hat{L}_j(k)^T]^T$, and $x(k) = [x_1(k)^T, ..., x_F(k)^T]^T$. Let us denote $\hat{T}_j^i(k)$ as the expected departure time of vehicle j from stop i and $\hat{L}_j^i(k)$ as the expected load of vehicle j when leaving stop i. The dynamic model for the position of vehicle j is as follows:

$$
\hat{X}_j(k+1) = \begin{cases} \sum\limits_{i=0}^{w_j(k)-1} H_j^i(t_k+\tau)\left(P_j^i(k) + \int\limits_{s=\hat{T}_j^i(k)}^{t_k+\tau} \hat{v}(s, p(s))\dfrac{(P_j^{i+1}(k) - P_j^i(k))}{||P_j^{i+1}(k) - P_j^i(k)||_2}ds\right) & \text{if} \quad \hat{T}_j^{w_j(k)}(k) > t_k+\tau \\ P_j^{w_j(k)}(k) & \text{if} \quad \hat{T}_j^{w_j(k)}(k) \le t_k+\tau \end{cases}
$$

$$
H_j^i(t) = \begin{cases} 1 & \text{if} \quad \hat{T}_j^i(k) < t \le \hat{T}_j^{i+1}(k) \\ 0 & \text{otherwise} \end{cases}
\tag{5}
$$

In expression (5), the parameter τ is defined as the time between the occurrence of the future probable call at instant $t_k+\tau$ and the occurrence of the previous call at t_k and can be tuned by a sensitivity analysis, as described by Cortés et al. (2009). The expected stop visited by the vehicle before instant $t_k+\tau$ is i^*, and it was visited at instant $\hat{T}_j^{i^*}(k)$. The stop $P_j^0(k)$ denotes the position of the vehicle at instant k.

The direction of movement is explicitly considered when stating and computing the state space variables. In particular, the position at any time $k+1$ is computed following expression (5); the new position depends on how the speed distributes in the direction determined by the segment between the two consecutive stops i and $i+1$ along vehicle's j route (i.e., vector $P_j^{i+1}(k) - P_j^i(k)$). As further research, the plan to extend this model to a real network representation, where the speed distribution will be computed at the arc level, although the authors of this paper believe that the current approach is able to reasonably represent many real situations.

In the model, if the vehicle reaches its last stop $w_j(k)$ and no additional tasks are scheduled for that vehicle, the vehicle will stay at that stop until a new request is assigned to it. In the simulation, the vehicle will proceed in the direction to the closest zone with a low availability of vehicles and a high probability of having a pickup-request.

The predicted departure-time vector depends on the speed and can be described by the following dynamic model

$$
\hat{T}_j^i(k+1) = \begin{cases} T_j^0(k) & i = 0 \\ t_k + \sum\limits_{s=1}^{i} \kappa_j^s(k) & i \ne 0 \end{cases}, \quad i = 0, 1, ..., w_j(k),
$$

$$
\kappa_j^s(k) = \int_{P_j^{s-1}(k)}^{P_j^s(k)} \frac{1}{\hat{v}(t_j(\omega), \omega)}d\omega
\tag{6}
$$

where $\kappa_j^i(k)$ is an estimation of the time interval between stops $i-1$ and i for the sequence associated with vehicle j at instant k.

The dynamic model that is associated with the vehicle load vector depends exclusively on the current sequence and its previous load. Analytically,

$$
\hat{L}_j^i(k+1) = \begin{cases} \min\{\bar{L}_j, L_j^0(k)\} & i = 0 \\ \min\left\{\bar{L}_j, L_j^0(k) + \sum\limits_{s=1}^{i}(2z_j^s(k) - 1)\Omega_j^s(k)\right\} & i = 1, ..., w_j(k) \end{cases}
\tag{7}
$$

where $z_j^s(k)$ and $\Omega_j^s(k)$ are defined in (3) and $\bar{L}_j$ is the capacity of vehicle j. A homogeneous fleet of small vehicles with a capacity of four passengers will be considered below.

Fig. 3 presents another sequence assigned to vehicle j at instant k, which corresponds to the tasks assigned to a vehicle.

3.3. Objective function for the dial-a-ride system

The purpose of this study is to provide the dispatcher with an efficient tool that captures the tradeoff between user and operator costs. The objective function is designed to address the fact that some users can become particularly annoyed if their service is postponed (either pickup or delivery). For example, Fig. 4 demonstrates the type of situation that could arise if such issues are not considered. In the figure, the sequences at time instant 45.22 [min] of four different vehicles that are in service are shown. The sequences start with a small circle representing the current location of the vehicle, and the numbers in the squares represent the coordinate of the requests: "$+k$" is the pickup location of the user who called at instant k, whereas "$-k$" represents the delivery point of the same user. In the figure, user "4" in vehicle V1, "22" in vehicle V5, "3" in vehicle V11, and "14" in vehicle V14 have experienced a considerable delay in their service. In a typical dynamic setting, those passengers will be annoyed; passengers "4", "3" and "14" are onboard the vehicle for a long time before reaching their destination, whereas user "22" is still waiting to be picked up at the origin. Thus, the level of service is not balanced as the number of re-routings that users "4", "3" and "14" have suffered through is higher than that of other users whose requests were received several minutes after (for example, user "82" requested service approximately 40 min later than user "4" but will be served completely before "4" will be served). In a proper formulation, a higher cost in the objective function is considered
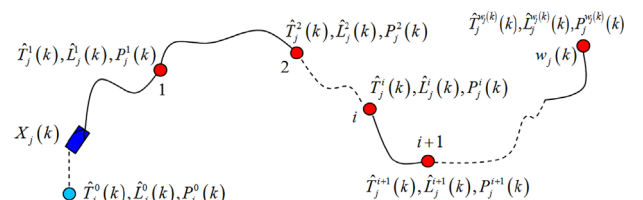

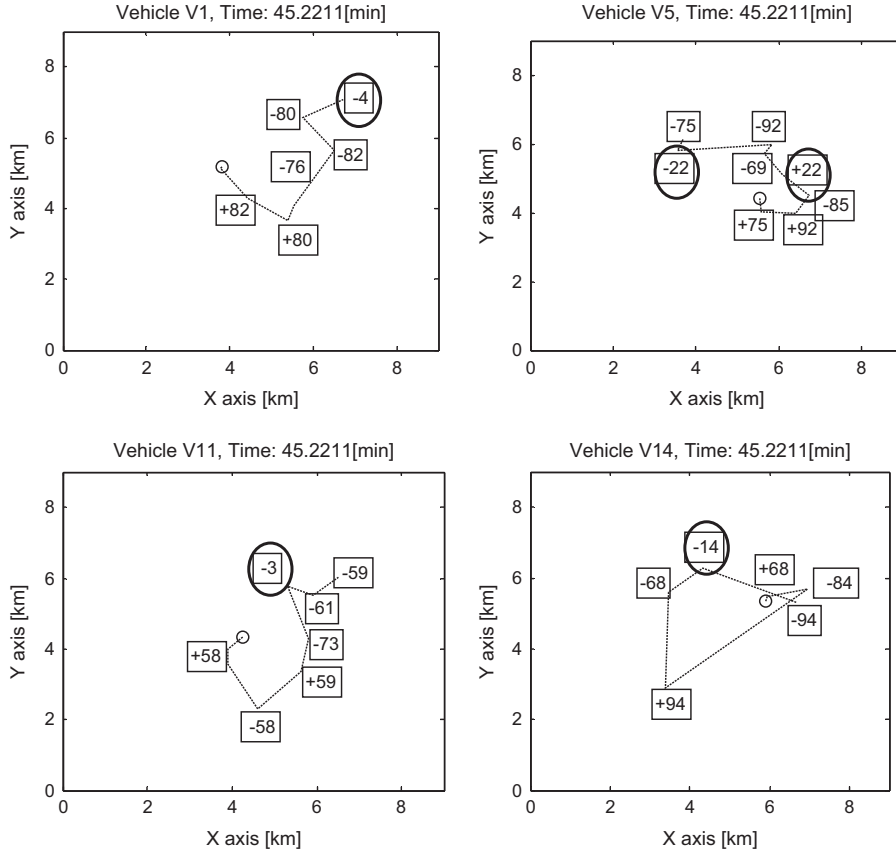**Fig. 3.** Representation of the sequence of vehicle j and its stops.

Fig. 4. Routes showing postponed users.

to penalize extremely long waiting or travel times in a different manner. Next, these ideas are formalized through an analytical expression.

The optimization variables are the current sequence $S(k)$, which incorporates the new request η_k and h_{max} future sequences $S^h = \{S(k+1)|_h, \ldots, S(k+N)|_h\}$, $h = 1, \ldots, h_{max}$, incorporating the prediction of future requests (scenarios). Thus, $U_k = \left\{ S(k), S^1, \ldots, S^{h_{max}} \right\}$ comprises all of the control actions to be calculated. The scenario h consists of the sequential occurrence of $N-1$ estimated future requests $\hat{\eta}_{k+1}^h, \hat{\eta}_{k+2}^h, \ldots, \hat{\eta}_{k+N-1}^h$ with probability p_h. This is in addition to the actual currently received request. The scenarios are obtained using historical data. This formulation can be viewed as a robust controller, where different scenarios of the uncertainty are tested to incorporate the effects of the unknown demand in the current decisions. Each scenario h can be viewed as one realization of the uncertainty in the demand, as shown in Fig. 5. In the MPC literature, there are many ways to describe uncertainty and noise, and different techniques have been proposed to achieve a robust performance (Bertsimas & Sim, 2004), constraint handling, and stability (Limón, 2002). In this paper, a fuzzy clustering method is used, as in Sáez et al. (2008). A reasonable prediction horizon N, which depends on the intensity of unknown events that enter the system and on the quality of the prediction model, is then defined. If the prediction horizon is greater than one, the controller adds the future behavior of the system into the current decision.

The proposed objective functions quantify the costs over the system of accepting the insertion of a new request. Such functions typically move in opposite directions. The first objective function (J_1), which considers the costs of the users, includes both waiting and travel times experienced by each passenger. The second objective function (J_2) is associated with the operational cost of running the vehicles of the fleet. Analytically, the proposed
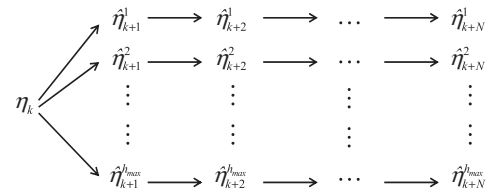

Fig. 5. Different realizations of the uncertainty in the demands.

objective functions for a prediction horizon N can be written as follows:

$$J_1 = \sum_{\ell=1}^{N} \sum_{j=1}^{F} \sum_{h=1}^{h_{max}} p_h \cdot (J_{j,h}^U(k+\ell) - J_{j,h}^U(k+\ell-1))$$

$$J_2 = \sum_{\ell=1}^{N} \sum_{j=1}^{F} \sum_{h=1}^{h_{max}} p_h \cdot (J_{j,h}^O(k+\ell) - J_{j,h}^O(k+\ell-1)), \tag{8}$$

where

$$J_{j,h}^O(k+\ell) = c_T(\hat{T}_j^{w_{j,h}(k+\ell)}(k+\ell) - \hat{T}_j^0(k+\ell))|_h + c_L \sum_{i=1}^{w_{j,h}(k+\ell)} (D_j^i(k+\ell))|_h \tag{9}$$

$$J_{j,h}^U(k+\ell) = \theta_v \sum_{i=1}^{w_{j,h}(k+\ell)} \left(f_v(r_j^i(k+\ell))(1-z_j^i(k+\ell)) \left(\underbrace{\hat{T}_j^i(k+\ell) - tr_{r_j^i(k+\ell)}}_{\text{re-routing time}} \right) \right) \Bigg|_h \cdot$$

$$+ \theta_e \sum_{i=1}^{w_{j,h}(k+\ell)} \left(f_e(r_j^i(k+\ell))z_j^i(k+\ell) \left(\underbrace{\hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)}}_{\text{waiting time}} \right) \right) \Bigg|_h \tag{10}$$

The performance of the vehicle routing scheme will depend on how well the objective function can predict the impact of possible rerouting due to insertions caused by unknown service requests. In (8), $J_{j,h}^U$ and $J_{j,h}^O$ denote the user and operator costs, respectively, associated with the sequence of stops that vehicle j must follow at a certain instant. In Eqs. (8)–(10), $k+\ell$ is the instant at which the ℓth request enters the system, measured from instant k. The number of possible call scenarios is h_{max}, and p_h is the probability of occurrence of the hth scenario. Expressions (9) and (10) represent the operator and user cost functions, respectively, related to vehicle j at instant $k+\ell$, which depend on the previous control actions and the potential request h, occurring with probability p_h, and $w_{j,h}(k+\ell)$ is the number of stops estimated for vehicle j at instant $k+\ell$ in scenario h. To explain the flexibility of the formulation and its economic consistency, the term related to the additional time experienced by passengers in this service (delivery time minus the minimum time for the user to arrive at their destination) considers a cost θ_v for each minute, and the term related to the total waiting time considers a cost θ_e for each minute. The terms in the objective functions for users are weighted by the functions f_v and f_e, which include a service policy for the users; therefore, the cost of a user that entered the system a long time ago is considered more important than that of another user who recently made a request. The following weighing functions are proposed:

Fig. 6. Closed-loop diagram of the MO-MPC for the dynamic dial-a-ride problem.

decisions based on the information of the system (process) and the values of the fleet attributes, which enables the evaluation of the model under different scenarios. The service demand η_k and traffic conditions $\phi(t, p)$ are disturbances in this system.

The following multiobjective problem is solved:

$$\operatorname*{Min}_{U_k} \ \{J_1, J_2\}$$
$$\text{s.t. operational constraints},$$
$$(13)$$

where J_1 and J_2 correspond to the objective functions defined in (8). The solution to this problem corresponds to a set of control sequences, which form the Pareto-optimal set. U_k^i is a feasible control action sequence in the sense that it satisfies all of the operational constraints. From the Pareto front solutions for the dynamic MO-MPC problem, it is necessary to select only one

$$f_v(r_j^i(k+\ell)) =$$
$$\begin{cases} 1 & \text{if} \quad \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} < \alpha(tr_{r_j^i(k+\ell)} - t_{0r_j^i(k+\ell)}) \\ 1 + \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} - \alpha(tr_{r_j^i(k+\ell)} - t_{0r_j^i(k+\ell)}) & \text{if} \quad \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} < \alpha(tr_{r_j^i(k+\ell)} - t_{0r_j^i(k+\ell)}) \end{cases} . \qquad (11)$$

Expression (11) implies that if the delivery time $\hat{T}_j^i(k+\ell)$ that is associated with user $r_j^i(k+\ell)$ becomes larger than α times its minimum total time ($tr_{r_j^i(k+\ell)} - t_{0r_j^i(k+\ell)}$), the weighting function $f_v(\cdot)$ increases linearly, resulting in a critical service for such a client. Regarding the waiting time factor, the following expression is proposed:

$$f_e(r_j^i(k+\ell)) = \begin{cases} 1 & \text{if} \quad \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} \le TT \\ 1 + \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} - TT & \text{if} \quad \hat{T}_j^i(k+\ell) - t_{0r_j^i(k+\ell)} > TT \end{cases} . \qquad (12)$$

The intuition behind (12) is analogous to (11). In addition, the operational cost (9) considers a component depending on the total distance traveled, weighted by a factor c_L, and a component depending on the total operational time, weighted by a unitary cost c_T in this case. Thus, $D_j^i(k+\ell)$ represents the distance between stops $i-1$ and i in the sequence of vehicle j. The framework presented here permits the inclusion of different objective functions proposed in the literature without changing the general approach and solution algorithms.

3.4. MO-MPC for the pickup and delivery problem

A systematic method of incorporating such a trade-off existing between both objective functions is through a multiobjective approach, which results in a general set of solutions, giving the dispatcher the opportunity to change the service policies in a more transparent manner by considering the Pareto front. The closed loop of the dynamic vehicle routing system is shown in Fig. 6. The MO-MPC represented by the dispatcher makes the routing
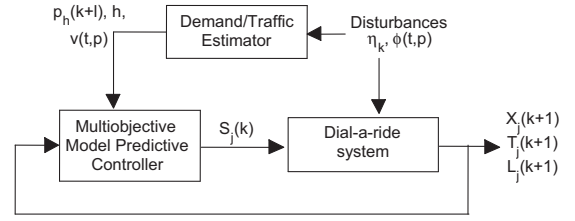
control sequence $U_k^i = \{S^i(k), S^{i,1}, ..., S^{i,h_{max}}\}$ and then apply the control action $S^i(k)$ to the system based on the receding horizon concept. A criterion related to the importance given to the user (J_1) and operator (J_2) costs in the final decision is needed to select this sequence. The solutions obtained from the multiobjective problem form a set, which includes, as a particular case, the optimal point obtained by solving the mono-objective problem. The proposed MO-MPC algorithm is divided into the following steps:

Step 0. Select a set of reasonable scenarios h and vehicle $j \in F$ candidate to serve the request k.

Step 1. The scenario h consists of the sequential occurrence of N requests $\eta_k, \hat{\eta}_{k+1}^h, \hat{\eta}_{k+2}^h, ..., \hat{\eta}_{k+N-1}^h$. For each vehicle $j \in F$ and each scenario h, solve 2^N multiobjective problems considering the cases where vehicle j is the one that serves none, one, or up to N of those requests. For example, if $N=2$, for each vehicle, solve four multiobjective problems considering the cases to serve none, η_k and $\hat{\eta}_{k+1}^h$, and η_k and $\hat{\eta}_{k+1}^h$. The multiobjective problem in this step is as follows:

$$\min_{\{S_j(k), S_j^h(k+1), ..., S_j^h(k+N-1)\}}$$
$$\times \left\{ \sum_{\ell=1}^{N} (J_{j,h}^U(k+\ell) - J_{j,h}^U(k+\ell-1)), \ \sum_{\ell=1}^{N} (J_{j,h}^O(k+\ell) - J_{j,h}^O(k+\ell-1)) \right\}$$
$$\text{s.t. operational constraints}$$

In this problem, the operational constraints are the capacity, consistency and no-swapping constraint (insertions maintaining the previous sequence); therefore, the Pareto-optimal set contains only feasible sequences. Some of these multiobjective problems are easy to solve, but the number of possible

solutions increases as the vehicle serves more requests. In fact, considering the no-swapping constraint, the number of possible solutions when request k is served by vehicle j is only $0.5\prod_{i=0}^{W-1}(w_j(k)+i)(w_j(k)+i-1)$, where $w_j(k)$ is the number of stops of vehicle j at instant k. The multiobjective problems in this step are the most time consuming, but they can be solved in parallel because they are not related to each other. The solution of this multiobjective problem is obtained using the Pareto-optimal sets through the use of the GA method described in Section 3.5.

Step 2. For a given scenario h, considering that only one vehicle can serve each request, obtain the Pareto-optimal set of the fleet in coordinated operation by solving the following multiobjective problem:

$$\min_{\{S(k),\, S^1,\, \ldots,\, S^{hmax}\}} \left\{ \sum_{j \in F} \sum_{\ell=1}^{N} (J_{j,h}^{U}(k+\ell) - J_{j,h}^{U}(k+\ell-1)), \right.$$
$$\left. \times \sum_{j \in F} \sum_{\ell=1}^{N} (J_{j,h}^{O}(k+\ell) - J_{j,h}^{O}(k+\ell-1)) \right\}.$$

The solution to this multiobjective problem is obtained using the Pareto-optimal sets through the use of the GA method described in Step 1 by combining the $|F|^N$ possible cases of cooperation between vehicles in such a way that the current request and each future request are served by only one vehicle. For example, with three vehicles $F = \{a, b, c\}$, for $N = 2$, the cases to be analyzed in this step are $|F|^N = 9$, considering that $v_1 \in F$ serves the current request η_k and $v_2 \in F$ serves the future request $\hat{\eta}_{k+1}^h$. The multiobjective problem of this step can be solved in parallel.

Step 3. Next, using the Pareto-optimal set of all scenarios h, solve the following multiobjective problem:

$$\min_{U_k} \left\{ \sum_h \sum_{j \in F} \sum_{\ell=1}^{N} p_h (J_{j,h}^{U}(k+\ell) - J_{j,h}^{U}(k+\ell-1)), \right.$$
$$\left. \times \sum_h \sum_{j \in F} \sum_{\ell=1}^{N} p_h (J_{j,h}^{O}(k+\ell) - J_{j,h}^{O}(k+\ell-1)) \right\}.$$

The solution to this multiobjective problem is obtained using the Pareto-optimal sets from Step 2 (which can be performed in parallel) by multiplying each Pareto front by the probability of occurrence of the associated scenario p_h and subsequently combining the different cases considering the different scenarios.

Step 4. A relevant step of this approach in the controller's dispatch decision is the definition of criteria to select the best control action at each instant under the MO-MPC approach. For example, once the Pareto front is found, different criteria regarding a minimum allowable level of service can be dynamically used to make policy-dependent routing decisions. In this work, the cases based on a weighted-sum and a ε-constraint criterion are used.

The algorithm retains optimality because the cooperation among vehicles is analyzed in a centralized manner. The main advantage of the algorithm is that each vehicle can be equipped with a multicore computer that calculates the Pareto fronts, where the different future scenarios could be evaluated in parallel in each core of the computer. It is possible to reduce the number of candidate vehicles in Step 0 to reduce the effect of exponential complexity growth; therefore, only those vehicles that are more likely to serve a request should be selected in Step 1.

The complexity can also limited by limiting the number of future scenarios.
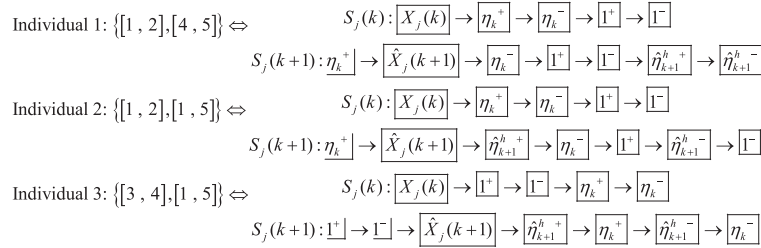
3.5. MO-MPC algorithm based on a GA for dynamic pickup and delivery

A GA is proposed to implement the MO-MPC method described in Step 1 of Section 3.4 because it can efficiently address mixed-integer, non-linear optimization problems. The main concept is to determine the Pareto-optimal set and subsequently determine the solution to be implemented as the control action. A potential solution of the GA is called an individual. The individual can be represented by a set of parameters related to the genes of a chromosome and can be described in binary or integer form.

As explained in the previous Section, 2^N multiobjective optimization problems are solved for each vehicle $j \in F$ and each scenario h. In the adapted GA that is used to solve the DPDP, the most expensive computational scenario is considered for vehicle j, assuming that it has to serve the current request η_k and the future $N-1$ requests, namely, $\hat{\eta}_{k+1}^h$, $\hat{\eta}_{k+2}^h$, ..., $\hat{\eta}_{k+N-1}^h$. The proposed vehicle sequences and state variables must satisfy the set of constraints given by the conditions of the dial-a-ride operation (precedence, capacity and consistency). The individual represents a possible control-action sequence $\{S_j(k), S_j^h(k+1), ..., S_j^h(k+N-1)\}$; the following constraints are considered when generating individuals:

(1) No swapping; therefore, a new call can be located within the previous sequence without modifying the previous order.
(2) Precedence (pickup must go before delivery).

Each of the N control actions, $S_j(k)$ and $S_j^h(k+\ell)$, $\ell = 1, ..., N-1$, is represented by two chromosomes with integer values between 1 and $w_j(k)$ for the first sequence and between 1 and $w_j(k+\ell)$ for the second sequence for $\ell = 1, ..., N-1$. The first gene corresponds to the position in the sequence of the pickup, whereas the second gene identifies the delivery point in the same manner. An individual will have a total of $2N$ genes, the first two defining the pickup and delivery points of request η_k in sequence $S_j(k)$, the next two genes including request $\hat{\eta}_{k+1}^h$ in sequence $S_j^h(k+1)$ and so on. As an example, assume that at the moment the new call occurs, the sequence of a certain vehicle j is $S_j(k-1): \boxed{X_j(k)} \rightarrow \boxed{r_{k-1}^{+}} \rightarrow \boxed{r_{k-1}^{-}}$ with $w_j(k-1) = 2$, meaning that vehicle j located at $X_j(k)$ is on its way to pick up passenger r_{k-1} at the geographical coordinate r_{k-1}^{+} and that the passenger is going to be dropped off later at delivery point r_{k-1}^{-}. In the two-steps-ahead-problem, the current request $\eta_k: \boxed{\eta_k^{+}} \rightarrow \boxed{\eta_k^{-}}$ and the next future estimated request $\hat{\eta}_{k+1}^h: \boxed{\hat{\eta}_{k+1}^{h\,+}} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,-}}$ are considered. The number of stops in the first sequence is known and is equal to $w_j(k) = 4$, noting that for the next instant (after τ s), the maximum number of stops will be $\hat{w}_j(k+1) = 6$ to generate the individuals. Once the position of the vehicle at the moment the next customer call is received is estimated, say, $\hat{X}_j(k+1)$, the number of stops visited during the τ s before the next request at $\hat{t}_{k+1} = t_k + \tau$ is calculated. When the location of the pickup and/or delivery of the estimated request $\hat{\eta}_{k+1}^h$ is within a segment of the sequence that was already visited at $\hat{t}_{k+1}$, then the pickup and delivery points are simply located at that time instant in the first place of the tasks. Next, some examples of individuals (solutions) for the described simple case are shown. Individual 1 does not need to adjust the sequence at instant $k+1$, unlike individuals 2 and 3, who must do so.

$$\text{Individual 1: } \{[1,2],[4,5]\} \Leftrightarrow$$
$$S_j(k): \boxed{X_j(k)} \rightarrow \boxed{\eta_k^+} \rightarrow \boxed{\eta_k^-} \rightarrow \boxed{1^+} \rightarrow \boxed{1^-}$$
$$S_j(k+1): \underline{\eta_k^+} \rightarrow \boxed{\hat{X}_j(k+1)} \rightarrow \boxed{\eta_k^-} \rightarrow \boxed{1^+} \rightarrow \boxed{1^-} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,+}} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,-}}$$

$$\text{Individual 2: } \{[1,2],[1,5]\} \Leftrightarrow$$
$$S_j(k): \boxed{X_j(k)} \rightarrow \boxed{\eta_k^+} \rightarrow \boxed{\eta_k^-} \rightarrow \boxed{1^+} \rightarrow \boxed{1^-}$$
$$S_j(k+1): \underline{\eta_k^+} \rightarrow \boxed{\hat{X}_j(k+1)} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,+}} \rightarrow \boxed{\eta_k^-} \rightarrow \boxed{1^+} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,-}} \rightarrow \boxed{1^-}$$

$$\text{Individual 3: } \{[3,4],[1,5]\} \Leftrightarrow$$
$$S_j(k): \boxed{X_j(k)} \rightarrow \boxed{1^+} \rightarrow \boxed{1^-} \rightarrow \boxed{\eta_k^+} \rightarrow \boxed{\eta_k^-}$$
$$S_j(k+1): \underline{1^+} \rightarrow \underline{1^-} \rightarrow \boxed{\hat{X}_j(k+1)} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,+}} \rightarrow \boxed{\eta_k^+} \rightarrow \boxed{\hat{\eta}_{k+1}^{h\,-}} \rightarrow \boxed{\eta_k^-}$$

Using genetic evolution, the chromosomes exhibiting the best fitness are selected to ensure the best offspring. The best parent genes are selected, mixed and recombined for the production of an offspring in the next generation. Two fundamental operators are used for the recombination of the genetic population: crossover and mutation. For the former, the portions of two chromosomes are exchanged with a certain probability to produce the offspring. The latter operator alters each portion randomly with a certain probability. To determine the pseudo Pareto-optimal set of MO-MPC, the best individuals are those that belong to the best pseudo Pareto-optimal set found until the current iteration (because there are solutions that belong to the pseudo Pareto-optimal set, although they have not been found yet). Solutions that belong to the best pseudo Pareto-optimal set will have a fitness function equal to a certain threshold (0.9 in this case), whereas the fitness function for all other solutions will be assigned a lower threshold (for example, 0.1) to maintain diversity in the solution. The complete GA procedure that is applied to this MO-MPC control problem is presented in Appendix A.

In the literature, a variety of evolutionary multiobjective optimization algorithms and methods to address constraints can be found (see the recent reviews of Mezura-Montes and Coello (2011) and Zhou et al. (2011)).

The GA implementation was conceived ad hoc to the specific DPDP treated in this work. The tuning parameters of the MO-MPC method based on a GA are the number of individuals, number of generations, crossover probability p_c, mutation probability p_m and stopping criteria. Because the focus is on finding the pseudo Pareto-optimal set, at each stage of the algorithm, the best individuals will be those who belong to the best Pareto-optimal set found until the current iteration. From the pseudo-optimal Pareto front, it is necessary to select only one control sequence $U_k^* = [u^*(k)^T, \ldots, u^*(k+N-1)^T]^T$, and from this sequence, the current control action $u^*(k)$ must be applied to the system according to the receding horizon concept. For the selection of this sequence, a criterion related to the importance given to both objectives J_1 and J_2 in the final decision is required. The GA approach in MO-MPC provides a sub-optimal Pareto front that is close to optimality.

In general, the selection of the algorithm is related to the application and its requirements. In the present paper, an ad hoc GA that is used to measure the benefits of the approach is proposed; however, the main contribution of this study is the MO-MPC framework for DPDPs. When using GAs, it is not possible to ensure a strict convergence criterion; however, GAs can be applied in real time when the stopping criteria includes a maximum number of iterations related to the maximum computational time available to solve the multiobjective problem (typically within a given sampling time). To ensure the applicability of GAs to the DPDP, the set of solutions from the previous time step are used as part of the population for the next instant (attaching the new request at the end of the sequence). Moreover, all of the solutions generated by the algorithm are feasible in terms of the capacity constraint, and therefore, at any instant, the dispatcher always has a reasonable solution for performing the dynamic routing under a real-time setting.

4. Simulation studies

A discrete-event simulation of a period of 2 h was conducted, which is representative of a working day (14:00–14:59, 15:00–15:59), over an urban area of approximately 81 km². A fixed fleet of 15 demand-responsive vehicles with a capacity of four passengers each is considered. Assume that the vehicles travel in a straight line between stops and that the transport network behaves based on a speed distribution with a mean of 20 [km/h].

A total of 250 calls were generated over the 2 h simulation period following the spatial and temporal distribution observed from the historical data. Regarding the temporal dimension, a negative exponential distribution for time intervals between calls at a rate of 0.5 [call/min] during both hours of the simulation was assumed. Regarding the spatial distribution, the pickup and delivery coordinates were randomly generated within each zone. The first 15 calls at the beginning and the last 15 calls at the end of the experiments were discarded from the statistics to avoid limit

Table 1
Pickup and delivery coordinates and probabilities: fuzzy zoning.

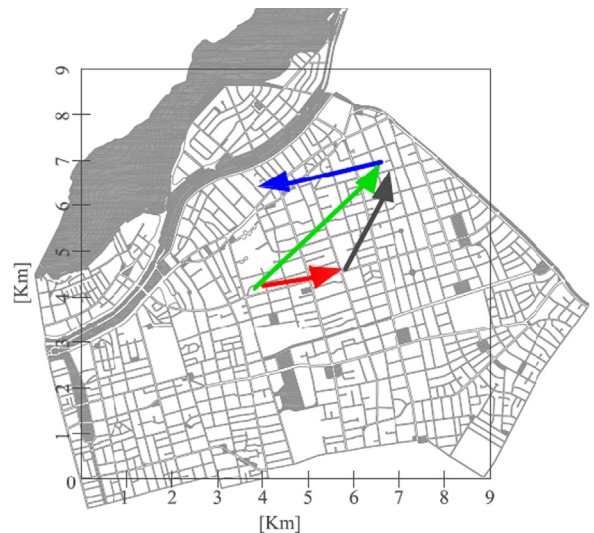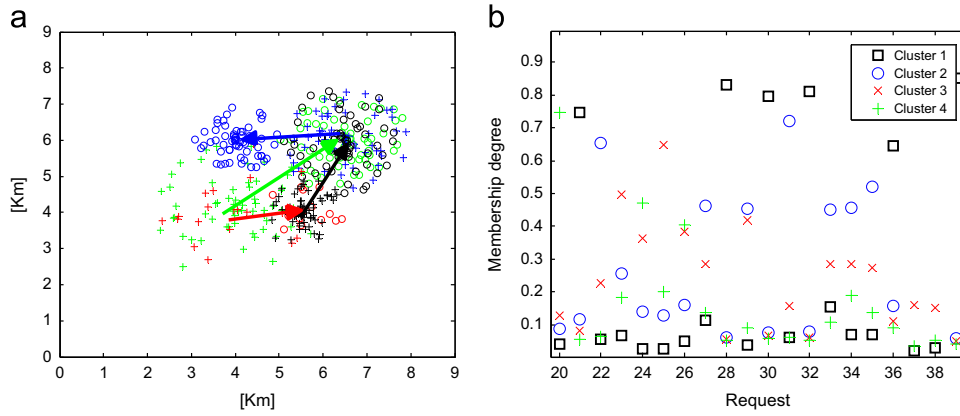| X pickup	Y pickup	X delivery	Y delivery	Probability
4.007	4.1847	5.6716	4.5576	0.119
3.9312	4.0303	6.4762	6.1463	0.1726
5.4013	4.0548	6.5659	5.9723	0.3512
6.4578	5.9338	3.9844	5.9785	0.3571

Fig. 7. Origin–destination trip patterns.

Fig. 8. (a) One realization of random demands, arriving with a negative Poisson distribution and (b) each call belongs to a different fuzzy cluster with different membership degrees.

Table 2
MO-MPC, user indices.

MO criterion		Travel time [min/pax]		Waiting time [min/pax]	
		Mean	Std	Mean	Std
1	$\lambda^T = \begin{bmatrix} 1 & 0 \end{bmatrix}$	9.36	3.66	4.52	2.74
2	$\lambda^T = \begin{bmatrix} 0.75 & 0.25 \end{bmatrix}$	9.79	4.25	4.47	2.49
3	$\lambda^T = \begin{bmatrix} 0.50 & 0.50 \end{bmatrix}$	10.19	4.49	4.60	2.99
4	$\lambda^T = \begin{bmatrix} 0.25 & 0.75 \end{bmatrix}$	10.48	4.75	5.38	3.06
5	$\lambda^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$	10.01	7.38	15.44	10.80
6	Criterion 1	9.36	3.66	4.52	2.74
7	Criterion 2a	10.32	4.75	4.62	2.67
8	Criterion 2b	10.76	5.36	5.63	3.58
9	Criterion 2c	10.63	6.09	7.25	4.59
10	Criterion 3	10.01	7.38	15.44	10.80

Table 3
MO-MPC, operator indices.

MO criterion		Time traveled [min/veh]		Distance traveled [km/veh]	
		Mean	Std	Mean	Std
1	$\lambda^T = \begin{bmatrix} 1 & 0 \end{bmatrix}$	88.16	7.55	24.84	1.86
2	$\lambda^T = \begin{bmatrix} 0.75 & 0.25 \end{bmatrix}$	75.17	11.06	20.61	2.94
3	$\lambda^T = \begin{bmatrix} 0.50 & 0.50 \end{bmatrix}$	67.57	12.78	18.62	3.51
4	$\lambda^T = \begin{bmatrix} 0.25 & 0.75 \end{bmatrix}$	61.67	12.57	16.95	3.17
5	$\lambda^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$	43.90	17.94	12.58	5.09
6	Criterion 1	88.16	7.55	24.84	1.86
7	Criterion 2a	74.99	8.76	20.91	2.19
8	Criterion 2b	69.56	11.52	19.92	3.05
9	Criterion 2c	71.40	10.53	20.39	2.80
10	Criterion 3	43.90	17.94	12.58	5.09

distortion (warm-up period). Finally, 10 replications of each experiment were carried out to obtain global statistics. Each replication consists of 2 h of simulation, during which 250 on-line decisions are made. Each replication required an average of 20 min of computation time using an Intel® Core™2 2.40 GHz processor.

The future calls are unknown for the controller. The typical trip patterns can be extracted from historical data using a systematic methodology. Cortés et al. (2008) and Sáez et al. (2008) obtained a speed distribution model along with the trip patterns, the latter being obtained through a fuzzy zoning method to define the most likely origin–destination configurations. The proposed fuzzy zoning method relies only on off-line data, without the knowledge of the parameters of the distribution used in the generation of the data in the simulation. This fuzzy zoning permits the generation of the trip patterns and their probabilities, as shown in Table 1 and Figs. 7 and 8.

The locations and times of occurrence of the service requests, together with the non-homogeneous distribution of speed over time and space, assumed in the experiments were chosen to represent a wide range of situations regarding traffic conditions and different city configurations.

The objective function is formulated by considering the following parameters: prediction horizon $N=2$, $\theta_v=16{,}7$ [Ch\$/min], $\theta_e=50$ [Ch\$/min], $c_T=25$ [Ch\$/min], $c_L=350$ [Ch\$/Km], $\alpha=1.5$, $TT=5$ [min]. Five different criteria are used for the MO-MPC-based weighted sum: $\lambda^T = \begin{bmatrix} \lambda_1 & 1-\lambda_1 \end{bmatrix}$, $\lambda_1=1$, 0.75, 0.5, 0.25 and 0 (the first five rows of Tables 2 and 3).

The results using the ε-constraint method (the last five rows of Tables 2 and 3) are associated with the following three criteria:

Criterion 1: Minimum user cost component.
Criterion 2: Nearest value to a given user cost (measured as travel plus waiting time penalties).
Criterion 3: Minimum operational cost component.

For *Criterion 2* (the nearest value to a given user cost), three references are considered: 400, 500 and 600 [Ch\$] for sub-cases (a), (b) and (c), respectively. In Table 2, the results are presented in terms of user indices: effective travel and waiting times per user (mean and *std*). In Table 3, the distance and time traveled per vehicle are presented as operator indices. Tables 2 and 3 indicate that *Criterion 1* and *Criterion 3* are equivalent to cases with $\lambda^T = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $\lambda^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$, respectively.

As shown in Tables 2 and 3, small standard deviations imply that travel and waiting times are more balanced among passengers, which is due in part to the specification of the objective function, in which functions f_v and f_e (Eqs. (11) and (12)) consider this issue by weighting the users' components (travel and waiting) differently in the objective function based on the total time that each customer has spent in the system; therefore, the dispatcher penalizes those customers that have suffered more rerouting more strongly, forcing the vehicle to drop them off at their destinations sooner in the sequence list. This effect creates a fair distribution of customers in terms of travel and waiting times, which is reflected in the small *std* indicators, as discussed above.

In Fig. 9, the mean user and operator costs are depicted for the weighted-sum method. The extreme case $\lambda^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$ in favor of the operator results in a poor level of service for users around the mean and in terms of bounding the standard deviation (Tables 2 and 3). In general, the intermediate values of λ (between
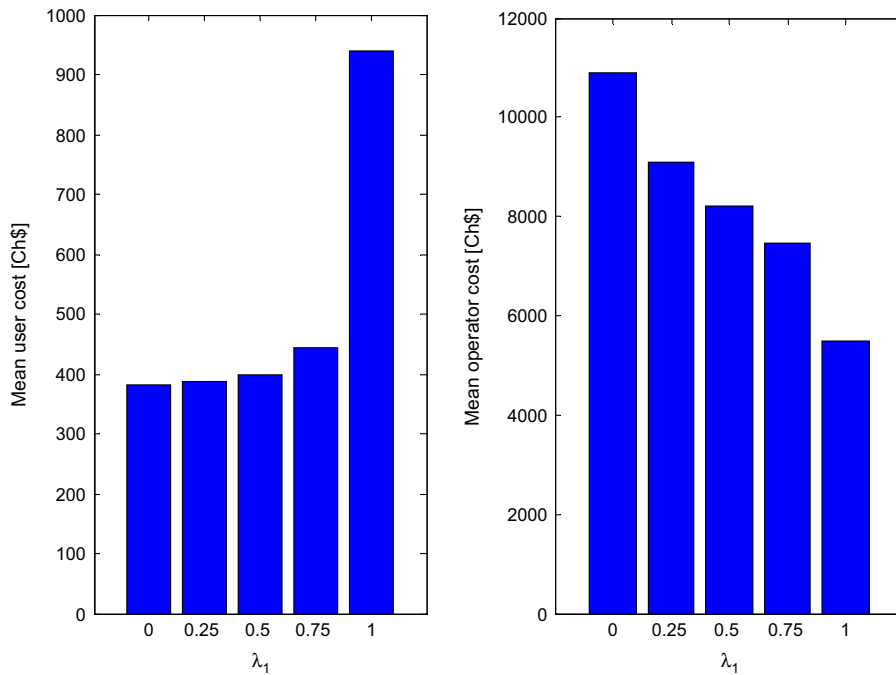
Fig. 9. Mean user and operator costs, multiobjective criterion $\lambda^T = \begin{bmatrix} 1-\lambda_1 & \lambda_1 \end{bmatrix}$.
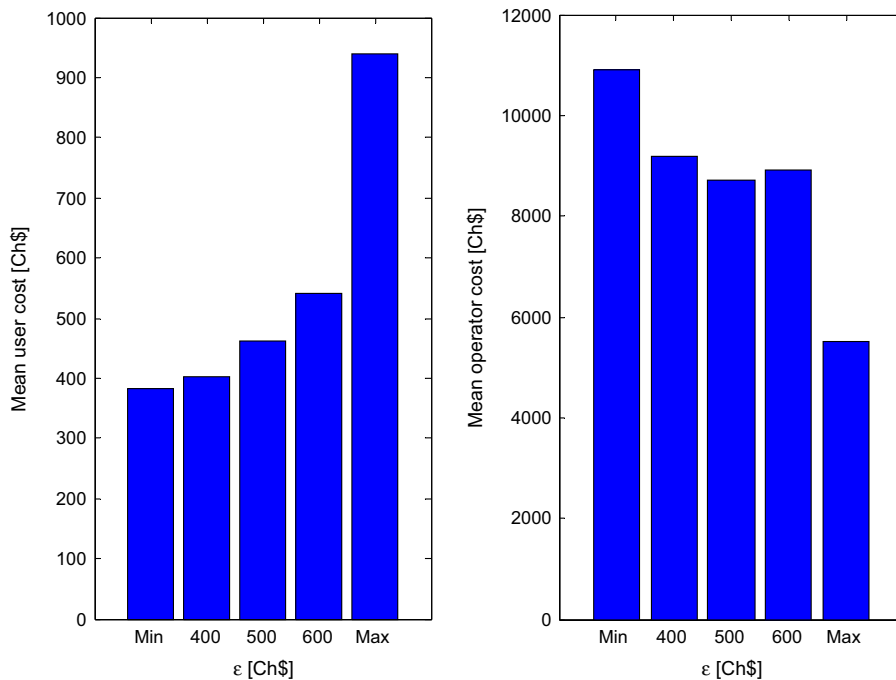

Fig. 10. Mean user and operator costs, ε-constraint criterion.

extreme cases) exhibit an increasing (decreasing) tendency, as expected, in the figure with respect to λ_1, where $\lambda^T = [1-\lambda_1, \lambda_1]$. A larger value of λ_1 indicates a higher mean user cost, whereas a smaller value of λ_1 indicates a lower mean operational cost (Fig. 9). A large value of λ_1 means that a strong penalty is associated with operator cost and a small penalty is associated with user cost; therefore, those policies will be advantageous for the operators. The opposite argument is analogous for the case of small values of λ_1. An interesting case observed in Fig. 9 is the drastic change (effect) on user costs when λ_1 increases from 0.75 to 1.00 (approximately double in the latter case for the weighted-sum method). This sensitivity demonstrates the importance of

maintaining policies around more favorable scenarios (upon initial inspection, in the ranges between 0.2 and 0.7) for users with a small impact on operators.

With regard to the ε-constraint method (Tables 1 and 2), an expected tendency can be observed, whereby *Criteria 1* and *3* clearly favor users and operators, respectively; the benefits for users when applying *Criterion 1* concern the waiting time, as the travel time remains almost invariant. The case of *Criterion 2* for the three thresholds defined above when applying the ε-constraint method graphically (see Fig. 10) can be highlighted, displaying more stability in the range 400–600 and exhibiting the same abrupt jump, as presented in Fig. 9, in terms of mean operator cost
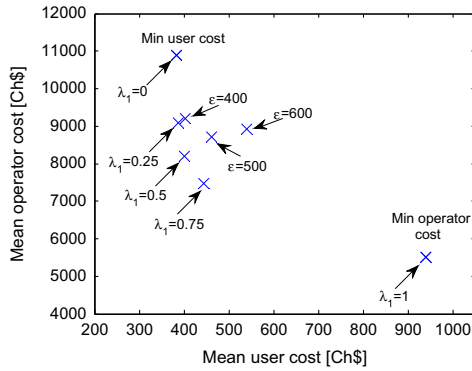
Fig. 11. Trade-offs in the overall performance.

with respect to the extreme case $\lambda^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$. Therefore, regardless of the methodology used, values of the weight (epsilon) close to extreme cases could become dangerous in terms of offering poor services for users when the operator is favored beyond what is necessary. The case of *Criterion 2* is also interesting in terms of the relation of the resulting mean user cost over the entire simulation, which fits the thresholds defined for each sub-case quite well.

Fig. 11 presents the trade-off in the overall performance (averaged over the entire simulation). As can be easily observed from the global performance in Fig. 11, the ε-constraint method failed to obtain the Pareto-dominant solution (in terms of overall performance). One reason for this behavior is the fact that in the dynamic setting, at some instants, the pre-defined value of epsilon could not be reached by any feasible solution; in such cases, the controller selects a solution that does not necessarily follow the trend of the objective function.

5. Conclusions

This study presents a new approach to solve the DPDP, represented here by a dial-a-ride operational scheme, through a MPC scheme using dynamic multiobjective optimization. Different criteria are used to obtain control actions for dynamic routing using the dynamic Pareto front. The criteria enable priority to be given to a service policy for users, ensuring the minimization of operational costs under each proposed policy. The service policies are approximately verified using the average of the repeated simulations. Under the implemented *on-line* system, it is easier and more transparent for the operator to follow service policies under a multiobjective approach instead of dynamically tuning weighting parameters. The multiobjective approach determines solutions that are directly interpreted as part of the Pareto front instead of results obtained with mono-objective functions, which lack direct physical interpretation (the weight factors are tuned, but they do not allow the application of either operational or service policies, such as those proposed in this study).

This paper discusses a transportation system for a dynamic point-to-point service. In this direction, one relevant contribution of this approach is in the effort to combine (i) a DPDP that includes a speed distribution that is dependent on both time and space, (ii) a scheme that is optimized by control theory and multiobjective optimization and (iii) a novel approach to modeling waiting and re-routing times with weights that are variable and depend on previous waiting times and impatientness due to rerouting actions. The complexity of the resulting formulation reveals one drawback of the approach: obtaining the solution set of the multiobjective problem requires a significant computational effort, which can be a serious issue in the context of a real application. To

address this issue, the use of a simple genetic algorithm to solve the multiobjective optimization problem is proposed in this paper. The next step is to explore and develop better algorithms for the real-time implementation of the scheme. The decision to first test a GA as a solution method was based on the fact that a GA solution platform was already available from work on a mono-objective dynamic problem in a previous publication (Sáez et al., 2008); therefore, the multiobjective GA was conceived within such a framework. Other heuristics of the same nature, such as PSO in a mono-objective scheme, have also been tested (Cortés et al., 2009). The MO-MPC formulation for dynamic pickup and delivery is generic in the sense that the implementation of any other evolutionary algorithm (e.g., PSO, ant colony, differential evolution) leaves the solution scheme nearly unchanged. To improve the computational efficiency of the methodology for solving a real-sized case, the problem could be decoupled using a distributed MPC scheme that is also available in the specialized literature.

It is possible to determine a good, representative, pseudo Pareto-optimal set in a dynamic context using evolutionary computation and other efficient algorithms that have been developed in recent years. Comparisons with various other methods, such as normal boundary intersection (NBI), normal constraint (NC), direct search domain (DSD), successive Pareto optimization (SPO), and game theory, will be part of further research. Future work will also focus on the analysis of different multiobjective criteria.

Appendix A

The complete procedure of the GA applied to this MO-MPC control problem is as follows:

Step 1. Set the iteration counter to $i = 1$ and initialize a random population of n individuals, i.e., create n random integer feasible solutions of the manipulated variable sequence. Not all of the individuals are feasible because of the constraints

$$x(k+\ell) \in \mathbf{X}, \quad \ell = 1, 2, ..., N,$$
$$u(k+\ell) \in \mathbf{U}, \quad \ell = 0, 1, ..., N-1.$$

The size of the population is I individuals per generation.

$$\text{Population } i \Leftrightarrow \begin{pmatrix} [u^1(k), u^1(k+1), ..., u^1(k+N-1)]^T \\ \vdots \\ [u^j(k), u^j(k+1), ..., u^j(k+N-1)]^T \\ \vdots \\ [u^I(k), u^I(k+1), ..., u^I(k+N-1)]^T \end{pmatrix}$$

Step 2. For each individual, evaluate J corresponding to the defined objective functions in (3). Next, obtain the fitness function of every individual in the population. When considering individuals belonging to the best pseudo Pareto-optimal set (the Pareto-optimal set obtained with the information available until that moment), a fitness function equal to 0.9 will be set;

otherwise, 0.1 will be used to maintain the solution diversity. If the individual is not feasible, penalize it (pro-life strategy).

Step 3. Select random parents from the population i (different vectors of the future control actions).

Step 4. Generate a random number between 0 and 1. If the number is less than probability p_c, choose an integer $0 < c_p < N-1$ (c_p denotes the crossover point) and apply the crossover to the selected individuals to generate an offspring. The next scheme describes the crossover operation for two individuals U^j and U^l, resulting in U^j_{cross} and U^l_{cross}.

$$U^j = \left[\left[u^j(k), u^j(k+1), ..., u^j(k+c_p-1)\right], \left[u^j(k+c_p), ..., u^j(k+N-1)\right]\right]^T$$

$$U^l = \left[\left[u^l(k), u^l(k+1), ..., u^l(k+c_p-1)\right], \left[u^l(k+c_p), ..., u^l(k+N-1)\right]\right]^T$$

$$\Downarrow$$

$$U^j_{cross} = \left[\left[u^l(k), u^l(k+1), ..., u^l(k+c_p-1)\right], \left[u^j(k+c_p), ..., u^j(k+N-1)\right]\right]^T$$

$$U^l_{cross} = \left[\left[u^j(k), u^j(k+1), ..., u^j(k+c_p-1)\right], \left[u^l(k+c_p), ..., u^l(k+N-1)\right]\right]^T$$

Step 5. Generate a random number between 0 and 1. If the number is less than probability p_m, choose an integer $0 < c_m < N-1$ (c_m denotes the mutation point) and apply the mutation to the selected parent to generate an offspring. Select a value $u^j_{mut} \in U$ and replace the value in the c_mth position in the chromosome. The next scheme describes the mutation operation for an individual U^j, resulting in U^j_{mut}.

$$U^j = \left[u^j(k), u^j(k+1), ..., u^j(k+c_m-1), \boxed{u^j(k+c_m)}, u^j(k+c_m+1), ..., u^j(k+N-1)\right]^T$$

$$\Downarrow$$

$$U^j_{mut} = \left[u^j(k), u^j(k+1), ..., u^j(k+c_m-1), \boxed{u^j_{mut}}, u^j(k+c_m+1), ..., u^j(k+N-1)\right]^T$$

Step 6. Evaluate objective functions J_1 and J_2 for all individuals in the offspring population. Next, obtain the fitness of each individual by following the fitness definition described in step 2. If the individual is unfeasible, penalize its corresponding fitness.

Step 7. Select the best individuals according to their fitness. Replace the weakest individuals from the previous generation with the strongest individuals of the new generation.

Step 8. If the tolerance given by the maximum generation number is reached (stopping criteria, i equals the number of generations), then stop; otherwise, go to step 3. Because the focus is on a real-time control strategy, the best stopping algorithm criterion corresponds to the number of generations (therefore, the computational time can be bounded).

References

Bemporad, A., & Muñoz de la Peña, D. (2009). Multiobjective model predictive control. *Automatica, 45*(12), 2823–2830.

Berbeglia, G., Cordeau, J., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research, 202*(1), 8–15.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53.

Branke, J., Middendorf, M., Noeth, G., & Dessourky, M. (2005). Waiting strategies for dynamic vehicle routing. *Transportation Science, 39*, 298–312.

Camacho, E. F., & Bordons, C. (1999). *Model predictive control.* London, U.K.: Springer-Verlag.

Chang, T. H., Chueh, C. H., & Yang, L. K. (2011). Dynamic traffic prediction for insufficient data roadways via automatic control theories. *Control Engineering Practice, 19*(12), 1479–1489.

Cordeau, J., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operation Research, 153*, 29–46.

Cortés, C. E., Núñez, A., & Sáez, D. (2008). Hybrid adaptive predictive control for a dynamic pickup and delivery problem including traffic congestion. *International Journal of Adaptive Control and Signal, 22*(2), 103–123.

Cortés, C. E., Sáez, D., Núñez, A., & Muñoz, D. (2009). Hybrid adaptive predictive control for a dynamic pickup and delivery problem. *Transportation Science, 43*(1), 27–42.

Desrosiers, J., Soumis, F., & Dumas, Y. (1986). 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.

Dial, R. (1995). Autonomous Dial a Ride transit – Introductory overview. *Transportation Research – Part C, 3*, 261–275.

Dotoli, M., Fanti, M. P., Mangini, A. M., Stecco, G., & Ukovich, W. (2010). The impact of ICT on intermodal transportation systems: A modelling approach by Petri nets. *Control Engineering Practice, 18*(8), 893–903.

Durillo, J., Nebro, A., Coello, C., García-Nieto, J., Luna, F., & Alba, E. (2010). A study of multiobjective metaheuristics when solving parameter scalable problems. *IEEE Transactions on Evolutionary Computation, 14*(4), 618–635.

Exadaktylos, V., & Taylor, C. J. (2010). Multi-objective performance optimisation for model predictive control by goal attainment. *International Journal of Control, 83*(7), 1374–1386.

Farina, M., Deb, K., & Amato, P. (2004). Dynamic multiobjective optimization problems: Test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation, 8*(5), 425–439.

Gambier, A. (2008). MPC and PID control based on multi-objective optimization. In *Proceedings of the American Control Conference* (pp. 4727–4732). Seattle, WA, USA.

Gambier, A., & Badreddin, E. (2007). Multi-objective optimal control: An overview. In *Proceedings of the IEEE international Conference on Control Applications, CCA 2007* (pp. 170–175). Singapur.

Gambier, A., Jipp, M. (2011). Multi-objective optimal control: An introduction. In *Proceedings of the 2011 8th Asian Control Conference (ASCC)* (pp. 1084–1089). Kaohsiung, Taiwan.

Garcia-Najera, A., & Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 38*(1), 287–300.

Haimes, Y., Tarvainen, K., Shima, T., & Thadatnil, J. (1990). *Hierarchical multiobjective analysis of large-scale systems.* New York, USA: Hemisphere Publishing Corp.

Hu, J., Zhu, J., Lei, G., Platt, G., & Dorrell, D. G. (2013). Multi-Objective model-predictive control for high-power converters. *IEEE Transactions on Energy Conversion, 28*(3), 652–663.

Ichoua, S., Gendreau, M., & Potvin, J. Y. (2006). Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science, 40*(2), 211–225.

Jaw, J., Odoni, A., Psaraftis, H., & Wilson, N. (1986). A heuristic algorithm for the multivehicle many-to-many advance-request dial-a-ride problem. *Transportation Research B, 20*, 243–257.

Jozefowiez, N., Semet, F., & Talbi, E. (2008). Multiobjective vehicle routing problems. *European Journal of Operational Research, 189*(2), 293–309.

Kerrigan, E. C., Bemporad, A., Mignone, D., Morari, M., & Maciejowski, J. M. (2000). Multi-objective priorisation and reconfiguration for the control of constrained hybrid systems. In *Proceedings of the American Control Conference* (pp. 1694–1698). Chicago, Illinois, USA.

Kerrigan, E. C., & Maciejowski, J. M. (2003). Designing model predictive controllers with prioritised constraints and objectives. In *Proceedings of the 2002 IEEE international symposium on computer aided control system design* (pp. 33–38). Glasgow, Scotland, UK.

Li, S., Li, K., Rajamani, R., & Wang, J. (2011). Model predictive multi-objective vehicular adaptive cruise control. *IEEE Transactions on Control Systems Technology, 19*(3), 556–566.

Limón, D. (2002). *Control predictivo de sistemas no lineales con restricciones: estabilidad y robustez* (Ph.D. thesis). Spain: Universidad de Sevilla.

Ma, X., & Jansson, M. (2013). A model identification scheme for driver-following dynamics in road traffic. *Control Engineering Practice, 21*(6), 807–817.

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*, 193–208.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization, 26*, 369–395.

Mezura-Montes, E., & Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation, 1*(4), 173–194.

Mitrovic-Minic, S., Krishnamurti, R., & Laporte, G. (2004). Double-horizon based heuristic for the dynamic pickup and delivery problem with time windows. *Transportation Research B, 38*, 669–685.

Mitrovic-Minic, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B, 38*, 635–655.

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*(2), 769–781.

Paquete, L., & Stützle, T. (2009). Design and analysis of stochastic local search for the multiobjective traveling salesman problem. *Computers & Operations Research, 36*(9), 2619–2631.

Paquette, J., Cordeau, J., & Laporte, G. (2009). Quality of service in dial-a-ride operations. *Computers & Industrial Engineering, 56*(4), 1721–1734.

Powell, W., Bouzaïene-Ayari, B., & Simão, H. (2007). Dynamic models for freight transportation. In: Cynthia Barnhart, & Gilbert Laporte (Eds.), *Handbooks in operations research and management science*, Vol. 14 (pp. 285–365). Amsterdam, The Netherlands: Elsevier.

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

Psaraftis, H. (1988). Dynamic vehicle routing problems. In B. L. Golden & A. A. Assad (Eds.). *Vehicle routing methods and studies* (pp. 223–248).

Sáez, D., Cortés, C. E., & Núñez, A. (2008). Hybrid adaptive predictive control for the multi-vehicle dynamic pickup and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research*, *35*(11), 3412–3438.

Savelsbergh, M., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17–29.

Tan, K. C., Cheong, C. Y., & Goh, C. K. (2007). Solving multiobjective vehicle routing problem with demand via evolutionary computation. *European Journal of operational research*, *177*, 813–839.

Topaloglu, H., & Powell, W. (2005). A distributed decision-making structure for dynamic resource allocation using non linear functional approximations. *Operations Research*, *53*(2), 281–297.

Wojsznis, W., Mehta, A., Wojsznis, P., Thiele, D., & Blevins, T. (2007). Multi-objective optimization for model predictive control. *ISA Transactions*, *46*(3), 351–361.

Yang, W. H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science*, *34*, 99–112.

Zambrano, D., & Camacho E.F. (2002). Application of MPC with multiple objectives for a solar refrigeration plant. In *Proceedings of the 2002 international conference on control applications* (pp. 1230–1235). *Vol. 2*, Cancun, Mexico.

Zhou, A., Qu, B. Y., Li, H., Zhao, S., Suganthan, P. N., & Zhang, Q. (2011). Multi-objective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*(1), 32–49.