

Multiairport Capacity Management: Genetic Algorithm With Receding Horizon

Xiao-Bing Hu, Wen-Hua Chen, *Member, IEEE*, and Ezequiel Di Paolo

Abstract—The inability of airport capacity to meet the growing air traffic demand is a major cause of congestion and costly delays. Airport capacity management (ACM) in a dynamic environment is crucial for the optimal operation of an airport. This paper reports on a novel method to attack this dynamic problem by integrating the concept of receding horizon control (RHC) into a genetic algorithm (GA). A mathematical model is set up for the dynamic ACM problem in a multiairport system where flights can be redirected between airports. A GA is then designed from an RHC point of view. Special attention is paid on how to choose those parameters related to the receding horizon and terminal penalty. A simulation study shows that the new RHC-based GA proposed in this paper is effective and efficient to solve the ACM problem in a dynamic multiairport environment.

Index Terms—Air traffic control, airport capacity management (ACM), genetic algorithm (GA), receding horizon control (RHC), terminal penalty.

I. INTRODUCTION

AT BUSY airports, severe congestion during peak periods, i.e., when air traffic demand exceeds available capacity, becomes the everyday reality around the world, particularly in the United States and Western and Central Europe [1]–[3]. The projected growth of the traffic demand will make the situation worse in the near future if no action is undertaken for capacity improvements. The role of airport capacity management (ACM) becomes especially significant. Many efforts have been made in studying methods and tools in order to provide an accurate and reliable prediction of airport capacity and air traffic demand for strategic traffic management programs [2], [4]–[8]. These models and methods can be used by traffic managers and controllers as an automated support tool for decision making on traffic flow and capacity management at airports during periods of congestion. In particular, for a given time period, runway configuration, weather forecast, and predicted arrival and departure demands for runways and fixes (input data), theoretically, one can determine an optimal strategy for managing arrival/departure traffic at an airport (output), i.e., the maximum number of flights can be accepted (arrivals)

and released (departures) during congested periods at the airport, how many flights are to be delayed, and how long.

Gilbo [2] reported a method for the optimization of airport capacity by dynamic allocation of the capacity over time between arrivals and departures. In general, the optimal solution provides time-varying capacity profiles that most effectively solve a predicted congestion problem by reflecting the dynamics of the traffic demand and the operational conditions at the airport. This approach was further extended in [9] to a much more complex airport system where the runways and arrival and departure fixes were considered jointly. The traffic flow through the airport system is optimized by taking into account the interaction between runway capacity and capacities of fixes. The work in [2] and [9] mainly focused on how to develop an effective optimizing algorithm for a static ACM problem, where no system uncertainties were considered. Chen and Hu [10] introduced the receding horizon control (RHC) strategy to the method reported in [2] to attack a dynamic ACM problem, where uncertainties in the predicted traffic demands and operational conditions at the airport were present. The RHC-related parameters and potential advantages were investigated in depth, and the RHC-based method proved to be more successful to optimize online the airport capacity profile in a dynamic environment. However, in these three papers, the modeling of the ACM problem was highly simplified such that the capacity optimization could be formulated as a linear programming problem, which could be easily and efficiently resolved. Recently, research attention has been moving to air traffic management in multiairport systems. Models and algorithms developed based on single airport systems have been extended to more complicated multiairport cases [11]–[13]. For example, the cutting plane algorithm and integer programming were applied to multi-ACM based on an open network model in [11]; dynamic programming and network topology were used for multiairport traffic flow coordination in [12]; and methods to attack the multiairport ground holding problem were particularly studied in [13].

This paper attempts to develop a new genetic algorithm (GA) based on the RHC strategy to attack the dynamic ACM problem in a multiairport system (MACM), where flights can be redirected between airports in order to minimize delay. Redirecting flights between airports makes the ACM problem more complicated, and therefore, it is difficult to apply deterministic optimization methods like linear programming to solve the problem effectively. As is well known, a GA is a large-scale parallel stochastic searching and optimizing algorithm, and it is effective for solving a wide range of complex optimization problems [14], [15]. The conventional way to apply GAs to

Manuscript received May 30, 2006; revised September 12, 2006, November 18, 2006, and November 22, 2006. The work of X.-B. Hu was supported by an ORS Award. The work of E. Di Paolo was supported by EPSRC under Grant EP/C51632X/1. The Associate Editor for this paper was M. Sokoloski.

X.-B. Hu and E. Di Paolo are with the Department of Informatics, University of Sussex, BN1 9QH Brighton, U.K. (e-mail: Xiaobing.Hu@sussex.ac.uk; ezequiel@sussex.ac.uk).

W.-H. Chen is with the Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU Leicestershire, U.K. (e-mail: W.Chen@lboro.ac.uk).

Digital Object Identifier 10.1109/TITS.2006.890067

the dynamic case of the ACM problem is to simply use a GA to repeat optimizing capacity profile for the rest of the operating day. Since it is so time consuming, the GA used in this conventional way can hardly meet the demand of real-time properties in practice. On the other hand, in a dynamic environment, there is always some unreliable information in the predicted traffic demands and/or the operational conditions. For example, some flights may be canceled while some unanticipated aircraft may ask for emergency landing, and the weather is also to an extent unpredictable. Therefore, the conventional way of optimizing capacity profile for the rest of the operating day will not necessarily result in actually optimal or even suboptimal solutions. To overcome these drawbacks of a conventional GA, we introduce the concept of RHC to GAs for the dynamic ACM problem. Simply speaking, RHC is an N -step-ahead online optimization strategy. At each time interval, based on current available information, RHC optimizes the particular problem for the next N intervals in the near future, but only the solution part corresponding to current interval is implemented. Clearly, since the RHC strategy only optimizes, in each time interval, the capacity profile over an N -step-long horizon other than the rest of the operating day, it can effectively improve the real-time properties of GA and reduce the influence of unreliable information. However, the introduction of RHC could also make the new GA short sighted. To guarantee the solution quality of the RHC-based GA, like [16], which reported such an algorithm for aircraft arrival sequencing and scheduling problem, some RHC-related parameters, such as time interval, length of receding horizon, and terminal penalty are investigated in depth.

The remainder of this paper is organized as follows: The basic idea of RHC is briefly explained in Section II, the model of MACM problem is given in Section III, and the novel RHC-based GA for the MACM problem is proposed in Section IV. Section V reports some interesting simulation results. The paper ends with some conclusions in Section VI.

II. BASIC IDEA OF RHC

RHC has proved to be a very effective online optimization strategy in the area of control engineering and is very successful when compared with other control strategies [17]. It is easy for RHC to handle complex dynamic systems with various constraints. It also naturally exhibits promising robust performance against uncertainties since the online updated information can be sufficiently used to improve the decision. Within this framework, decisions are made by looking ahead for N steps in terms of a given objective function, and only the decision for the first step is actually implemented. Then, the implementation result is checked, and a new decision is made by taking into account updated information and looking ahead for another N steps. RHC has now been widely accepted in the area of control engineering. Recently, attention has been paid to applications of RHC to areas like management and operations research. For example, theoretical research work on how to apply model predictive control (another name for RHC) to a certain class of discrete-event systems was presented in [18] and [19], and many practical implementations of RHC in

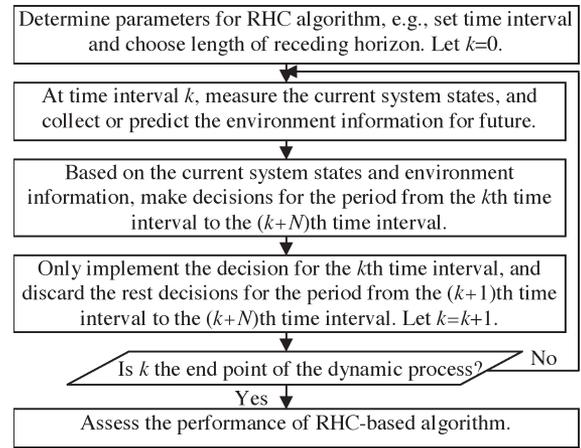


Fig. 1. Flow chart of RHC.

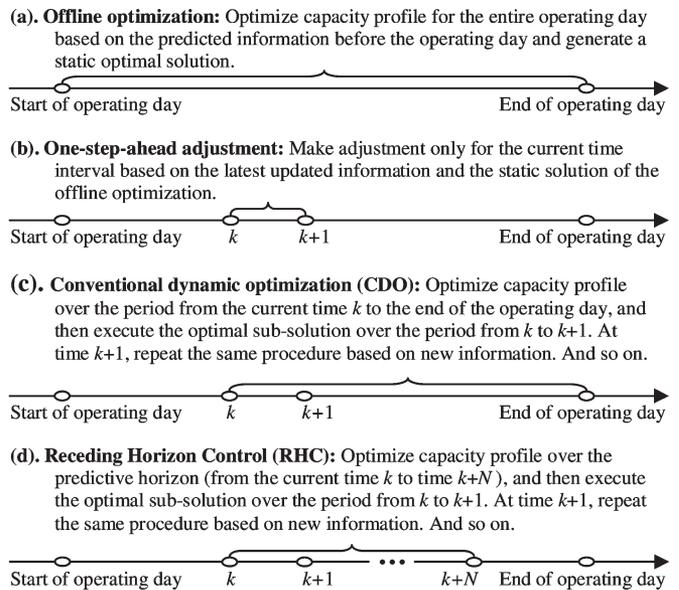


Fig. 2. Some optimization strategies.

the area of commercial planning and marketing were reported in [20]. However, as mentioned in [21], research on applying RHC to areas other than control engineering is just starting.

The basic idea of RHC for dynamic optimization problems is illustrated by the flow chart given in Fig. 1. Fig. 2 compares the RHC strategy with some other conventional optimization strategies in an intuitive way. Due to system uncertainties, the offline optimization strategy, as shown in Fig. 2(a), is not suitable for dynamic optimization processes. However, most algorithms in the literature on the ACM problem are mainly tested by using the offline strategy: the so-called static version; e.g., see [2] and [9]. The one-step-ahead (OSA) adjustment strategy in Fig. 2(b) is often used in the real practice of ACM due to its simplicity. OSA adjustment can be considered as a special case of RHC, i.e., the length of the receding horizon is $N = 1$. This special RHC is always criticized for being short sighted. The conventional dynamic optimization (CDO) strategy in Fig. 2(c) is another straightforward way to handle the dynamic ACM problem. This strategy often suffers from heavy online computational burden, and its

performance is relatively too sensitive to uncertainties on the current predicted information. The RHC strategy in Fig. 2(d) has good potential to retain/minimize the merits/demerits of both OSA and CDO.

However, integrating the RHC strategy into a GA to develop an effective and practicable method to solve the dynamic ACM problem in a multi-airport system requires much more than simply using any kind of GA as the online optimizer in the RHC scheme. To make them work in harmony, in the first place, the GA-based online optimizer should be designed from a dynamic point of view—more precisely speaking, from an RHC point of view. Therefore, unlike other literature, we do not have a so-called static version of algorithms in this paper but directly design our RHC-based GA for solving the ACM problem in a dynamic environment. As will be explained in depth later, some RHC-related parameters, particularly terminal penalty, which is widely used by the RHC in control engineering, are adopted to design the online GA-based optimizer.

III. FORMULATION OF MACM PROBLEM

A. Basic Concepts

A multi-airport system comprises one (or more) main airport and some adjacent satellite airports. The main airport mainly serves international passenger and cargo flights in and out of the area, while the satellite airports focus on most national flights and also serve as bases for certain airlines. Usually, these satellite airports also have facilities to serve international flights, and they have some regularly scheduled international flights, most of which are cargo flights. Due to the congestion at the main airport, it happens quite often that some arrival flights (either national or international), which are originally scheduled to land at the main airport, have to be redirected to one of these satellite airports. Sometimes, due to bad weather or other unexpected poor operational conditions, as well as congestion at a certain satellite airport, some or all of its scheduled arrival flights have to be redirected to other satellite airports but usually not to the main airport. If such an operation of rearranging arrival flights is inevitable, cargo flights should always be the first under consideration, then national passenger flights, and then international passenger flights in turn. This operation of dynamically rearranging arrival flights between airports can effectively improve the traffic volume in the area. A simplified scheme of a three-airport system that reflects the arrival–departure processes in the system is shown in Fig. 3.

Each airport comprises some arrival fixes, departure fixes, and a runway system. There are separate sets of arrival and departure fixes located in the near-terminal airspace area (50–70 km off the airport) so that the arrival fixes serve only arrival flow, and the departure fixes serve only departure flow. The runway system on the ground serves both arrival and departure flows.

The arrival flights are assigned to special arrival fixes, and before landing, they should pass the fixes. After leaving runways, the arrival flights follow the taxiways to the gates at the terminal. The departure flights, after leaving the gates, are

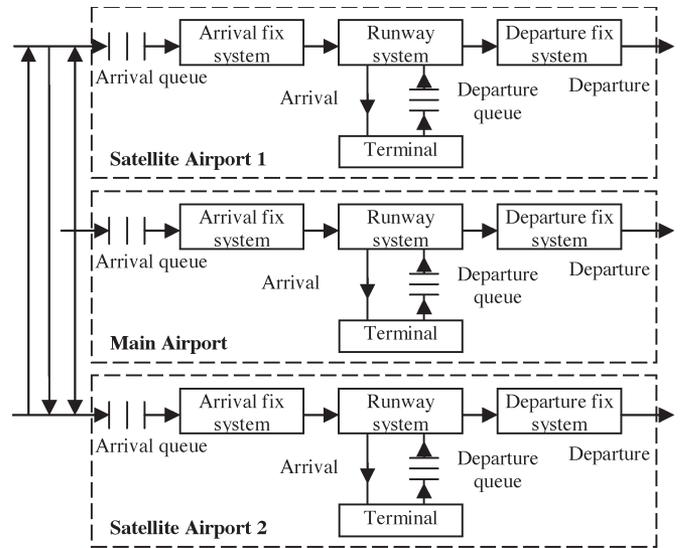


Fig. 3. Arrival–departure scheme of a three-airport system.

headed for the runways and, after leaving runways, go through the departure fixes. The departing flights are also assigned to the special fixes.

The arrival queues are formed before the fixes (see Fig. 3). This means that the flights that pass through the fixes must be accepted at the runways. If there is an arrival queue, a certain number of flights should be delayed. Some of them are to be delayed in the air and some of them on the ground at the departure airports. Those delayed in the air could either wait to land at its destination airport or be redirected to another airport. The departure queue is formed before the runway system, and flights can be delayed either at their gates or on the taxiway.

The arrival and departure fixes have constant capacities (service rate), which show the maximum number of flights that can cross the fix in a certain interval. In this paper, except where it is explicitly indicated, the time interval for capacity allocation is 15 min long, and therefore, the traffic demand for the airport is given by the predicted number of arriving and departing flights per each 15-min interval in the operating day. These capacities determine the operational constraints in the near-terminal airspace. The operational limits on the ground (runways) are characterized by arrival capacity and departure capacity. These capacities are generally variable and interdependent.

Basically, the runway system is the bottleneck resource of the airport. One reason is that the total capacity at all arrival/departure fixes is usually larger than the possible maximum arrival/departure capacity on the runways. Another reason is that arrival/departure flights can be reassigned to other arrival/departure fixes if the previously assigned fix is saturated. Like in [10], for the sake of simplicity, only capacities on the runways are considered in this paper.

The optimal allocation of arrival capacity and departure capacity on the runways is crucial to air traffic flow management. That is, if a large value is set for arrival capacity, more departure flights have to be delayed; otherwise, more arrival flights have to wait in the air. There are a number of major airports with

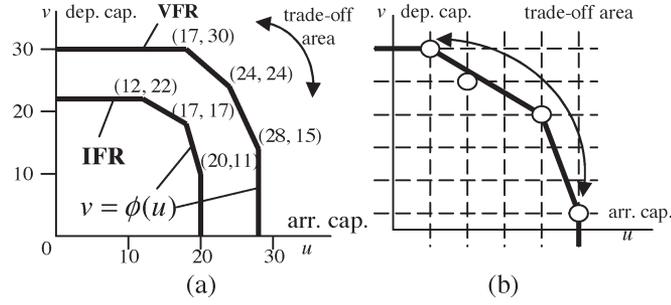


Fig. 4. Airport arrival–departure capacity curves (15 min).

runway configurations that practice the tradeoff between arrival and departure capacities. For these configurations, the arrival capacity u and the departure capacity v are interdependent and can be represented by a functional relationship $v = \Phi(u)$. Generally, given a time interval, the function is a piecewise linear convex one. The graphical representation of the function on the “arrival capacity–departure capacity” plane is called the airport capacity curve [2], [7]. Fig. 4 illustrates a 15-min capacity curve with the tradeoff area. The representation of airport runway capacity through the capacity curves is a key factor in the model.

Besides runway configurations, weather conditions also have a significant influence on the arrival and departure capacities at the airport. Weather conditions are clustered into four operational weather categories that reflect conventional limitations on visibility and ceiling, namely 1) visible flight rules (VFR), 2) marginal VFR, 3) instrument flight rules (IFR), and 4) low IFR. Capacity curves vary for these four different weather categories. For the sake of simplicity, only two weather conditions, i.e., VFR and IFR, are considered in this paper. Fig. 4(a) gives an example of the airport capacity curves for VFR and IFR operational conditions, where the IFR capacities are approximately 30% less than VFR capacities.

In a multiairport system, each airport may have quite different capacity curves, depending on its infrastructure and facilities. The ACM problem in a multiairport system aims, based on the capacity curves of airports and the operation of rearranging arrival flights between airports, to dynamically allocate capacity over time between arrivals and departures at each airport such that the arrival demand in the area and the departure demand at each airport are optimally met in terms of a specified objective function. The choice of the objective function is an important step in formulating the problem. The effectiveness of arrival and departure operations in the system can be measured by the total delay time of the flights being served (i.e., the total waiting time in the arrival and departure queues) or by the total number of flights in the queue during the operating day. These two measures both reflect the physical essence of the problem and are strongly correlated; larger queues mean longer delays. Which of the measures to use in the objective function depends on factors such as the type of input data available and the simplicity of obtaining the optimal solutions. For the same reason given in [2], the total number of flights in the queues has been chosen to construct the objective function for optimal capacity management.

B. Modeling of a Multiairport System

A constrained state-space-based model is given as follows to describe the dynamics of the airport capacity system, i.e., the functional relationship between the input data (airport capacities, traffic rearrangement, and predicted information) and the output (arrival and departure queues)

$$\begin{cases} x_i(k+1) = \max \left(0, x_i(k) + a_i(k) - u_i(k) + \sum_{j \neq i, j=1}^n z_{j,i}(k) \right. \\ \quad \left. - \sum_{j \neq i, j=1}^n z_{i,j}(k) \right) \\ y_i(k+1) = \max(0, y_i(k) + d_i(k) - v_i(k)) \\ x_i(0) = x_{i,0}, \quad i = 1, \dots, n, \quad j = 1, \dots, n \\ y_i(0) = y_{i,0} \end{cases} \quad (1)$$

subject to constraints

$$0 \leq v_i(k) \leq \phi_i(k, u_i(k)), \quad \phi_i(k, u_i(k)) \in \Phi_i \quad (2)$$

$$z_{i,j}(k) \leq x_i(k) + a_i(k)$$

$$i = 1, \dots, n, \quad j = 1, \dots, n, \quad i \neq j \quad (3)$$

$$z_{j,m}(k) = 0 \text{ and } z_{m,i}(k) = 0, \text{ if } z_{i,j}(k) > 0$$

$$i, j, m = 1, \dots, n, \quad i \neq j, \quad i \neq m \quad (4)$$

where k is the discrete time index, n is the number of airports in the system, $x_i(k)$ and $y_i(k)$ are, respectively, the arrival queue and departure queue at the i th airport by the beginning of the k th time interval, $a_i(k)$ and $d_i(k)$ are, respectively, the demand measured as the number of flights per time interval for arrivals and for departures at the i th airport at the k th time interval, $u_i(k)$ and $v_i(k)$ are, respectively, the arrival capacity and departure capacity at the i th airport at the k th time interval, $z_{i,j}(k)$ is the number of flights redirected from the i th airport to the j th airport at the k th time interval, $\phi_i(k, u_i(k))$ is the i th airport’s arrival/departure capacity curve function that depends on the operational conditions (e.g., weather conditions) at the k th time interval, and Φ_i is a set of capacity curve functions that represent all runway configurations of the i th airport under all weather conditions. $x_i(k)$, $y_i(k)$, $a_i(k)$, $d_i(k)$, $u_i(k)$, $v_i(k)$, and $z_{i,j}(k)$ are all nonnegative integers. Clearly, in this system, $x_i(k)$ and $y_i(k)$ are system states or output data and $a_i(k)$ and $d_i(k)$ are input data, while the control signals include $u_i(k)$, $v_i(k)$, and $z_{i,j}(k)$.

The model describes the concerned dynamics in a very straightforward way. For example, at each airport, the arrival queue by the beginning of the next interval depends on the arrival queue, the arrival demand, the arrival capacity, and the operation of rearranging flights between airports at current interval. If the current arrival capacity can cover all of the existing queue, new demand, and rearranged flights for the current interval, there will be no arrival queue by next interval; otherwise, those flights out of current capacity will be delayed as the queue at the beginning of next interval. The same goes for the departure case. The interaction between the arrival traffic and the departure traffic in the system is simply but well described by $z_{i,j}(k)$ and constraints (2)–(4). Constraints (3)

and (4) are specially given for the operation of rearranging arrival flights between airports. For example, constraint (4) implies that each airport can either redirect its own arrival flights to others or accept flights redirected from others but cannot do both at the same time. In the above model, we assume that the main airport is also capable of accepting redirected flights from satellite airports.

As discussed before, the actual queues under the real demands and real operational conditions in the system during the operating day are the main concern of the algorithm. Therefore, the performance of the proposed RHC-based GA will be judged by the objective function defined as

$$J_1 = \sum_{k=1}^T \sum_{i=1}^n \left(\alpha_i(k)x_i(k) + (1 - \alpha_i(k))y_i(k) + \beta_i(k) \sum_{j \neq i, j=1}^n z_{j,i}(k) \right) \quad (5)$$

where T denotes the number of 15-min intervals in the operating day, the coefficient $0 \leq \alpha_i(k) \leq 1$ determines the priority rate for arrivals at the i th airport at the k th time interval, the corresponding priority rate for departure is $(1 - \alpha_i(k))$, and the coefficient $\beta_i(k)$ determines the penalty for the operation of rearranging arrival flights between airports, which implies that the ideal situation is for all arrival flights to land at their originally scheduled airports. The determination of either $\alpha_i(k)$ or $\beta_i(k)$ depends on air traffic scenario and system infrastructure.

IV. RHC-BASED GA FOR MACM PROBLEM

The methodology of designing our RHC-based GA follows the common practice of GAs: Design the structure of chromosomes (data structures containing an evolvable description of a possible solution to a problem), choose a fitness function, define genetic operators, and introduce some necessary heuristic rules. In addition, for each step, we need to take an extra factor into account, i.e., how to integrate the concept of RHC. For general GA-related terms and techniques, we can refer to [14] and [15].

A. Structure of Chromosomes

In the RHC strategy, let us suppose that the receding horizon is N steps long. Then, at each time interval in the dynamic MACM problem, the capacity management and traffic rearrangement will be optimized online only for the current N intervals into the future. In other words, at the k th time interval, we just need to decide $u_i(k+l|k)$, $v_i(k+l|k)$, and $z_{i,j}(k+l|k)$, $l = 1, \dots, N$. Hereafter, $(\cdot|k)$ means the associated variable is predicted or calculated at the k th time interval. A chromosome represents a potential solution to the MACM problem over the receding horizon and, therefore, is constructed based on $u_i(\cdot|k)$, $v_i(\cdot|k)$, and $z_{i,j}(\cdot|k)$. The capacity allocation $(u_i(\cdot|k), v_i(\cdot|k))$ is subject to a certain capacity curve as depicted in Fig. 4, i.e., the point $(u_i(\cdot|k), v_i(\cdot|k))$ must be within the area encircled by both the axes and the capacity curve. It is easy to see that when choosing a point $(u_i(\cdot|k), v_i(\cdot|k))$ in

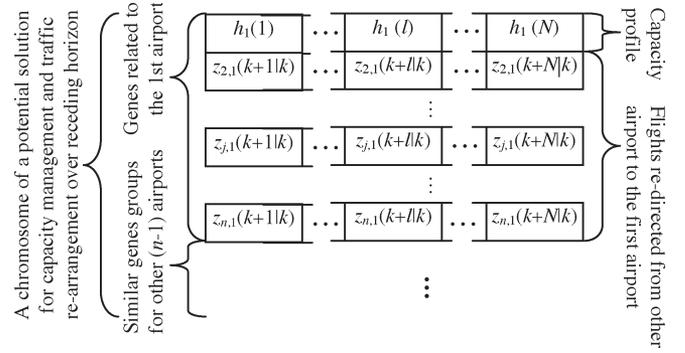


Fig. 5. Structure of chromosomes.

order to optimally serve both arrival and departure demands, we do not need to search the whole area encircled by the axes and the capacity curve but only need to test those points on or nearest the boundary of the tradeoff area, and we call these points tradeoff points (TOPs). For example, in the case given in Fig. 4(b), the four TOPs, represented by circles, can cover any workout by other points subject to the capacity curve. TOPs can be automatically determined by a computer program according to the following rule: For a valid point, which has integer coordinates (u, v) and is encircled by the axes and the capacity curve, if there exists another valid point that has the same arrival capacity u but a larger departure capacity v , or which has the same departure capacity v but a larger arrival capacity u , then the first valid point is not a TOP. By using TOPs, the size of solution space will be significantly reduced without sacrificing any performance. Suppose there are $H_i(k+l|k)$ TOPs in the capacity curve for the $(k+l)$ th time interval at the i th airport. Then, the structure of chromosomes is given in Fig. 5, where the i th $(N + N(n-1))$ genes group represents the capacity management and traffic rearrangement over the receding horizon at the i th airport. For the i th airport, the first N genes records the capacity profile over the receding horizon, and $h_i(l) \in [1, \dots, H_i(k+l|k)]$ is the serial number of TOP chosen for the $(k+l)$ th time interval; while the following $N(n-1)$ genes determine $z_{i,j}(\cdot|k)$ for traffic rearrangement, $j = 1, \dots, n$, and $j \neq i$.

Each chromosome defines a potential solution for capacity management and traffic rearrangement over the receding horizon. By checking the TOPs related to $h_i(\cdot)$, one has the values of $u_i(\cdot|k)$ and $v_i(\cdot|k)$. Together with $z_{i,j}(\cdot|k)$, one can calculate $x_i(\cdot|k)$ and $y_i(\cdot|k)$ over the receding horizon according to the model given by (1)–(4) and then assess the online solution quality by the new cost function

$$J_2(k) = \sum_{l=1}^N \lambda_i(l) \sum_{i=1}^n \left(\alpha_i(l)x_i(k+l|k) + (1 - \alpha_i(l)) \times y_i(k+l|k) + \beta_i(l) \sum_{j \neq i, j=1}^n z_{j,i}(k+l|k) \right) \quad (6)$$

where $\lambda_i(l) \geq 0$ are weighting coefficients that determine the contribution of queues and redirected flights in each interval to the total cost.

B. Fitness Function

Based on $J_2(k)$, the fitness of the associated chromosome can be simply defined as

$$f = (J_{2,\max}(k) - J_2(k)) / J_{2,\max}(k) \quad (7)$$

where $J_{2,\max}(k)$ denotes the maximum cost in a certain generation of chromosomes. In general, due to the RHC strategy, not all air traffic for the rest of the operating day is considered in each time interval. The implementation of this RHC-based solution will obviously have a certain influence on the management of those flights out of the current receding horizon. Neither (6) nor (7) does anything to assess this influence. This could end up with a short-sighted algorithm.

In our RHC-based GA, the idea of using terminal penalty in the RHC proposed in control engineering is borrowed to define a novel fitness function as

$$\begin{aligned} \tilde{J}_2(k) = & J_2(k) + \rho \frac{J_2(k)}{M_{AC}(k)} \left(N_{AC}(k) - \tilde{M}_{AC}(k) \right) \\ & \times \left(\frac{N_{AC}(k) - \tilde{M}_{AC}(k)}{T - \min(T-1, (k+N))} \right) \Bigg/ \left(\frac{M_{AC}(k)}{N} \right) \end{aligned} \quad (8)$$

$$f = \left(\tilde{J}_{2,\max}(k) - \tilde{J}_2(k) \right) / \tilde{J}_{2,\max}(k) \quad (9)$$

where $\rho \geq 0$ is a weighting coefficient, T is the same as defined in (5), $N_{AC}(k)$ is the predicted number of total flights in the rest of the operating day, $M_{AC}(k)$ is the number of those flights under consideration over the receding horizon, $\tilde{M}_{AC}(k)$ is the number of those flights that will be allowed to land or depart over the receding horizon according to the optimization, and $\tilde{J}_{2,\max}(k)$ denotes the maximum $\tilde{J}_2(k)$ in a certain generation of chromosomes.

The second term on the right-hand side of (8) is the terminal penalty, which assesses the influence of the current traffic management on those flights outside the receding horizon. From (8), one can see that the terminal penalty is a function in terms of the average cost and the density of those flights under consideration over the receding horizon and the number and the density of those flights that are outside the receding horizon after the current run of online optimization routine. A larger average cost usually means fewer flights will be allowed to land or depart during the current time interval, and therefore, more flights will be left for future optimization processes. The number of those flights that will be allowed to land or depart over the receding horizon according to the optimization $\tilde{M}_{AC}(k)$ is probably different from $M_{AC}(k)$, which is the number of those flights under consideration over the receding horizon. Basically, a smaller $\tilde{M}_{AC}(k)$ means more flights are delayed into future management. If more flights are left for future optimization, or if the density of such flights, i.e., $(N_{AC}(k) - \tilde{M}_{AC}(k)) / (T - \min(T-1, (k+N)))$, is larger than the density of those flights under consideration over the receding horizon, i.e. $M_{AC}(k)/N$, the current solution de-

termined by the chromosome will have a stronger negative influence on the future. ρ is a constant weighting coefficient that determines the contribution of the terminal penalty to $\tilde{J}_2(k)$. Coefficient ρ needs to be chosen carefully. If the predicted information is more reliable, ρ can be set relatively larger to avoid short-sighted performance. However, if ρ is too large, the influence of uncertain information in the future will become unnecessarily significant. The choice of ρ also depends on the maximum $N_{AC}(k)$. Specially, for a given ρ , if the maximum $N_{AC}(k)$ is very large, then the algorithm could also become sensitive to uncertainties. Basically, for each different airport system, extensive simulation studies and/or experiments are necessary to find a proper value for ρ . In this paper, we set $\rho = 0.05$ for $N_{AC}(k) \leq 500$.

From (9), one has that if $\tilde{J}_2(k)$ is smaller, the fitness of the corresponding chromosome is larger, and consequently, it is relatively more likely to survive through the evolution and to produce offspring.

C. Genetic Operators

In our RHC-based GA for the MACM problem, mutation randomly changes the value stored in a gene within a certain range that is related to the variable represented by that gene. For example, $h_i(l)$ can vary randomly between 1 and $H_i(k+l|k)$, and the change of $z_{i,j}(\cdot|k)$ is subject to (3).

There are two kinds of crossover operators. In the first case, we randomly pick up two chromosomes and then exchange their genes associated with the $(k+l)$ th interval. In the other, for two given chromosomes, we exchange their genes that determine capacity profiles or traffic rearrangement over the receding horizon. Either kind of crossover could be chosen to apply to a certain chromosome at the same probability.

D. Heuristic Rules for Setting Algorithm Parameters

To improve the solution quality as well as to increase the converging speed of GA, special problem-oriented heuristic rules are always introduced for setting algorithm parameters in various practices of GA. The following are some heuristic rules proposed for our RHC-based GA to resolve the MACM problem.

- Since redirecting arrival flights to other airports will result in penalty according to (6), such operation should be avoided whenever possible in the first place. To this end, when a chromosome is initialized, those genes associated with capacity profiles are generated randomly first, and then those $z_{i,j}(\cdot|k)$ -related genes are filled based on the result of capacity allocation. In other words, we need to calculate new queues based on the capacity allocation before generating $z_{i,j}(\cdot|k)$ -related genes. If there is no queue at the i th airport by the end of the $(k+l)$ th interval, then $z_{i,j}(k+l|k) = 0$ for $j = 1, \dots, n, i \neq j$. Otherwise, $z_{i,j}(\cdot|k)$ will be given small values at high probability.
- Whenever $k+l > T$, all genes are set to zero.
- Basically, a larger N , i.e., the length of receding horizon, means a larger solution space to search. Therefore, N_g , which is the the maximum generations for evolution,

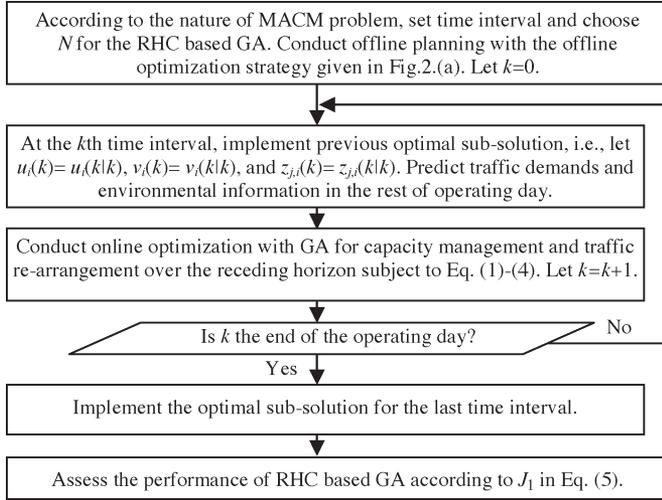


Fig. 6. Flow chart of RHC-based GA for MACM problem.

and N_p , which is the population of a generation, are set according to N as

$$N_g = 30 + 5 \max(0, (N - 8)) \quad (10)$$

$$N_p = 40 + 10 \max(0, (N - 8)). \quad (11)$$

- Self-adaptive crossover and mutation probabilities are introduced to prevent excellent chromosomes from being destroyed by evolutionary operators and to promote the evolution of inferior chromosomes [22], [23]. The crossover and mutation probabilities are calculated in (12) and (13), shown at the bottom of the page, where P_c and P_m are, respectively, the crossover and mutation probabilities for evolving a certain chromosome in a certain generation, f_{\max} is the maximum fitness in the generation, and f_{avg} is the average fitness of the generation. When the generation converges to a local optimum ($f_{\max} - f_{\text{avg}}$ is very small), according to (13), P_m will increase to diversify the following generation. In the reverse case ($f_{\max} - f_{\text{avg}}$ is very large), P_c will increase to speed up convergence. Furthermore, for those chromosomes with larger fitness, their P_c and P_m will be relatively smaller so that they can be protected effectively; otherwise, for those with smaller fitness, a larger P_c and P_m will be applied to improve them.

E. Flow Chart of the RHC-Based GA

With the above technical preparations, our RHC-based GA for the MACM problem can eventually be developed by

simply following the framework of common RHC algorithms, as illustrated by the flow chart in Fig. 6.

The successful design of the RHC-based GA partially depends on a proper choice of the length of receding horizon N . If N is too small, most useful information could be missed, and therefore, the RHC algorithm could be short sighted and exhibit poor performance. On the other hand, if N is too large, the computational burden will become very heavy, and in addition, much more unreliable information in the future will be used and could degrade the solution quality of the algorithm.

To assess whether a RHC-based GA is properly designed as well as to fairly compare with other relevant literature, the last step in the flow chart uses the objective function in (5) to assess the final performance.

V. SIMULATION RESULTS

For simplicity, we consider a two-airport system in this section. Traffic flow data at each airport are initially predicted over a 3-h period, and it is assumed that there is no traffic beyond the 3-h period on that day, i.e., the operating day is 3 hours long. Each time interval is set as 15 min, so there are 12 15-min intervals in the operating day. Table I shows the predicted arrival and departure demands at the two airports for each 15-min interval of the operating day. The TOPs in capacity curves for VFR and IFR operational conditions at each airport are given in Table II.

In this section, the airport capacity allocation is optimized in three different ways, namely 1) CDO-based GA (CDO_GA); 2) the proposed RHC-based GA (RHC_GA); and 3) the RHC-based linear programming method in [10] (RHC_LP). In CDO_GA, chromosomes have similar structures, as illustrated in Fig. 5, but represent potential solutions for the rest of operating day. In (6), N should be replaced by T . Fitness is calculated according to (7) rather than (9). Since RHC_LP in [10] is just for the ACM problem at a single airport, in our simulation, we apply it to each airport separately and then assess its performance in terms of J_1 in (5). For RHC_GA and RHC_LP, $N = 3$ in the simulation except explicitly indicated. The weighting coefficient $\beta_i(l) = 0.5$. To make a fair comparison with [10], the weighting coefficient $\lambda_i(l)$ for the CDO-based and RHC-based methods is, respectively, given by

$$\lambda_i(l) = 13 - l, \quad l = 1, \dots, 12 \quad (14)$$

$$\lambda_i(l) = 5 - l, \quad l = 1, \dots, 4. \quad (15)$$

Table III gives an example of capacity allocation and traffic rearrangement during the operating day. The results are calculated under RHC_GA with $\alpha_i(k) = 0.5$ and $N = 3$ based on

$$P_c = \begin{cases} 0.8(f_{\max} - f_{\text{avg}})/f, & (f_{\max} - f_{\text{avg}}) < f \\ 0.8, & (f_{\max} - f_{\text{avg}}) \geq f \end{cases} \quad (12)$$

$$P_m = \begin{cases} 0.4(f_{\max} - f)/(f_{\max} - f_{\text{avg}}), & (f_{\max} - f) < (f_{\max} - f_{\text{avg}}) \\ 0.4, & (f_{\max} - f) \geq (f_{\max} - f_{\text{avg}}) \end{cases} \quad (13)$$

TABLE I
ARRIVAL FLOW AND DEPARTURE FLOW OVER OPERATING DAY

The <i>k</i> th interval		1	2	3	4	5	6	7	8	9	10	11	12	Sum
Airport 1	Arr.	26	38	42	29	6	13	14	20	40	25	13	12	278
	Dep.	36	32	9	15	7	10	17	33	34	22	13	1	229
Airport 2	Arr.	20	21	4	29	30	8	15	11	9	19	30	8	204
	Dep.	27	11	4	24	5	7	25	13	7	17	25	1	166

TABLE II
TOPS IN CAPACITY CURVES FOR VFR AND IFR

TOPs		1	2	3	4	5	6	7	8	9	10	11
AirPort 1	VFR	17, 30	18, 29	19, 28	20, 27	21, 26	22, 25	24, 24	25, 21	26, 19	27, 17	28, 15
	IFR	12, 22	13, 21	14, 20	15, 19	16, 18	17, 17	18, 15	19, 13	20, 11	--	--
AirPort 2	VFR	14, 26	15, 25	16, 24	18, 23	19, 22	21, 21	22, 18	23, 15	24, 12	--	--
	IFR	10, 19	11, 18	12, 17	13, 16	15, 15	16, 12	17, 9	--	--	--	--

TABLE III
EXAMPLE OF CAPACITY MANAGEMENT BY RHC_GA ($\alpha_i(k) = 0.5$)

<i>k</i> th 15-min time interval	Airport 1					Airport 2				
	Arrival			Departure		Arrival			Departure	
	Cap.	$z_{2,1}$	Que.	Cap.	Que.	Cap.	$z_{1,2}$	Que.	Cap.	Que.
1	19	0	7	28	8	19	0	1	22	5
2	24	0	21	24	16	22	0	0	18	0
3	25	0	17	21	4	23	21	2	15	0
4	25	0	21	21	0	21	0	10	21	3
5	26	0	1	19	0	24	0	16	12	0
6	24	0	0	24	0	24	0	0	12	0
7	21	0	0	26	0	15	0	0	25	0
8	17	0	0	30	3	21	3	0	21	0
9	17	0	9	30	7	23	14	0	15	0
10	24	0	10	24	5	19	0	0	22	0
11	26	1	0	19	0	21	0	8	21	4
12	24	0	0	24	0	21	0	0	21	0
Total	272	1	86	290	43	253	38	37	225	12
RHC_LP	278	--	143	229	77	204	--	31	166	12

TABLE IV
COMPARATIVE RESULTS IN STATIC ENVIRONMENT

	Case 1 ($\alpha_i(k) = 0.5$)				Case 2 ($\alpha_i(k) = 0.7$)			
	Arr./Dep Queue	RA Traf.	J_1	ACT (sec.)	Arr./Dep Queue	RA Traf.	J_1	ACT (sec.)
RHC_GA	150/51	29	115	19.75	63/147	46	111.2	20.39
CDO_GA	167/63	23	126.5	321.5	81/186	40	132.5	315.9
RHC_LP	174/89	--	131.5	0.23	113/219	--	144.8	0.28

the assumption that no uncertainties are present, i.e., the actual demands are the same as the predicted ones in Table I, and two airports are always under the VFR operational condition. From Table III, it is evident that the optimal solution provides a time-varying capacity profile and traffic rearrangement, both of which efficiently solve the predicted congestion problem by reflecting the dynamics of traffic demand in the system. The last row in Table III shows the total queues optimized by RHC_LP. One can see that, at Airport 1, the total arrival/departure queue under RHC_GA is 57/34 flights less than the corresponding queue under RHC_LP at the small cost that the arrival queue at Airport 2 increases by six flights. According to the common objective function J_1 given in (5), RHC_GA achieves a value of 108.5, which is much smaller than 131.5, which is the value under RHC_LP. The main reason for this is that RHC_LP does not support the operation of rearranging traffic due to the

linear-programming-based optimizer, while under RHC_GA, a total of 39 arrival flights are redirected between Airports 1 and 2. However, before we can make further conclusions on RHC_GA, extensive simulation study needs to be conducted.

One of the main purposes of this section is to compare RHC_GA with RHC_LP and CDO_GA in order to assess our new algorithm in terms of both performance and computational efficiency. The RHC strategy is expected to bring benefits in a dynamic environment, but it is still necessary to investigate the performance of RHC_GA in a static environment, i.e., there are no uncertainties, and the real arrival and departure traffic and operational conditions are exactly the same as predicted. Table IV shows some results of this comparison, where the actual traffic demands are the same as the predicted ones in Table I, and the operational condition is VFR. In each case, the data related to either RHC_GA or CDO_GA are based on

TABLE V
COMPARATIVE RESULTS IN DYNAMIC ENVIRONMENT

	Case 3 ($\alpha_i(k)=0.5$)				Case 4 ($\alpha_i(k)=0.7$)			
	Arr./Dep Queue	RA Traf.	J_1	ACT (sec.)	Arr./Dep Queue	RA Traf.	J_1	ACT (sec.)
RHC_GA	381/151	53	292.5	21.03	211/427	71	311.3	21.98
CDO_GA	409/189	45	321.5	331.7	258/449	65	347.8	321.0
RHC_LP	435/219	--	327	0.25	305/482	--	358.1	0.27

TABLE VI
INFLUENCE OF N ON RHC_GA (TIME INTERVAL IS 15 min)

N	1	2	3	4	5	6
J_1	120.7	113.6	115	116.1	118.9	122.4
$J_1 \times 3$	362.1	340.8	345	348.3	356.7	367.2
ACT(s)	8.73	14.47	19.75	25.14	39.62	85.39

TABLE VII
INFLUENCE OF N ON RHC_GA (TIME INTERVAL IS 5 min)

N	1	2	3	4	5	6	7	8	9
J_1	366.5	337.8	328.6	330.2	325.7	329.1	331.8	334.2	341.5
ACT(s)	4.21	6.74	8.02	11.25	13.39	15.44	17.79	20.17	23.68

50 simulation runs, while the data of RHC_LP are based on one simulation run. Hereafter, ACT stands for average computational time. Table V further gives results of comparing three algorithms in a dynamic environment, where 20% of predicted traffic demands and predicted operational conditions are uncertain. Uncertainties are generated randomly over the operating day, and then, this history of uncertainties is saved in a database as one record. Each time the three algorithms are tested and compared, the same record of uncertainties is applied. For each case listed in Table V, 20 records of uncertainties are used; with each record of uncertainties, 50 simulation runs are conducted under either RHC_GA or CDO_GA, while one simulation is run under RHC_LP.

From Tables IV and V, one can see that, due to no traffic rearrangement in RHC_LP, its performance is the worst, which means in a multi-airport system, the operation of traffic rearrangement between airports can effectively reduce the total queue. Although CDO_GA also considers traffic rearrangement, the CDO strategy requires optimizing the traffic flow over the rest of the operating day, which usually means a huge solution space for the GA-based optimizer to search, and therefore, the performance is not significantly improved compared with RHC_LP. By integrating the RHC strategy into GA, our new algorithm achieves the best performance. As for online computational efficiency, RHC_LP is the fastest, because it employs a linear programming algorithm. CDO_GA, again due to the CDO strategy, is the slowest algorithm. Again, thanks to the RHC strategy, the computational time of RHC_GA is significantly reduced when compared with CDO_GA. Since each time interval is 15 min (900 s), our RHC_GA is perfectly ready for real-time implementations. Therefore, from Tables IV and V, one will come to the conclusion that the RHC_GA proposed in this paper is the best algorithm in terms of tradeoff between solution quality and computational efficiency. Tables IV and V also show that the arrival priority coefficient $\alpha_i(k)$ has significant influence on the final solution.

Another objective of this section is to study how to choose those RHC-related parameters, such as length of horizon and time interval, for RHC_GA such that the RHC strategy can be effectively integrated into GA for the dynamic MACM problem. The results of study are listed in Tables VI and VII. In Table VI, each time interval is as before, i.e., 15 min, and the length of the receding horizon N changes from 1 to 6; while in Table VII, each time interval is reduced to 5 min, and N changes within the range of 1 to 9. In Tables VI and VII, the traffic demands given in Table I are used. For Table VII, where each time interval is just 5 min, each traffic demand for a 15-min interval in Table I is randomly divided into three successive 5-min intervals, and those TOPs given in Table II also need to be modified according to the 5-min interval. In the simulation, at $\alpha_i(k) = 0.5$, there are no uncertainties on either demands or operational conditions, and with each N , 50 simulation runs are conducted.

From Tables VI and VII, one can clearly see what is the influence of N on the performance and the computational burden of RHC_GA. In general, as N increases, the performance of RHC_GA improves at first and then degrades when N is too large. This is understandable. Due to the nature of GA, the performance should degrade with the length of the receding horizon, but a too short receding horizon could result in short-sighted performance, e.g., when $N = 1$. Clearly, in this simulation, the receding horizon should be within 15 to 45 min in order to achieve the best performance. Online computational burden is no doubt increasing as N goes up.

One should note that the value of J_1 approximately indicates how many flights are delayed. When the time interval is 15 min, as in Table VI, one flight equals a 15-min delay, while in Table VII, where the time interval is 5 min, one flight means a 5-min delay. Therefore, to fairly compare the data in Tables VI and VII, one needs to multiply those values of J_1 in Table VI by 3, as shown in the third row of Table VI. Comparing $J_1 \times 3$ in Table VI with J_1 in Table VII, one can see

that, generally, RHC_GA achieves better performance with the 5-min interval than with the 15-min interval. This is reasonable because a shorter time interval means more flexibility in the airport operation. Table VII illustrates that shortening each time interval, e.g., from 15 to 5 min, is an effective way to improve the flexibility and performance of the ACM.

VI. CONCLUSION

A GA based on the concept of RHC is proposed for solving the dynamic ACM problem in a multiairport system, where flights can be redirected between airports. The methodology of integrating the RHC strategy into a GA for real-time implementations in a dynamic ACM environment is systematically studied. How to design the GA from an RHC point of view and how to choose those methodology-related parameters such as time interval, length of receding horizon, and terminal penalty are investigated in depth. Simulation results show that the new method proposed in this paper is effective and efficient to solve the ACM problem in a dynamic multiairport environment.

However, more effort is required before any real applications of the proposed method can happen. For example, a more complicated model should be used, more real traffic flow data and scenarios are needed for experiments, more widely comparative work should be carried out, and further systematic analyses are required.

REFERENCES

- [1] "1993 Aviation system capacity plan," in "FAA, Office of System Capacity and Requirements," Washington, DC, Rep. DOT/FAA/ASC-93-1, 1993.
- [2] E. P. Gilbo, "Airport capacity: Representation, estimation, optimization," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 3, pp. 144–154, Sep. 1993.
- [3] G. Hutting, W. Busch, and N. Gronak, "Growing demand and capacity of airports," *Transp. Res., A*, vol. 28-A, no. 6, pp. 501–509, 1994.
- [4] A. Blumstein, "An analytical investigation of airport capacity," Cornell Aeronautical Lab., Buffalo, NY, Rep. TA-1358-8-1, 1960.
- [5] W. J. Swedish, "Upgraded FAA airfield capacity model," MITRE Corp., Bedford, MA, Tech. Rep. MTR-81W16, 1981, vol. 1 and 2.
- [6] M. F. Medeiros, "Advanced traffic management system automation," *Proc. IEEE*, vol. 77, no. 11, pp. 1643–1652, Nov. 1989.
- [7] I. Frolow and J. H. Sinnott, "National airspace system demand and capacity modeling," *Proc. IEEE*, vol. 77, no. 11, pp. 1618–1624, Nov. 1989.
- [8] V. H. L. Cheng, V. Sharma, and D. C. Foyle, "A study of aircraft taxi performance for enhancing airport surface traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 2, pp. 39–54, Jun. 2001.
- [9] E. P. Gilbo, "Optimizing airport capacity utilization in air traffic flow management subject to constraints at arrival and departure fixes," *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 5, pp. 490–503, Sep. 1997.
- [10] W. H. Chen and X. B. Hu, "Receding horizon control for airport capacity management," in *Proc. Control*, Bath, U.K., Sep. 6–9, 2004, pp. 1–6.
- [11] Z. K. Shi, P. Zhou, and L. J. Wang, "Multi-airport open network for air traffic flow management," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 9–11, 2003, pp. 53–57.
- [12] L. Bianco, P. Dell'Olmo, S. Giordani, and G. Lulli, "Models and algorithms for multi-airport traffic flow coordination," in *Proc. ATM*. Capri, Italy, Sep. 22–26, 2002.
- [13] G. Andreatta and L. Brunetta, "Multiairport ground holding problem: A computational evaluation of exact algorithms," *Oper. Res.*, vol. 64, no. 1, pp. 57–64, Jan. 1998.
- [14] E. Goldberg, *Genetic Algorithms in Search Optimization & Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1994.
- [16] X. B. Hu and W. H. Chen, "Genetic algorithm based on receding horizon control for arrival sequencing and scheduling," *Eng. Appl. Artif. Intell.*, vol. 18, no. 5, pp. 633–642, Aug. 2005.
- [17] D. W. Clarke, *Advances in Model-Based Predictive Control*. London, U.K.: Oxford Univ. Press, 1994.
- [18] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," *Automatica*, vol. 37, no. 7, pp. 1049–1056, Jul. 2001.
- [19] T. Van Den Boom and B. De Schutter, "Properties of MPC for max-plus-linear systems," *Eur. J. Control*, vol. 8, no. 5, pp. 453–462, 2002.
- [20] S. Chand, V. N. Hsu, and S. Sethi, "Forecast, solution, and rolling horizons in operations management problems: A classified bibliography," *Manuf. Serv. Oper. Manag.*, vol. 4, no. 1, pp. 25–43, Dec. 2002.
- [21] W. H. Chen, T. van den Boom, and B. De Schutter, "Discussion on: 'Properties of MPC for max-plus-linear systems' by T. van den Boom and B. De Schutter," *Eur. J. Control*, vol. 8, no. 5, pp. 463–464, 2002.
- [22] Z.-Y. Wu, "An new adapted GA & its application in multimodal function optimization," *Control Theory Appl.*, vol. 16, no. 1, pp. 127–129, 1999.
- [23] X. B. Hu, S. F. Wu, and J. Jiang, "On-line free-flight path optimization based on improved genetic algorithms," *Eng. Appl. Artif. Intell.*, vol. 17, no. 8, pp. 897–907, Dec. 2004.



Xiao-Bing Hu received the B.S. degree in aviation electronic engineering from the Civil Aviation Institute of China, Tianjin, China, in 1998, the M.S. degree in automatic control engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001, and the Ph.D. degree in aeronautical and automotive engineering from Loughborough University, Leicestershire, U.K., in 2005.

He is currently a Research Fellow with the Department of Informatics, University of Sussex, Brighton,

U.K. His major field of research includes predictive control, artificial intelligence, air traffic management, and flight control.



Wen-Hua Chen (M'00) received the M.Sc. and Ph.D. degrees from the Department of Automatic Control, Northeast University, Shenyang, China, in 1989 and 1991, respectively.

From 1991 to 1996, he was a Lecturer with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 1997 to 2000, he held a research position and then a Lectureship in control engineering with the Center for Systems and Control, University of Glasgow, Glasgow, U.K. He currently holds a Senior

Lectureship in flight control systems with the Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, U.K. He has published one book and more than 80 papers in journals and conferences. His research interests are the development of advanced control strategies and their applications in aerospace engineering.



Ezequiel Di Paolo received the M.S. degree in nuclear engineering from the Instituto Balseiro, Bariloche, Argentina, and the D.Phil. degree from University of Sussex, Brighton, U.K.

He is currently a Senior Lecturer with the Evolutionary and Adaptive Systems Group, University of Sussex. He is also with the Centre for Research in Cognitive Science, University of Sussex. His research interests include adaptive behavior in natural and artificial systems, biological modeling, evolutionary robotics, and enactive cognitive science.