Receding horizon control for airport capacity management

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- This is a journal article. It was published in the journal, IEEE Transactions on Control Systems Technology [© IEEE] and is also available at: http://ieeexplore.ieee.org. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/4005

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Receding Horizon Control for Airport Capacity Management

Wen-Hua Chen, Senior Member, IEEE, and Xiao-Bing Hu

Abstract—A major goal of air traffic management is to strategically control the flow of traffic so that the demand at an airport meets but does not exceed the operational capacity in a dynamic environment. This paper uses the concept of receding horizon control (RHC) to conduct real-time planning for airport capacity management (ACM). It is shown that RHC provides a generic and flexible framework for developing real-time allocation algorithms for airport capacity in a dynamic and uncertain environment, and existing approaches such as the one step ahead adjustment can be considered as special cases of this approach. Robustness against the change of the environment and demands and computational efficiency are two advantages when applying RHC to the ACM problem, which are illustrated by a case study.

Index Terms—Airport capacity management (ACM), delay, optimization, queue, receding horizon control (RHC).

I. INTRODUCTION

THE phenomenon of growing air traffic demand should be met by a concomitant improvement in airport capacity [1], [2]. The role of airport capacity management (ACM) becomes especially significant, and extensive research work has been conducted to attack the problem in the past decades. The accurate and reliable prediction of airport capacity and its demands is crucial to the effectiveness of the strategic traffic management programs. There exist methods and tools for predicting air traffic demand, e.g., see [3], and the problem of predicting airport capacity is now also well resolved, e.g., see [2], [4], and [5]. Besides presenting an empirical approach to estimate airport capacity, [2] also reported a method for optimization of airport capacity using the derived estimates. The optimization is achieved by allocation of the capacity over time between arrivals and departures. In general, the optimal solution provides time-varying capacity profiles. This approach was further extended in [6] to a more complicated airport system where the runways and arrival and departure fixes were considered jointly, and the limits of the fixes were also taken into account.

The previously mentioned models and methods can be used by traffic managers and controllers as automated support tools for decision making on capacity management at airports during periods of congestion. For a given time period, runway configuration, weather forecast, and predicted arrival and departure demands for runways and fixes (input data), one can determine the optimal strategy for managing arrival/departure traffic at an airport (output), i.e., how many flights can be accepted (arrivals) and released (departures) during congested periods at the airport, how many flights are to be delayed, and by how long. In the real world, airports operate in a very dynamic environment; demands and weather vary with time. Therefore, automated support tools are required to reallocate the airport capacity constantly according to updated input data at each time interval. In this paper, ACM with changes or uncertainties in the demand and operational conditions is referred to as a dynamic environment; in contrast to that, a static environment stands for ACM problems where the real situation such as arrival demand and weather condition is the same as predicted before the operation. Most existing tools were developed for a static environment such as in [1], [2], and [6], and could not directly cope with the changes of environment and demands. Their focus was on developing offline planning tools for ACM.

This paper introduces the concept of receding horizon control (RHC), or model predictive control (MPC), into the real-time optimization of airport capacity allocation in response to the change of the demands and environment. Simply speaking, RHC is an $N$-step ahead online optimization strategy. At each time interval, based on the current available information, RHC optimizes the concerned problem for the next $N$ intervals in the near future, and only the part of the solution corresponding to the current interval is implemented. At the next interval, RHC repeats the similar optimizing procedure for another $N$ intervals ahead based on updated information. RHC has now been widely accepted in the area of control engineering, and proven to be promising in many aspects against other control strategies [7]. Recently, attention has been received in applications of RHC or similar ideas to other areas such as management and operations research [8], [9]. However, the research work on applying RHC to areas other than control engineering is just beginning [10].

On the other hand, the idea of rolling horizon, similar to RHC, is sometimes applied for aircraft arrival sequencing and scheduling at airports [10]–[12]. However, no systematical work has ever been reported to introduce the concept of RHC into the ACM problem. In this paper, we attempt to effectively apply the RHC strategy to solve the ACM problem. The main goal is to investigate whether RHC could be a generic and promising approach for real-time airport capacity management.

II. MODELING OF ARRIVAL-DEPARTURE SYSTEM AT AN AIRPORT

In this paper, the traffic demand for the airport is given by the predicted numbers of arriving and departing flights per each 15-min interval in the operating day; therefore, the time interval for capacity allocation is 15 min long. The operational limits on the ground (runways) are characterized by arrival capacity and departure capacity. These capacities are generally variable and interdependent.
How to optimally allocate the 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 runway configurations that practice the tradeoff between arrival and departure capacities. For these configurations the arrival capacity $a$ and departure capacity $d$ are interdependent and can be represented by a functional relationship $v = \Phi(u)$. Generally, the function is a linear piecewise convex one. Graphical representation of the function on the “arrival capacity–departure capacity” plane is called the airport capacity curve [2], [5]. Fig. 1(a) 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 optimization 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: visual flight rules (VFR); marginal VFR (MVFR); instrument flight rules (IFR); and Low IFR (LIFR). Capacity curves vary for these four different weather categories. For the sake of simplicity, only two weather conditions, VFR and IFR, are considered in this paper. Fig. 1(b) gives the airport capacity curves for VFR and IFR operational conditions at the Chicago O’Hare International Airport (ORD) [2]. The coordinates of vertices of the curves show some capacity values (the first number corresponds to the arrival capacity). According to Fig. 1(b), the IFR capacities are approximately 30% less than the VFR capacities.

With capacity curves at an airport, one can establish the following constrained state-space based model to describe the dynamics of the airport capacity system, [2], [6], i.e., the functional relationship between the input data (airport capacities and predicted information) and the output (arrival and departure queues)

$$\begin{align*}
x(k+1) &= \max(0, x(k) + a(k) - u(k)) \\
y(k+1) &= \max(0, y(k) + d(k) - v(k)) \\
x(0) &= x_0 \\
y(0) &= y_0
\end{align*}$$

subject to constraint

$$0 \leq v(k) \leq \phi(k, u(k)), \phi(k, u(k)) \in \Phi$$

where $k$ is the discrete time index, $x(k)$ and $y(k)$ are, respectively, the arrival queue and departure queue at the beginning of the $k$th time interval, $a(k)$ and $d(k)$ are, respectively, the demand for arrivals and for departures at the $k$th time interval, $u(k)$ and $v(k)$ are, respectively, the airport arrival capacity and departure capacity at the $k$th time interval, $\phi(k, u(k))$ is the arrival/departure capacity function that depends on the operational conditions (e.g., weather conditions) at the $k$th time interval, and $\Phi$ is a set of capacity function curves that represent all runway configurations of the airport under all weather conditions. It shall be noted that $x(k)$, $y(k)$, $a(k)$, $d(k)$, $u(k)$, and $v(k)$ are all nonnegative integers and the ACM model in (1) is a birth-death process. This makes the dynamic process of the ACM much different from other applications of RHC, or MPC, where usually the process dynamics can be described by linear/nonlinear differential or difference equations [7], [13].

Models (1) and (2) describe the dynamics of an airport capacity system in a straightforward way. For example, the arrival queue at the beginning of the next interval depends on the arrival queue, the arrival demand, and the arrival capacity at the current interval. If the current arrival capacity can cover both existing queue and new demand for the current interval, there will be no arrival queue by the next interval; otherwise, those flights out of the current capacity will be delayed as the queue at the beginning of the next interval. Similar is the departure case. The interaction between the arrival traffic and the departure traffic is described by constraint (2).

III. RHC of ACM

There are many papers studying the ACM problem in a static environment [2], [6]. The tools developed in these works can be used for offline planning, i.e., optimally allocate the arrival and departure capacity for each time interval according to predicted information, which are referred as offline optimization in this paper and illustrated in Fig. 2(a) for the purpose of predicted information. However, in the real world, the traffic demands and operational conditions at the airport may vary over time, and are difficult to predict precisely. Therefore, how to use the updated information of the demands and operational conditions in ACM operation becomes strategically important. In other words, the ACM problem needs to be addressed from a dynamic point of view.

This paper aims to use the RHC strategy to conduct real-time airport capacity allocation in a dynamic environment, and the optimization of airport capacity refers to the best allocation of airport capacities between arrivals and departures that optimally satisfies the real (not predicted) demands over the operating day under the real (not forecasted) operational conditions at the airport. The basic idea of RHC is illustrated in Fig. 2(d).

There are, in general, two intuitive ways to explore an airport capacity allocation strategy developed originally for a static environment to a dynamic environment. One is to use the tool in advance to optimize capacity allocation at the airport for an operating day. After the offline optimal solution is generated, necessary adjustment needs to be made, by either traffic managers/controllers or tools, according to updated information for the current time interval. This is referred to as one-step ahead (OSA) adjustment as shown in Fig. 2(b). Obviously this method might be shortsighted and is unlikely to achieve the optimal solution in a dynamic environment. Another way is to online apply
(a). Offline optimization: Optimize capacity profile for the entire operating day based on the predicted information before the operating day and generate a static optimal solution.

Start of operating day \hspace{1.5cm} \text{End of operating day}

(b). One-step-ahead adjustment: Make adjustment only for the current time interval based on the latest updated information and the static solution of the offline optimization.

Start of operating day \hspace{1.5cm} k \hspace{1.5cm} k+1 \hspace{1.5cm} \text{End of operating day}

(c). Conventional dynamic optimization (CDO): Optimize capacity profile over the period from the current time \( k \) to the end of the operating day, and then execute the optimal sub-solution over the period from \( k \) to \( k+1 \). At time \( k+1 \), repeat the same procedure based on new information. And so on.

Start of operating day \hspace{1.5cm} k \hspace{1.5cm} k+1 \hspace{1.5cm} \ldots \hspace{1.5cm} k+N \hspace{1.5cm} \text{End of operating day}

(d). Receding Horizon Control (RHC): Optimize capacity profile over the predictive horizon (from the current time \( k \) to time \( k+N \)), and then execute the optimal sub-solution over the period from \( k \) to \( k+1 \). At time \( k+1 \), repeat the same procedure based on new information. And so on.

Start of operating day \hspace{1.5cm} k \hspace{1.5cm} k+1 \hspace{1.5cm} \ldots \hspace{1.5cm} k+N \hspace{1.5cm} \text{End of operating day}

Fig. 2. Some optimization strategies.

A capacity allocation strategy for the rest of the operating day at each time interval, which is referred to as the conventional dynamic optimization (CDO) shown in Fig. 2(c). This often leads to a large amount of online computational burden and may lead to difficulties in real-time implementation, in particular for large scale problems, while the resulting solution is not necessarily better since the reliability of the information usually decays with time.

These two intuitive approaches can actually be considered as special cases of RHC framework proposed in this paper; they are two extreme cases of RHC framework. OSA adjustment is an RHC algorithm with the length of the receding horizon \( N = 1 \), while the conventional dynamic optimization is equivalent to an RHC algorithm with a sufficiently large receding horizon. Therefore, RHC provides a generic and flexible framework for dynamic allocations of airport capacity, where the length of the receding horizon and the performance index for each step of optimization are design parameters. It may achieve much better tradeoff among performance (in terms of a given performance index), online computational burden, and robustness and adaptation to the change of environment and demands.

The effectiveness of arrival and departure operations at the airport can be measured by either 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. The weighted total number of flights in the queues has been chosen as the performance index for capacity optimization in this paper, i.e., the performance of the proposed RHC algorithm is judged by

\[
J_1(k) = \sum_{i=1}^{N_T} \left( \alpha(i)x(i) + (1 - \alpha(i))y(i) \right)
\]

(3)

where \( N_T \) denotes the number of 15-min intervals in the operating day, the coefficient \( 0 \leq \alpha(i) \leq 1 \) determines the priority rate for arrivals at the \( i \)th time interval, and the corresponding priority rate for departures is \( (1 - \alpha(i)) \), which depends on the difference in cost and safety caused by arrivals and departures.

However, \( J_1 \) in (3) is a static performance index, and, therefore, needs to be modified properly into a dynamic one for online optimization. On one hand, optimization of the performance index (3) at each step does not necessarily give the best overall performance in terms of the same performance; on the other hand, only a finite number of steps ahead are considered for online optimization in the RHC framework. To compensate that, in general, a terminal penalty has to be introduced in the cost for each step optimization, for example, see [7], which takes into account the time-to-go cost. Therefore, for each step, the online optimization performance index is chosen as

\[
J_2(k) = \sum_{i=1}^{N} \beta(i) \left( \alpha(i)x(k+i|k) + (1 - \alpha(i))y(k+i|k) \right) + J_1(x(k+N|k), y(k+N|k))
\]

(4)

where \( N \) denotes the length of the receding horizon, \( (k|k) \) shows that the corresponding variables are calculated or predicted at time instant \( k \), and \( \beta(i) \geq 0 \) is introduced to reflect the contribution of queues in each interval to the total cost. A special case for \( \beta(i) \) is constant as in [6]. However, since the predicted information for the far future in the receding horizon is more likely to change, \( \beta(i) \) should decrease as \( i \) increases. The length of receding horizon \( N \) should be set carefully to avoid either being shortsighted or including too much unreliable information. The choice of \( N \) mainly depends on the reliability of the predicted information of the operating day. When the reliability is high, a reasonably large \( N \) could lead to a good result; but after a critical value, increasing \( N \) may bring more benefits. \( J_1(x(k+N|k), y(k+N|k)) \) is the penalty taking into account the cost beyond the horizon for optimization. In general, it is a function of the final arrival and departure queue in the planning horizon.

Choosing proper parameters for the performance index (4) is crucial to a successful implementation of the RHC. The main considerations are to improve the real-time ability (reducing the computational burden) and to increase robustness and adaptation to the change of information but without significant degradation of the performance in the absence of the changes or uncertainties. More discussion about the design of RHC algorithms can be found in the case study in Section IV.

Based on the previous performance index, our RHC algorithm for the ACM problem is described as follows.

Step 1) At the beginning of the operating day, i.e., \( k = 0 \), take the information of the actual arrival queue \( x(0) \), departure queue \( y(0) \) and operational condition \( \phi(0,u) \), and the predicted information over the receding horizon, i.e., predicted arrival demands \( a(k|0) \), predicted departure demands \( d(k|0) \), and predicted weather conditions \( \phi(0,u) \), \( i = 1, \ldots, N \), supplied by other departments.

Step 2) At time interval \( k \), solve the online optimization problem formulated as follows:

\[
\min_{u(k|k), \ldots, u(k+N-1|k), x(k|k), \ldots, x(k+N-1|k)} J_2(k) \quad k \geq 0
\]

(5)
subject to (1) and (2). Then, allocate the airport capacities for the next 15-min interval according to the first part of the optimal solution, i.e.,

\[ u(k) = u(k|k), \quad v(k) = v(k|k). \]  

(6)

Step 3) Check if the current time interval is the last one of the operating day or not. If it is, go to Step 5); otherwise, go to Step 4).

Step 4) At the beginning of the \((k + 1)\)th time interval, update the new actual arrival queue \(x(k+1)\), departure queue \(y(k+1)\), and operational condition \(\phi(k+1)\), and the predicted information for the next receding horizon, i.e., predicted arrival demands \(a(k+1, u)\), and predicted departure demands \(d(k+1, u)\), and predicted weather conditions \(\phi(k+1, i)\), \(i = 1, \ldots, N. \) Let \(k = k + 1\), and then go to Step 2).

Step 5) Calculate the cost of the operating day according to \(J_1\).

At each time interval, after the information about the environment, demands, and the implementation results of the last decision are updated, the optimization problem (5) can be considered as one in a static environment with all the required information. Hence, any algorithm developed for ACM in a static environment can be employed. In this paper, the algorithm in [2] is adopted as the online optimizer in the proposed RHC method, where the ACM problem is reformed as a linear programming problem, but many results drawn in this paper can be easily extended to other optimization algorithms.

IV. CASE STUDY

In this section, the airport capacity allocation is optimized in three different ways: OSA adjustment, conventional dynamic optimization and the proposed RHC method. For the sake of identification, hereafter, they are denoted as OSA, conventional dynamic optimization (CDO), and RHC, respectively. The basic traffic flow data are taken from [6], where heavy traffic at the ORD was predicted over the 3-hr period on February 12, 1993 from 16:45 to 19:45 local time. For the sake of simplicity, it is assumed in this paper that there is no traffic outside the above 3-hr period on that day, i.e., the operating day is just 3-hr-long. The 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 airport for each 15-min interval of the operating day. The airport capacity curves for VFR and IFR operational conditions are shown in Fig. 1(b). OSA needs an offline plan based on the predicted demands and weather conditions over the operating day by solving the minimization problem with \(J_1\) in (3) and \(N_T = 12\).

Comparing RHC with OSA and CDO in a static environment, i.e., no uncertainties are present, is a necessary step to check whether RHC is well designed. Since in the absence of uncertainties, i.e., no change in demands or operational conditions, the offline planning and CDO shall give the best performance. This is used as a guide to tune the receding horizon size, the weighting coefficients \(\beta(i)\), and the cost function, and to check the RHC performance. Similar to most of the applications of the RHC concept, initially a terminal weighting term was introduced in the cost function \(J_2\) to take into account the remaining cost. However, it was then found that similar performance as offline planning and CDO, i.e., the optimal performance, can be achieved without a terminal penalty. As a result, a terminal weighting term was not introduced in the cost function. In the case study, various weighing coefficients and the length of the receding horizon are investigated. In general, the weighing coefficients depend on the uncertainties or the change of the demands and environment. It is found that \(\beta(i) = 2N - i\) gives quite good results. Other weighting coefficients may also give acceptable results. In the following tables, the receding horizon for RHC is chosen as four-steps long, i.e., \(N = 4\). In that case, the same weighting coefficients used in [2] and [6], i.e., \(\beta(i) = 1\), achieve a little bit worse performance.

Table II gives an example of the airport capacity allocation during the operating day. The results were calculated with RHC under the assumption that no uncertainties are present, i.e., the actual demands are the same as the predicted ones in Table I, and the airport is always under the VFR operational condition. It is evident in Table II that the optimal solution provides a time-varying capacity profile that most efficiently solves the predicted congestion problem by reflecting the dynamics of the traffic demand at the airport. The arrival priority coefficient \(\alpha(i)\) has significant influence on the final solution. As shown in Table II, \(\alpha(\hat{i}) > 0.5\), which reflects the reality that airborne delay is less desirable than ground delay due to safety and cost concerns in real air traffic management, results in reduced arrival queues but increased departure queues. This observation is also valid for the other tables.

More simulation results are given in Tables III–VII, where “A.Q.” stands for arrival queue, “D.Q.” for departure queue, and “C.T.” for computational time.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>ARRIVAL FLOW AND DEPARTURE FLOW OVER THE OPERATING DAY [6]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4th 15-min interval</td>
<td>Arrival</td>
</tr>
<tr>
<td>Arr. Flow</td>
<td>26</td>
</tr>
<tr>
<td>Dep. Flow</td>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>CAPACITY ALLOCATION WITH RHC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4th 15-min interval</td>
<td>Arrival</td>
</tr>
<tr>
<td></td>
<td>(\alpha(i) = 0.5)</td>
</tr>
<tr>
<td>Arrival</td>
<td>24</td>
</tr>
<tr>
<td>Departure</td>
<td>24</td>
</tr>
<tr>
<td>Arrival</td>
<td>24</td>
</tr>
<tr>
<td>Departure</td>
<td>24</td>
</tr>
<tr>
<td>Arrival</td>
<td>25</td>
</tr>
<tr>
<td>Departure</td>
<td>25</td>
</tr>
<tr>
<td>Arrival</td>
<td>26</td>
</tr>
<tr>
<td>Departure</td>
<td>26</td>
</tr>
<tr>
<td>Arrival</td>
<td>24</td>
</tr>
<tr>
<td>Departure</td>
<td>24</td>
</tr>
<tr>
<td>Arrival</td>
<td>24</td>
</tr>
<tr>
<td>Departure</td>
<td>18</td>
</tr>
<tr>
<td>Total</td>
<td>278</td>
</tr>
</tbody>
</table>
Tables III and IV are some results of studying the RHC performance in a static environment, where the actual traffic demands are the same as the predicted ones in Table I. In Table III, the entire operating day is under the VFR operational condition, while in Table IV, IFR is for the first four 15-min intervals, and VFR for the rest. Tables III and IV show that three strategies achieve almost the same performance in a static environment, which implies that the RHC algorithm proposed in this paper brings very satisfactory solutions of capacity allocation in a static environment.

The real airport capacity management is a dynamic and uncertain process; neither the traffic demand nor the operational condition could be predicted precisely. Therefore, to attack the ACM problem, main attention should be paid to comparing the performance of OSA, CDO, and RHC in a dynamic environment. The main results of the corresponding simulation study are given in Tables V–VII. For Table V, it is assumed that the operational condition is fixed as VFR, and the original predicted traffic demands are given in Table I, but the actual demands in each 15-min interval may vary randomly. For Table VI, the actual demands are supposed to be the same as the predicted ones in Table I, but the actual operational condition varies randomly. For Table VII, both the actual traffic demands and the actual operational condition can change randomly. At the beginning of each 15-min interval, updated information on traffic demands and/or operational condition is available but only 80% of the predicted information is correct. To fairly assess the performance of planning algorithms in this paper, Monte Carlo simulations are conducted. The data in Tables V and VI are the average result of 400 runs of each associated case, and in Table VII, 1000 runs of each associated case.

Tables V–VII show that, RHC always gives the best solution of actual airport capacity allocation in terms of the cost, J1. OSA, due to its shortsighted performance in a dynamic environment, gives almost the worst results. CDO seems as good as RHC in Cases 5 and 6, but in Cases 7–10, where the operational condition varies randomly, its performance degrades similarly to that of OSA. Compared with OSA or CDO, RHC reduces the traffic queues by up to 10%.

Computational time is another issue when the optimization is conducted in real time. From Table III to Table VII, one can see that, in the case of a 3-hr-long operating day, OSA requires the least computational time, while CDO is the most time consuming. Since the linear programming algorithm in [2] is very efficient, all computational time in the tables can be ignored compared with the 15-min interval, but it might be crucial for an airport with a much longer operating day and a complicated airport structure.

In the simulation study above, the receding horizon of RHC is fixed as four steps long, i.e., N = 4. As discussed before, N needs to be chosen carefully in order to make a good balance between performance and computational burden. Fig. 3 shows the influence of the value N on the proposed RHC method in Cases 5–10. Since the real values of minimal J1 achieved in different cases span from 100 to 300, it would become very difficult, if not impossible, to recognize the trend of each curve. To avoid this, Fig. 3 uses the relative values of minimal J1, i.e., in each case, the minimal J1 associated with different horizon length N is divided by the minimal J1 of N = 4. With relative values, all six cases can be compared into Fig. 3 without losing the visibility of cure trends. From Fig. 3, one can see that the best performance is achieved when 3 ≤ N ≤ 6; N = 1 and N = 2 lead to quite poor short-sighted performance; and N > 6 makes the method vulnerable to uncertainties. Fig. 3 shows that the computational time almost increases linearly with N. This is mainly because
the linear programming algorithm in [2] is adopted as the online optimizer.

V. CONCLUSION

Inability of airport capacity to meet the growing air traffic demand is a major cause of congestion and costly delays. This paper introduces the concept of RHC into the problem of real-time management of airport capacity in response to the change of environment and demands. It is interesting to notice that, different from many RHC applications, stability is not a main concern in this application and terminal penalty is not necessary for achieving good performance. It is shown that the RHC provides a generic and flexible framework for developing real-time airport capacity allocation strategies according to the change of airport capacity and demands. The case study shows that, compared with other strategies, the RHC strategy can achieve comparable performance in the absence of uncertainty but better performance in a dynamic environment, with much higher computational efficiency.

REFERENCES


