Applied Intelligence 22, 183–206, 2005

Methods for Operations Planning in Airport Decision Support Systems*

JESÚS GARCÍA HERRERO, ANTONIO BERLANGA AND JOSÉ M. MOLINA
Departamento de Informática, E.P.S. Universidad Carlos III de Madrid, Colmenarejo, Madrid, 28270, Spain
jgherrer@inf.uc3m.es
aberlan@ia.uc3m.es
molina@ia.uc3m.es

JOSÉ R. CASAR
Dpto. Señales, Sistemas y Radiocomunicaciones, E.T.S.I.T. Universidad Politécnica de Madrid, Madrid 28040, Spain
jramon@grpss.ssr.upm.es

Abstract. Simulation and decision support tools can help airport ground controllers to improve surface operations and safety, leading to enhancements in the process of traffic flow management. In this paper, two planning approaches for automatically finding the best routes and sequences for demanded operations are proposed and analyzed. These approaches are integrated into a general decision support system architecture. The problem addressed is the global management of departure operations, moving aircraft along airport taxiways between gate positions and runways. Two global optimization approaches have been developed together with a suitable problem representation: a modified time-space flow algorithm and a genetic algorithm, both aimed at minimizing the total ground delay. The capability and performance of these planning techniques have been illustrated on simulated samples of ground operations at Madrid Barajas International Airport.

Keywords: airport traffic control, flow management, stochastic optimization, planning

1. Introduction

International air traffic organizations are calling for an effort to improve ground operations. When traffic demand increases beyond the available resources, congestion and ground delays occur, and both passengers and airlines suffer from disrupted services and their economic consequences. Over the previous few decades, airport capacities have become clearly insufficient to meet air traffic demand, which has been growing on a worldwide scale. During peak periods of traffic flow or when capacity falls due to bad weather conditions, demand temporarily exceeds the available operational capacity, and severe congestion occurs resulting in expensive delays for users and airlines. Often, the obvious solution of building new airports or additional runways to enlarge existing facilities is either not feasible or very limited. Therefore, there is a need for research on new ATM procedures to increase efficiency in the usage of current resources. Current control methods used at most airports rely on visual surveillance from the control tower, oral communication with pilots, or traffic monitoring and planning by human-mediated processes performed by controllers. These methods are quite inefficient when the number of ground operations increases, especially where there are complex aerodrome configurations, or when visibility falls due to weather conditions, leading to controller overload and a sharp drop in airport capacity.

Air traffic organizations, such as FAA (Federal Aviation Administration) or ICAO (International Civil

*This work was funded by CICYT (TIC2002-04491-C02-02) and CAM (07T/0034/2003 1).

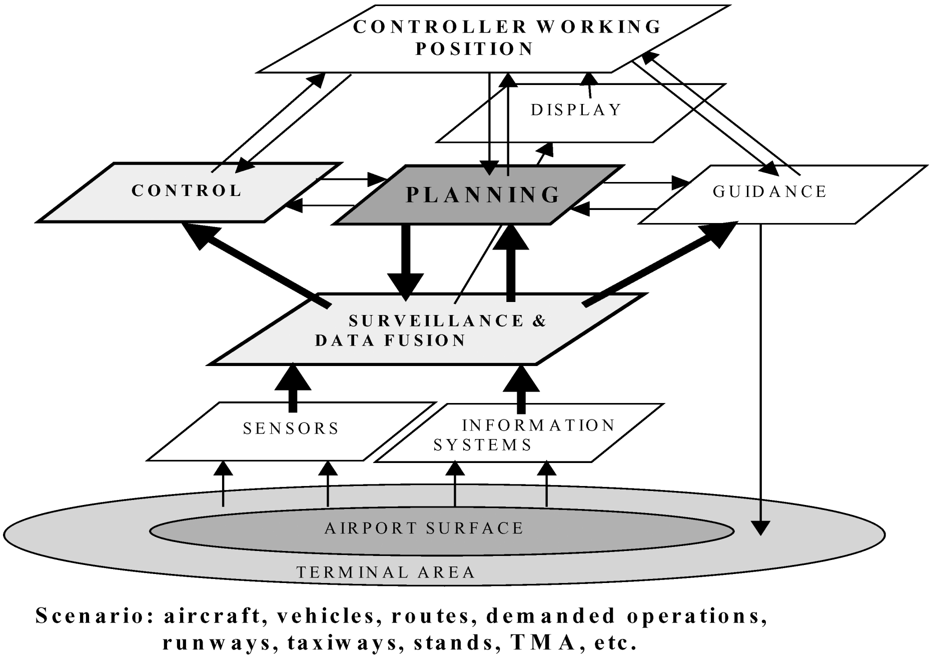


Figure 1. A-SMGCS functional architecture.

Aviation Organization), have set up some important international programs to increase the levels of automation in these areas. The set of technologies and procedures identified as support for future airport traffic management are what constitute the A-SMGCS (Advanced Surface Movement, Guidance and Control Systems, [1]) concept, whose development will be aimed at increasing safety and efficiency in surface airport operations. Basically, four interdependent functional layers have been identified as part of this concept (see Fig. 1): surveillance, control, planning and guidance.

These layers will implement independent interconnected modules, interchanging information to carry out different tasks in the traffic management process. These tasks include gathering and fusion of information from sensor and information systems (*surveillance*), semi-automatic supporting tools to detect hazardous traffic conflicts (*control*), selection of appropriate sequences and ground routes for operations (*planning*) and distribution of aids to pilots for landing and surface navigation (*guidance*). The ultimate objective of A-SMGCS development is improved efficiency in airport surface movements. Both the ICAO and FAA indicate several aids and operational modifications to be considered [2, 3]. Basically, surface traffic planning should assist controllers to manage departure and arrival operations, considering time schedules and taxiing maneuvers. The most critical situation arises when total demand temporarily exceeds airport capacity. A decision must be made on which operations should be delayed at gates (ground holding) and which routes should be assigned to minimize total delay.

A prototype A-SMGCS system is currently being implemented at Madrid Barajas International Airport, Spain's busiest airport. Our group has already researched the early steps toward automated airport traffic management, regarding sensor data fusion [4], image processing [5] and conflict detection [6]. With respect to planning, the current mode of operation is a segregated scheme, with one runway used exclusively for landings and the other for takeoffs, simply for the benefit of direct management and fixed configurations. However, a mixed mode, with runways used for both landings and takeoffs, could potentially increase available capacity, since the en-route separation between aircraft necessarily leads to lower runway usage when they are assigned for landings only. Currently, the flight plans have pre-assigned gates and runways, there are fixed routes from gates to runways depending on airport configuration, and ground controllers select the starting time to meet the time slots delivered by European Central Flow Management Unit (CFMU). An automatic scheme that would dynamically select appropriate routes and schedules for demanded departures to achieve minimum ground delay is an open line of research, considering what potential advantages it could have over current modes of operation.

The work presented here is aimed at analyzing some important aspects of the airport-planning problem, which is one of the least mature of A-SMGCS functions. Alternative techniques have been developed and integrated in a prototype system [7]: IPAGO (Intelligent Planning for Airport Ground Operations). The techniques analyzed here have been developed to improve the performance currently achieved using conventional procedures (a purely mental process performed by human controllers). The goal is to integrate the information available in A-SMGCS to automatically provide controllers with satisfactory suggestions on the complex situations that they face.

The planning problem of searching optimal time-space assignments is NP hard. A possible strategy for dealing with this complexity would be to transform the problem, using some reasonable simplifications, so that it can be solved by classical techniques, such as network flow algorithms on a space-time basis or dynamic programming. As this search is complex in computational terms, artificial intelligence techniques, such as planning or stochastic optimization [8], could also be used to allow more details concerning the problem to be addressed. If formulated in general abstract terms, the problem to be solved becomes a search problem in the possible-solutions space, where different techniques can be explored and compared.

A lot of research has been done on *Traffic Flow Management* problems, in both the communication and transport networks fields. In the case of airspace traffic, there are approaches based on temporal and spatial operations research techniques complemented with heuristics [9–12], dynamic programming [13], and evolutionary algorithms for different levels of Air Traffic Control, such as traffic assignment [14, 15], design of airspace sectors [16] or en-route conflict resolution [17].

Regarding traffic flow management at airports, there is strong interest in improving the use of available capacity. Simulation tools modeling airport operations, such as TAAM (Total Airspace and Airport Modeller) ([18]), SIMMOD (Simulation Model) (FAA) or TARMAC (Taxi and Ramp Management and Control) (DLR) [19], have been applied to analyze alternative configurations and bottlenecks at airports like Schiphol [18], Orly, C. De Gaulle, [20] or St. Louis [21]. Simulation has been complemented with data analysis to study the capacity enhancement derived from expansions or reconfigurations at airports such as DWF [22] or Newark [23].

With respect to specific techniques for planning airport operations, most approaches aim to optimize the use of runways or minimize congestion at destination airports. So, there are techniques aimed at computing appropriate landing sequences, such as [24], and schedules to assign multiple runways to landings [25] (segregated mode). Integer programming has also been applied to on-line optimize the mixed assignment of takeoffs and landings to runways depending on demand in [26, 27] and has recently been expanded to include collaborative decision-making paradigms [21, 28]. Other approaches, such as [29–31], decide the delays on the ground to solve future problems on arrival at destination airports. Finally, airport ground planning may address the details of surface operations, considering surface routes for taxiing. Some approaches search for optimal routes, considering individual operations one by one and taking the previously assigned traffic as constraints, [32], while only a few papers address the search for global solutions [20, 33, 34]. They use heuristics and genetic algorithms to explore appropriate decisions.

In this paper, two different approaches have been applied to the problem of global airport surface planning to evaluate their problem-solving capability and assess their performance. A modified flow algorithm has been developed to get the optimal flow distribution over airport segments, starting from gates and ending at runways. It calculates the maximum number of demanded operations that can be routed during a fixed planning period for a given situation of airport occupation (current and predicted) and demanded operations. This is done after modeling the planning problem as a network with time- constrained arcs. Therefore, flow distribution techniques are applied to dynamically get solutions with a deterministic (non-heuristic) optimal scheme. The drawback of this method is the loss of the individual route plans for each demanded operation, as this method considers the operations as indistinguishable flow units.

The other approach analyzed uses a stochastic-optimization method: the genetic algorithm paradigm. The problem was addressed the other way around. The number of demanded departures was fixed, and the aim of the optimization method was to minimize the time required to carry out all operations, allowing a more flexible problem representation and including specific

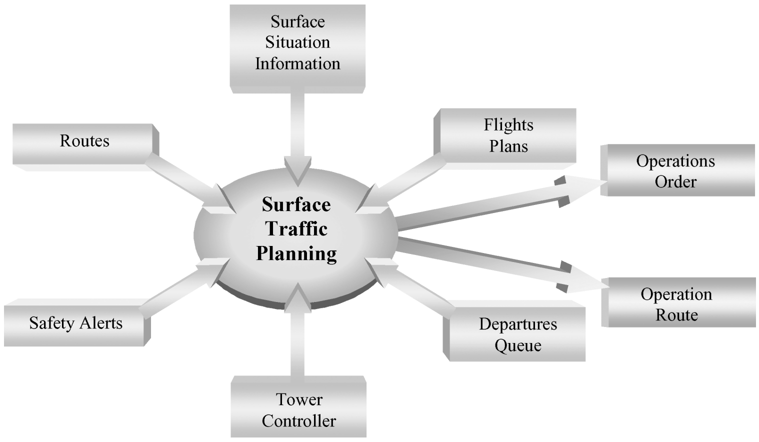


Figure 2. Inputs and outputs of surface traffic planning.

considerations for individual operations, such as assigned runway or weight category. In the final section, both techniques have been quantitatively compared under similar conditions, and we have explored what advantages automatic planning has over current methods with fixed routes.

The paper is organized as follows. First, a generic formulation of system architecture and information inputs is given in Section 2, introducing the main components of the DSS and the graph-based representation of airport status. Then, Section 3 states the problem to be solved by the airport planning function. The characteristics and specific features of the problem are compared with other traffic flow management situations, stressing the simplifications considered to develop the analyzed approaches. Section 4 details the techniques proposed for airport planning. First, we give some background on network flow algorithms and genetic algorithms, and then we detail the proposed adaptations and enhancements for applying these paradigms to the problem considered. Section 5 presents some experimental results from simulating representative scenarios on the available platform, illustrating the feasibility of the proposed techniques and goal achievement. Finally, some conclusions and further work are presented in Section 6.

2. System Architecture

An A-SMGCS planning function, generally conceived as a DSS, should integrate the elements needed to represent real situations and to implement the techniques supporting airport traffic flow management. A functional graph of the information gathered and outputs of surface traffic planning is shown in Fig. 2.

The inputs for the planning procedure are basically the operations scheduled in the departure queue and all the available airport resources to be allocated. The airport resources are the runways and all alternative routes linking them with the apron areas. They will be represented in the airport layout and affected by the currently observed situation of surface traffic and planned operations.

The architecture developed for the surface traffic planning system is shown in Fig. 3 and has three basic elements: user interface, intelligent planning system, and information gathering modules. The information gathering modules are the link connecting the core planning algorithms with the current airport situation, gathering all the necessary information from the

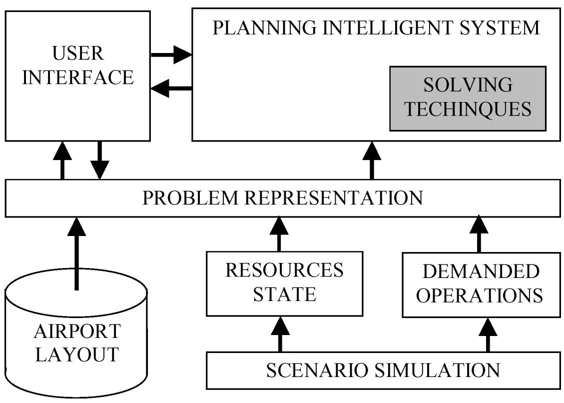


Figure 3. System structure.

available information systems. This information is integrated and transformed into an internal representation that the system uses to represent the search problem and then work out the solutions.

The basic element for representing resources, the airport layout (runways, taxiways and aprons), is loaded first to build a directed graph containing constrained capacity arcs, transit nodes and flow-source nodes. All the constraints to be considered for each planned interval should be collected and entered in this graph. Constraints cover runway status, including time slots previously allocated for other operations, the current and predicted surface-traffic situation, safety alerts and controller-imposed modifications. Graph source nodes represent the origin of the demanded operations, basically airport passenger terminals for takeoffs and landings fixes from close airspace.

The system will be illustrated by an example of traffic planning at Madrid Barajas International Airport. An internal representation of airport layout, in the shape of a directed graph, is generated first, as illustrated in Fig. 4. It depicts the two runways, three passenger terminals and taxiways linking them at Madrid Barajas International Airport.

The graph is divided into two different basic elements: nodes and arcs. A node is a reference place in the airport layout, where an aircraft can be located, representing both a waypoint in a trajectory, generally junctions between runways and taxiways or holding areas before accessing runways. An aircraft path, or route, is defined as a sequence of nodes, each one also associated with an estimated time of arrival. Nodes are linked by means of arcs. An arc has three attributes: direction, represented in Fig. 4 by arrows, cost, representing the time needed to cover the arc, and capacity. The available capacity of each arc represents the free space, where flow units (operations) may be allocated over time for each planned interval. Capacity, therefore, represents the real resources to be managed by the system. These resources have been represented as capacity vectors for each arc. This will reflect the difference between maximum capacity and planned operations. Additionally, the existing and predicted traffic for each arc reported by the A-SMGCS Surveillance

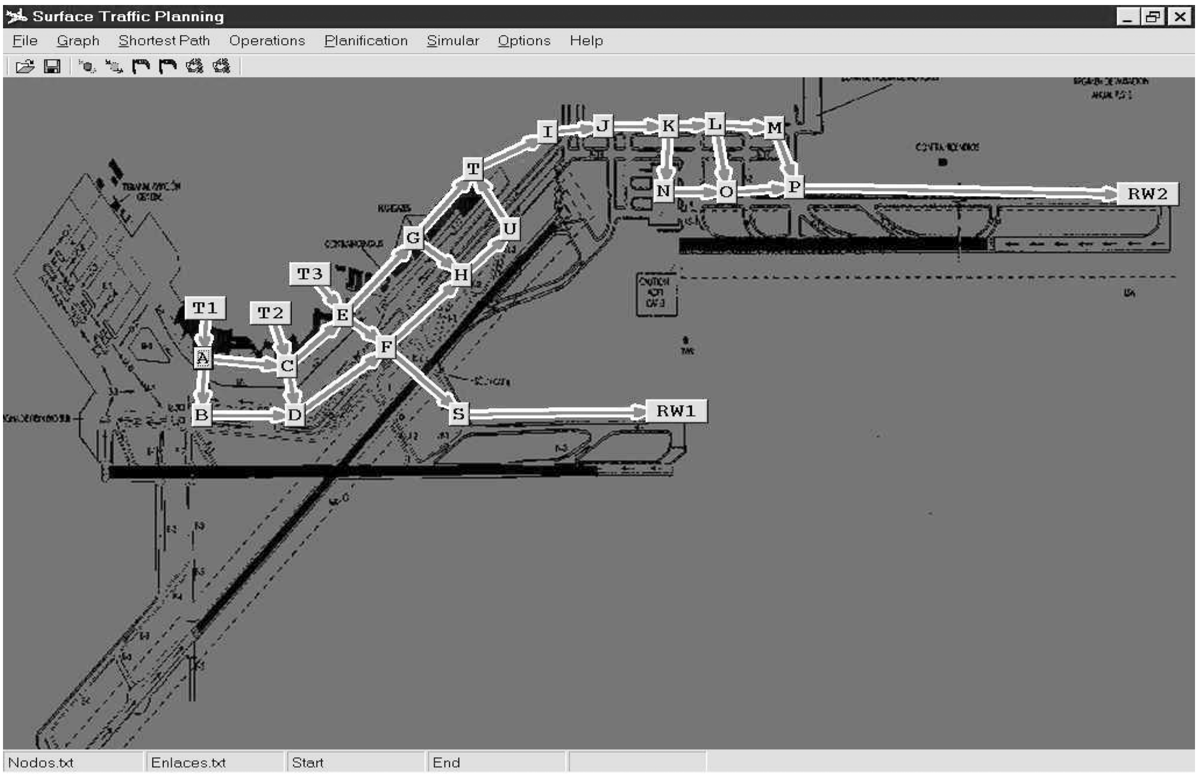
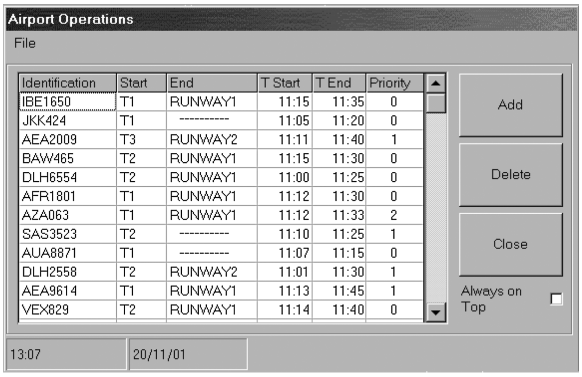


Figure 4. Graph representation of Madrid Barajas International Airport.



Airport Operations

File

Identification	Start	End	T Start	T End	Priority
IBE1650	T1	RUNWAY1	11:15	11:35	0
JKK424	T1	----------	11:05	11:20	0
AEA2009	T3	RUNWAY2	11:11	11:40	1
BAW465	T2	RUNWAY1	11:15	11:30	0
DLH6554	T2	RUNWAY1	11:00	11:25	0
AFR1801	T1	RUNWAY1	11:12	11:30	0
AZA063	T1	RUNWAY1	11:12	11:33	2
SAS3523	T2	----------	11:10	11:25	1
AUA8871	T1	----------	11:07	11:15	0
DLH2558	T2	RUNWAY2	11:01	11:30	1
AEA9614	T1	RUNWAY1	11:13	11:45	1
VEX829	T2	RUNWAY1	11:14	11:40	0

Add

Delete

Close

Always on Top

13:07

20/11/01

Figure 5. Example of departure queue.

subsystem should also be considered in the available capacity, allowing the deviations between planned operations and real-time observation to be dynamically corrected. When a planned route is occupying a segment, the capacity of this segment is decreased one unit for the intervals during which the aircraft is supposed to be crossing the segment. Explicit representation of available capacity is mainly to be used by planning techniques, although it is also useful as a graphical tool in the DSS for reporting the occupation status for all possible routes to the controller and justifying why the suggested solutions are appropriate for the current situation.

The maximum available capacity for each arc, defined as the number of possible incoming operations by time unit, depends basically on the safe minimum longitudinal separation between operations, and aircraft groundspeeds. Depending on these characteristics, specific values for costs and arc capacities in the graph representing the airport will be detailed in Section 5.

The demanded operations that are to make use of the available airport resources are represented by six attributes: operation identification, airport origin and destination points, initiation and estimated completion times, and priority. Going back to the example, Fig. 5 illustrates a possible situation with 12 operations to be planned in the next 30 minutes. These operations are entered in the departure queue. They will share the airport resources (runways and some taxiways) also assigned to some landing operations that will take place in the same time interval (and, therefore, representing constraints on the search for minimum-delay solutions).

The operations to be served have the following noteworthy feature. The origin points (terminals T1 or T2) of some operations, like the second one, JKK424, have been defined, whereas the destination points have not. This means that any destination point (in this case RUNWAY1 or RUNWAY2) would be valid for these operations. The planning procedure is, therefore, free to allocate the takeoff runway, achieving the solution with a global minimum delay. When there are not many constraints, as in this case, the system is potentially able to find more solutions and improve the global objective function (at the cost of extending the search space). As explained in the next section, the planning function is aimed at optimizing the departure sequence, assigning a route and a starting time to each demanded operation. Solutions will be a set of routes, each one represented by vectors of waypoints, referred to as nodes in the directed graph and associated with the estimated time interval for passing through. The implemented algorithms will try to optimize the maximum flow, while minimizing the average delay of operations.

Taking into account the format selected to represent the airport planning problem (both resources and operations), the core intelligent system (see Fig. 2) has been implemented using two alternative techniques to search for appropriate solutions, according to the optimization criterion, capacity constraints and demanded operations for each planned interval.

3. Airport Planning Problem

As mentioned in Section 1, the A-SMGCS planning function is aimed at efficiently managing airport ground traffic, reducing delays of operations. Like other *Traffic Flow Management* problems, it searches for the optimal allocation of resources to maximize traffic flow, where the definition of resources and flows is domain specific. Here, resources can be represented as a set of space-time positions (4-D trajectories), and the flow assignation problem has to do with how to design and assign these space-time trajectories to demanded traffic, where the solution should satisfy an optimization criterion and constraints.

3.1. Problem Definition

Airport planning should provide routes for each aircraft, considering demanded operations for both landing and takeoff. These operations share the airport resources, namely runways, taxiways and aprons (gates). Therefore, the system must sequence and time-allocate operations to minimize a global cost function: the sum of all taxiing and waiting times. Thus, the problem to be solved has basically two aspects:

- Finding routes for operations with minimum delays (taxiing length).
- Finding a sequence of operations and time schedule (allocation of time delays) to achieve optimal use of capacity.

With these considerations, the planning system should provide time-space trajectories (each represented as a series of nodes in the graph) and associated times to be assigned to the demanded operations sharing airport resources. As mentioned earlier, the system must contain a suitable and complete representation of the real problem to be solved, including the operations for allocation, the status of resources occupation, constraints, controller instructions, close airspace situation with the traffic arriving from the terminal area, etc. This internal representation will be filled in with the available information in the system, including access to surveillance function output, information systems and databases, user interface, etc. Then, planning function algorithms will generate the possible alternative solutions according to this problem representation, which will then be translated into the final actions to be taken by the controller and pilots. The planning process is conceived as part of the DSS. It must, therefore, be flexible and responsive to indications, modifications, etc., introduced by the controller through the interface. To be useful to end users, the two basic characteristics required of the searched solutions are that they must generated from a global and dynamic point of view:

- The generation of global solutions implies considering all operations to be served and the status of all resources at the same time, rather than generating particular solutions useful only for individual interests. So, a global cost function, such as the sum of all delays resulting from a solution, must be evaluated to decide the most profitable actions.
- The scheme must dynamically integrate the information gathered about the current status of traffic, operations served, and other events, such as controller instructions, conflicts, etc. So, it should be reactive to the evolution of the global status and select the best solution at any time. If anomalous or hazardous situations are detected, or there is controller interaction to indicate adequate modifications or constraints, the flow management system should dynamically adapt and find the best solution.

These requirements on the search for global solutions can be illustrated by an incremental example. If a single departure operation in the queue is to be assigned, considering an empty airport, the system would obviously output the shortest path to the closest runway, generated straightforwardly by a shortest-path algorithm, such as Dijkstra's or A* algorithms [20]. However, if this shortest path is already occupied by another operation, i.e. considering a non-empty but pre-assigned airport situation, the system must now decide between two basic alternatives: delaying the starting time until the resources are freed or selecting alternative routes for the two operations to follow at the same time. Finally, there are situations that will involve several simultaneously demanded operations competing for the same planning interval, while sharing resources with other operations in progress. The system will now have to decide their sequence, scheduled timetable and routes assigned to each operation to achieve the final goal of global minimum delay. Section 5 details some illustrative examples of problems to be solved.

So, airport traffic flow management is a planning problem with particular features. Decisions must be taken about the details of a set of operations to be served, where the control tower is a centralized position. It must take into account constraints on operations, such as separation to guarantee safety and minimum time intervals in the use of runways, and constraints on available resources, since they may be occupied by other pre-assigned operations (for instance, landings delivered by the close airspace ATC center have higher priority). If all possible maneuvers for individual trajectories were considered, the decision variables for planning would certainly be complex. By way of a simplification, the system will decide only about routes and initial delays, supposing that each operation spends any waiting time required to reach its allocated slot stationary at the gate, which is the preferred situation from the viewpoints of safety and manageability.

Going back to the example of ground operations planning at Madrid Barajas International Airport, all operations will be allocated to time-space routes or delayed until later intervals once the planning has been completed. The system displays the status of operations against time, where the position of each aircraft is indicated by its operation identification, as shown in Fig. 6.

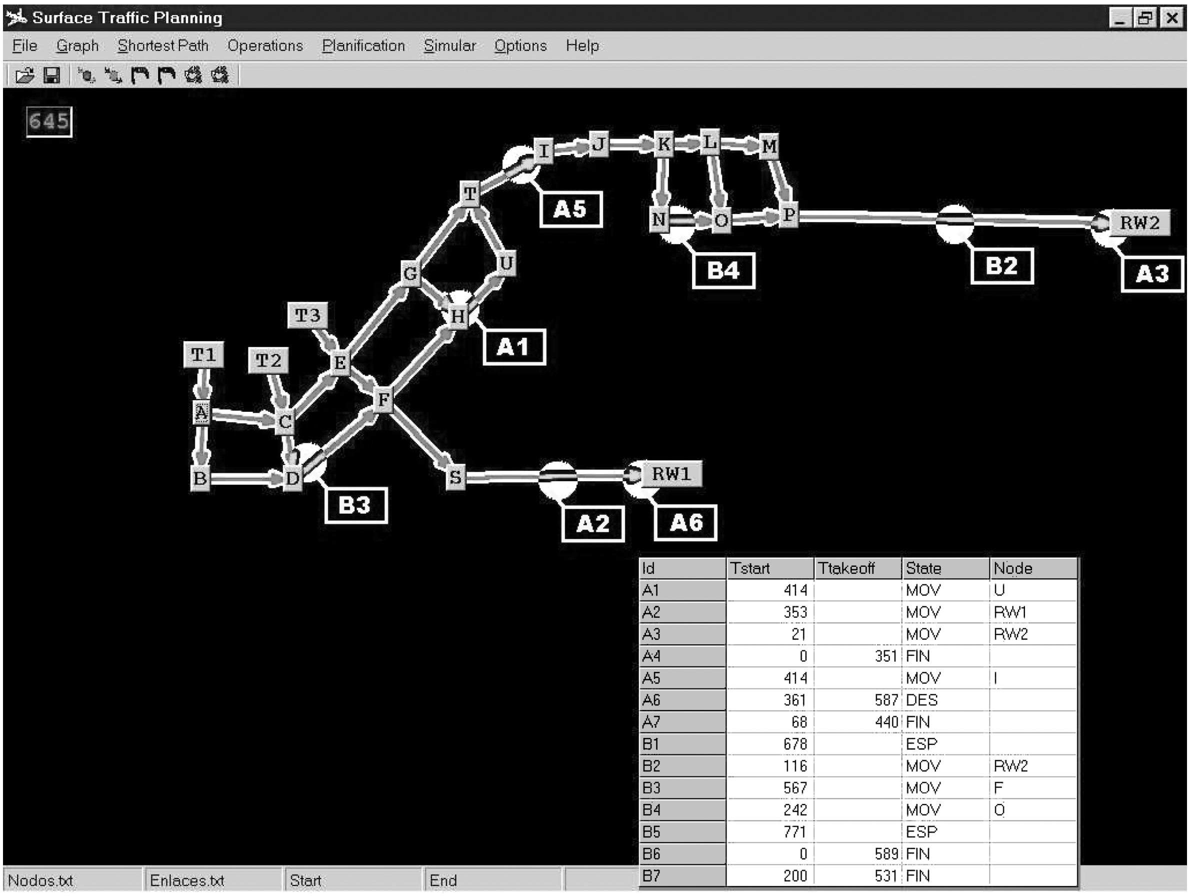


Figure 6. On-line information of surface operations.

3.2. *Simplifying Assumptions for Modeling the Airport Planning Problem*

For the planning algorithms considered here, we have developed a simplified model of airport conditions and aircraft motion on the ground. A simplified representation of the airport, operations and capacities has been selected, allowing a preliminary analysis of potential advantages derived from the introduction of a global and dynamic planning function. The main assumptions are listed below:

1. Aircraft move on ground with uniform motion between the gates and runways. The same value of average groundspeed has been considered for all operations.
2. Once the ground movement plan is delivered for an aircraft, there is no uncertainty about the trajectory it will follow. It will take the allocated route and start at the indicated time, moving with uniform motion from the gate to the runway.
3. All the delay suffered by an operation is translated to the initial waiting time at the gate, which is the preferred situation under normal conditions. The planning function will decide the starting time for an operation and, once it starts to move, it will not stop until it arrives at its destination.
4. All information about ground operations is well known in advance for the whole duration of the planning interval. It is assumed that traffic plans for operations performed during the medium-term planning period are known and that all aircraft will tightly follow their plans. This aspect is not in conflict with the requirement of dynamic reaction for planning, since the continuous operation of algorithms will be renewing information about predicted scenarios throughout this time interval.

Therefore, summarizing these assumptions, all the information about airport surface occupation and demanded operations for allocation is known in advance for a centralized function to decide plans, and information is continuously renewed in time to allow dynamic search of best decisions against time.

So, the tool suggests the optimum solutions under these conditions, which can be helpful for suggesting to the controller which the promising alternatives are. To be a completely useful tool in operational conditions, some further steps would be needed. The most important aspect would be a flexible interface to continuously re-define the problem, taking into account modifications made by controllers. Deviations in operations with respect to allocated plans could be considered to correctly model the available resources and then decide appropriate plans for reaction. Changes in plans should, as far as possible, rule out "jumps". To do this, only the deviated operations should be considered as variable decisions, not moving the other already assigned plans. After identifying a deviation, the other assigned operations can be considered as fixed constraints in the available capacity, and then the affected operation can be re-designed. The exception is when, due to a deviation, some allocated operations violate constraints, in which case they must be also considered for re-planning.

Finally, more details about the operations should be included in the models, such as holding positions at taxiing or before takeoff, variations in groundspeed, or variable time separations in runways depending on weight categories. Finally, the uncertainty in maneuvers and speed throughout the planned time could be considered in the generated plans.

4. Techniques for Ground Planning

Having represented the airport planning problem in a directed graph format as described in previous sections, here we analyze some applicable techniques to generate the solutions, according to the pursued goals, demanded operations and constraints. Then, the finally proposed schemes are detailed.

In the first place, the above-mentioned simplifications have transformed the airport- planning problem into a flow management problem so that we can apply classical network flow algorithms, now extended to cover a space-time search space. The basic approach in this case has been to fix a planning time period, represent the demanded operations as flow units, and search paths in the graph to achieve a maximum flow with minimum transit delays. This must be done considering all operations for planning at the same time and their possible schedules against the available time-varying capacities. In this case, the required minimum separations and assumed aircraft groundspeeds have been translated to arc capacities, as indicated in the previous section, fixing the flow constraints to be handled by these algorithms.

The second approach considered in this paper involves running an explicit search using an artificial intelligence technique based on stochastic optimization: the GA paradigm. To do this, the routes and time schedules for all demanded operations are represented as decision variables in a constrained space, where the minimum separations between aircraft are explicitly modeled and the optimum solution is searched.

However, the two techniques do not work in exactly the same way. The GA approach directly includes the individual operations in the encoded problem. The solution, therefore, refers to each operation: allocated route and schedule. This does not apply to the flow management algorithm, MCMF, where operations for allocation are first abstracted as flow units. This will deliver the optimum flow distribution, from which the individual operations should be extracted later. Therefore, the designed MCMF algorithm has the advantage of finding the optimum distribution, but only when operations are not distinguishable: it takes the number of operations from each terminal and selects how and when to deliver them to runways in order to minimize the sum of delays. However, only the GA technique can consider limitations concerning individual operations. For instance, some operations may be constrained to depart only from a certain runway. Therefore, it should consider only routes from the departing terminal ending at that runway. Another constraint may be a time separation depending on the specific weight category of aircraft, which can be considered only when individual operations are analyzed.

Finally, any technique applied within this framework could consider a set of priority levels. The simplest possibility is iterative algorithm operation, each iteration dealing with one level of priority and assigning the operations in this set, and leaving these allocations as constraints for the next iteration. One possibility would be to increase the priority of operations delayed during the last planning interval.

In the following, the bases for both approaches are summarized and then each of the proposed approaches is detailed.

4.1. Classical Flow Algorithms

The algorithms for flow management on networks come from the field of operations research [35,36], specifically from optimization techniques applied to integer-constrained linear programming. They are well-known methods for outputting optimal routes and flow distributions in networks under stationary conditions (all flows are characterized with constant values or long-term statistics).

The basic structure handled by these algorithms is a directed graph (V, E), where V is a set of nodes and E is a set of directed arcs or edges linking the nodes. The network is able to move some commodity along the arcs, where the flow is defined as the quantity of commodity moved per time unit. A positive-valued real variable, x_1, is defined containing the flow distribution for all arcs in the network, $1 \in E$, according to the direction defined by each arc. Each node N in the graph is classed as one of three possible types, depending on the flow balance of arcs leaving and arriving at the node, b_N: **source**, if $b_N > 0$, **sink:** when $b_N < 0$ and **transit:** when $b_N = 0$.

The main results available for network flows are for a simple type of network, referred to as a basic network, characterized by two properties:

- There is a single source, S, and a single sink, T.
- For every arc, there is a positive number called capacity, c_1, defining the maximum flow that can be assigned ($x_1 \leq c_1$).

Besides, when there are defined costs per flow unit for each arc, d_1, we have a weighted basic network. As we will see later, the assumption of basic networks is not a severe constraint, since there are simple transformations that can be applied to more generic networks to convert them into basic networks [35]. The three main results from network flow algorithms that will be considered are briefly presented below.

4.1.1. Maximum Flow Algorithm. This problem involves searching the flow distribution in the network arcs, x_1, that maximize the flow between S and T and satisfy all network constraints (the same flow leaving S arrives at T, the flow balance in transit nodes is zero, and the condition $0 \leq x_1 \leq c_1$ is satisfied by all arcs $1 \in E$).

A classical algorithm, taken from Ford and Fulkeson [36], computes the maximum flow in the network by incrementally increasing, whenever possible, the flow along augmenting paths. It works with an extension of the edge set, defining, for each occupied arc, another in the opposite direction with as much capacity as the assigned flow in the original direction. This allows a backtracking mechanism to find new flow-augmenting paths by re-allocating previous routes and increasing the flow in the network.

4.1.2. Minimum Cost Path. This problem involves searching the route with minimum cost between S and T to send a flow unit: minimize x_1, $1 \in E\{\sum_{1 \in E} d_1 x_1\}$, satisfying the same network constraints as above.

An optimal and efficient algorithm for this problem is Dijkstra's algorithm [36], which computes the solution in $O(m \log n)$ time, where n, m are the number of nodes and arcs, respectively. Unfortunately, Dijkstra's algorithm is only applicable when all costs d_1 are positive. A solution for the general case with positive and negative costs is the Bellman-Ford algorithm [37], which finds the solution in $O(nm)$ time. As we will see next, although the original airport graph has positive costs (the time needed to cross each edge), the transformation of Ford-Fulkerson's method to discover flow-augmenting paths introduces arcs with negative costs, which rules out the application of Dijkstra's algorithm.

4.1.3. Minimum-Cost Maximum-Flow (MCMF) Algorithm. Finally, a combination of the algorithms solving the two above-mentioned problems, maximum flow and minimum cost, can send a certain amount of flow, F, between source and sink nodes in a basic network, at minimum cost. Besides, if the flow quantity F is increased until the network is fully saturated, the problem addressed is then the delivery of maximum flow between source and sink at minimum cost (Minimum-Cost Maximum Flow, MCMF [36], algorithm). The MCMF steps are as follows:

- *Step 0*. Find the shortest path between source S and sink T and send as much flow as possible.
- *Step 1*. Find the shortest path between S and T, considering an expanded network. Non-saturated arcs have the original cost, saturated arcs have infinite cost, and for each arc with flow higher than zero, a fictitious arc in the opposite direction and with negative cost is considered.

- *Step 2.* Send the maximum possible quantity of flow along the shortest path found in step 1. For fictitious arcs included in the path, the allocated flow will be subtracted from the respective original arcs in the original directions.
- *Step 3.* Repeat steps 1, 2 until there are no more unsaturated arcs available to find new paths.

Therefore, this algorithm alternates steps to find flow augmenting paths in the constrained network and to select those with minimum cost. Step 0 is the first action for initialization, which is equivalent to step 1 before having sent any flow unit. The algorithm iterates until no more flow units can be sent from S to T, so it achieves the maximum flow through the network with minimum cost.

4.2. *Proposed Planning Algorithm Based on Extensions of Classical Flow Algorithms*

The main goal here was to extend the classical network-flow algorithms described in 4.1 and adapt the airport problem representation to develop the desired solutions according to the formulation stated in Section 3. The goal is to find routes and schedules for the required operations (flow units), achieving an optimum usage of available capacity, adapted to dynamic conditions. The enhancements proposed here are two-fold. First, the introduction of time scheduling for operations (decision on time slots) in the search space variables of flow management algorithms, extending the dimensions of variables usually handled (which do not consider dynamic variations in flows or capacities). Secondly, the application of some basic transformations to the graph representing the problem so as to address important practical issues, such as deciding the initial operation delays, assigning multiple sources to multiple sinks and explicitly considering intersections. The decided plans allocated to the demanded operations will dynamically depend on the airport conditions (available capacities of runways and taxiways during the planning period, other operations required for the same period, etc.).

So, the decision variables (flows for each arc) will consider the time dimension. To do this, both allocated flows and available capacities have now been represented by vectors, with as many components as time units considered for the planning interval. The MCMF algorithm has been reformulated for a representation of flows and capacities with N components, corresponding to N time intervals considered for planning: $x_1[k], c_1[k], k = 1, \ldots, N$.

The MCMF procedure is now applied considering time dependence. To allocate a flow quantity to an arc, the occupied time interval is first computed and then compared with the available capacity in the respective interval. The key aspect for calculating the respective time interval (index of vectors) is the assumed constant-speed motion with known mean value for aircraft groundspeeds. Once this correspondence between node positions and time intervals has been defined, the basic structure of the MCMF algorithm (Section 4.1.3) is unchanged.

This extension of the MCMF algorithm, together with the explicit representation of waiting nodes in the graph to be considered in the route-decision variables, will allow us to find flow vectors for allocation. So the sequence and schedule of operations have been naturally included in the flow optimization process. The examples presented in Section 5.1 illustrate how the solutions handle time and space criteria to search the optimum distribution of demanded operations: those achieving a maximum "packing" of departures.

The other aspects covered before applying it to airport planning are the above-mentioned transformations of the graph representing the airport: multiple sources and sinks, delay nodes and intersection nodes, which are briefly presented next.

4.2.1. Multiple Sources and Sinks. Multiple sources and sinks (representing, for instance, the allocation of operations coming from several terminals to several runways) can be solved by a simple transformation [36]. Original sources and sinks are transformed into transit nodes connected to two unique "super-nodes": SS, including all source nodes, and ST, including all sink nodes. The edges linking original sources and sinks with super nodes have zero cost, and capacity equals the number of demanded operations in the case of source nodes or infinity capacity in the case of sink nodes. Figure 7 presents an example where a network with three sources and two sinks has been transformed into a basic network. Arc capacities are specified in Fig. 7 between brackets, while the costs are entered in squares drawn on the arcs.

4.2.2. Waiting Nodes. To account for waiting periods in the generated routes, each source node will be linked to a series of new nodes in the graph: the waiting nodes. There will be as many waiting nodes as there are time

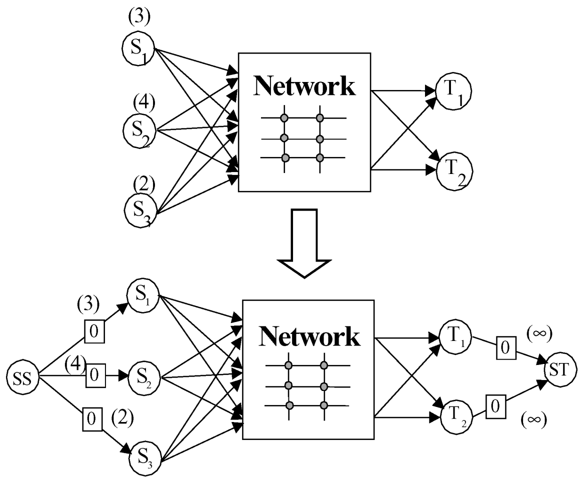


Figure 7. Multiple sources and sinks.

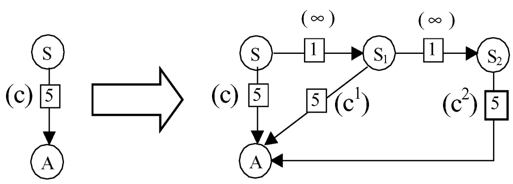


Figure 8. Introduction of two waiting nodes: S_1, S_2.

intervals considered for delay (if an operation is not assigned after considering all possible waiting nodes, it will be delayed until the next planning interval). The edges connecting waiting nodes with each other (for instance, go from status with a one-interval delay to status with a two-interval delay, and so on) have unlimited capacity, but cost equals the number of delayed time intervals. Arcs connecting new waiting nodes with the other original nodes in the network have the same cost as the original ones, but their capacity vectors will be shifted to the left along the time axis as many units as the time delayed. Figure 8 shows an example, with the original nodes on the left-hand side (a source node S connected to transit node A), and the introduction of two waiting nodes on the right side.

$$
\begin{aligned}
c &= [4, 4, 2, 2, 3, 3, 4, 4, 4, 4]\\
c^1 &= [4, 2, 2, 3, 3, 4, 4, 4, 4, 4]\\
c^2 &= [2, 2, 3, 3, 4, 4, 4, 4, 4, 4]
\end{aligned}
$$

Two delay nodes are included, S_1, S_2, each delayed one time unit. Therefore, the cost for each arc is 1 and the original vector capacity, c, is shifted 1 and 2 units to left.

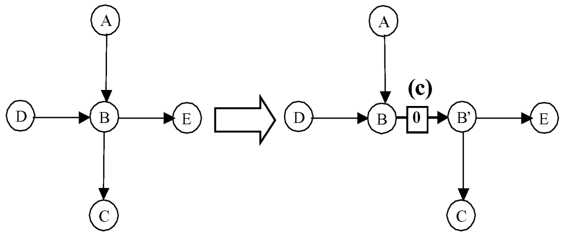


Figure 9. Constrained-capacity intersection node.

4.2.3. Intersection Nodes. Finally, it is interesting to note that the computation of the available arc capacities explicitly considers minimum separation, but affects only longitudinal separation. Depending on velocity, the longitudinal separation is defined by the maximum number of operations that can enter an arc in a time unit. However, the number of operations crossing a node sometimes needs to be limited. For instance, the number of operations that can access a junction area where several routes may intersect needs to be fixed to avoid separation conflicts. This can be done by a simple transformation, introducing a fictitious new transit node and arc with zero cost, but bounded capacity (note this is the opposite situation to waiting nodes). An example is given in Fig. 9, where node B is expanded with fictitious node B' and a new arc with zero cost and constrained capacity (for instance, the same as the other taxiway arcs).

The same transformation could be applied in other situations if the model were further extended. For instance, we could identify nodes in a path with special capacities, such as holding areas at the end of taxiways or areas in front of runway departure zones.

4.3. *Stochastic Optimization*

Network problems are one of the earliest applications of a kind of stochastic global optimization technique namely evolutionary computation [8]. Genetic algorithms (GA) search the space of combinatorial solutions, providing fast and accurate solutions. In the field of transportation management, and particularly air traffic management, the work developed by [14–16, 20, 24, 33] proposed a GA to improve some aspects of ATM. In this paper a GA, inspired by the abovementioned work, incorporating an ad hoc mutation operator and fitness function, schedules the demanded operation.

4.4. Genetic Algorithms for Operation Planning

In this paper, the algorithm described in [38], namely the canonical genetic algorithm, was applied to generate the surface movements plan. The canonical GA was chosen for its simplicity and because it is a generic procedure, which is desirable when there is not information about the structure of the search space. We defined a plan as the departure schedule and the paths that a set of aircraft follow from gates to takeoff runways. The objective is to find the plan that reduces the average delay per operation, subject to the conflict-free operations constraint.

J. Holland formally introduced genetic algorithms (GA) [39], and their characteristics have made them widely applied in optimization problems, especially to combinatorial problems. The procedure of searching the solution provides some important characteristics like robustness and parallelism, but their use is inadvisable if the search space is small or global optimality is required. The planning of the surface operations at an airport, formulated as a combinatorial problem, has a very large solutions space and an approximate solution should be good enough. Thus, the use of the GA paradigm is justified because the trade-off between the quality of the solutions and processing time is advantageous.

The three most important aspects of using GA are the definition and implementation of:

- The genetic representation. Each solution is coded as an instance of the vector of the decision variables. This codification is called the "genotype" of a solution.
- The objective function. The criteria to measure "the goodness" of a solution are typically implemented by means of a function, namely a "fitness function". The fitness function is applied to convert the genotype into a phenotype.
- The genetic operators. The search space is explored and exploited by applying three operators that produce new solutions starting from preexisting ones.

4.4.1. The Genetic Representation. For GA, an air traffic ground plan, $P = (\vec{r}, \vec{t})$, is codified with two sequences of numbers of length equal to the demanded departures, d.

$$\vec{r} = (r_1, r_2, \ldots, r_d)$$
$$\vec{t} = (t_1, t_2, \ldots, t_d)$$

For each i-th operation, a plan allocates a route, r_i, selected from a predefined set of all possible routes, and the time, t_i, that the aircraft will delay its departure from the gate. This special codification provides an easy implementation of the crossover and mutation operators, adapted to the characteristics of the problem. Restriction of unfeasible solutions, such as operations allocated to the same route at the same time, cannot be taken into account in the codification, the fitness function will penalize the solutions that violate the constraints. The codification only restricts r_i and t_i to valid values, r_i in the range of possible operations and t_i as an integer between 0 and the maximum delay.

4.4.2. The Objective Function The fitness value measures how a solution solves a problem. The solution schedules the demanded departures and allocates a path for each one. Demanded operations are registered as a list of two-dimensional vectors, (g, p), containing the departure gate and the takeoff runway. The gate is always assigned but the takeoff runway may or may not be not specified. Figure 10 shows an example of the allocation of operations with a plan. We find that the order of the departure schedule starts with operations BAW465 and DHL6554 and the operation AFR1801 starts moving along the allocated route after a twenty-second delay. The last operation, JKK424, routed through path 38, has a five-minute delay.

To calculate the fitness values, the surface movements are simulated according to the assumptions listed in Section 3.2. The following quality measures are assessed and used as parameters of the following fitness function:

$$f = o \cdot t_o + w \cdot t_w + t + 50c - 50k + r$$

Solution			List of Demanded Operations		
$\vec{r}$	$\vec{t}$		Identification	Gate, g	Runway, p
24	66	⟷	IB1650	T1	1
38	297	⟷	JKK424	T1	1
8	208	⟷	AEA2009	T3	2
20	0	⟷	BAW465	T2	1
7	0	⟷	DLH6554	T3	1
15	20	⟷	AFR1801	T2	ANY
38	211	⟷	AZA063	T1	2
20	174	⟷	SAS3523	T2	1

Figure 10. Operations demanded and solution.

The terms are defined as follows:

- Number of incorrect origin gates and destination runways, o, w. When a route is allocated, the gate or the destination runway could be wrong, and not matchthe requirements for the operation. Any restrictions on the permitted destination runway or origin gates are included in the time intervals concerned for as long as these restrictions are violated.
- Time to carry out the whole plan, t. The simulation finishes when the last aircraft has taken off. The optimization process will tend to minimize this total time needed to carry out all operations.
- Number of conflicts, c. When two aircrafts violate the safety distance, a conflict is reported. Obviously, any plan containing just one conflict is unacceptable. Therefore, this parameter is strongly weighted. By the twentieth generation or so, the best plan has no conflicts.
- Number of takeoffs, k. The objective is to get plans that process all demanded departures.
- Average delay per operation, r. The time that an operation is delayed must be as short as much as possible.

The GA is designed to minimize the fitness value of solutions. This technique transforms optimization problems into search problems, subject to the requirement of incorporating the domain constraints into the fitness function. Therefore, the solutions to this problem are searched by relaxing the constraints and including penalties for the solutions that do not satisfy the constraints. All constraints are global, since they depend on the relationship between pairs of individual solutions. This is the usual methodology in similar constrained problems (see [38]) and is more efficient at finding appropriate solutions than a direct encoding with constraints.

The parameters of the GA applied in this paper are summarized in Table 1.

Table 1. GA parameters.

Population size	200
Ending criteria	Generations = 200
Selection	Tournaments of size 4
Crossover rate	100%
Mutation rate	1%
Time Decrement Mutation rate	5%

4.4.3. The Genetic Operators. The main idea behind GA performance is cumulative selection. This in itself, however, does not provide a full explanation of the question of why it works. Cumulative selection is by no means a new concept. It appears in stochastic optimization and other similar descent gradient methods. The innovation is the incorporation of inheritance of characteristics and the variation triad. These features resemble, in a simplified form, biological natural selection. The genetic operators: selection, crossover and mutation, implement these features.

Because the operator must be adapted to a particular problem, many genetic operators have been reported in the literature. In this paper, tournaments selection [38] was the selection scheme chosen for selecting the individuals in the population that reproduce to generate the next generation. This selection is scheme is appropriate for wide search spaces where it is likely to find solutions with similar properties, so it is important avoiding premature convergence. Tournaments selection gives more opportunities to explore solutions with worse initial fitness, with the cost of slower convergence. The crossover operator produces new solutions by recombining existing ones. In this paper, a single-point crossover has been used. The crossover operator had to be modified because plans are coded as two sequences of numbers. The same crossover point is applied to both parts of two parent plans to engender the two offspring.

Two mutation operators are used in this paper. One is traditional mutation described in the canonical GA, and the other was included to decrease the delay time of the operations. A random variation of the delay time, uniformly distributed in the range $[-8, 2]$, is applied with 5% probability. The range of variation is skewed towards negative values, for the sake of favoring the solutions with small delay time. This new mutation operator was included to speed up the appearance of small delay-time solutions, and these parameters were tuned after exploring different experiments.

Figure 11 describes the main steps of the algorithm for generating a new population of solutions.

5. Experimental Results

In this section, we present results after applying the two proposed planning techniques to scenarios generated by simulation. They have been analyzed using the airport represented as a directed graph as defined previously. The platform tool, IPAGO, can represent and

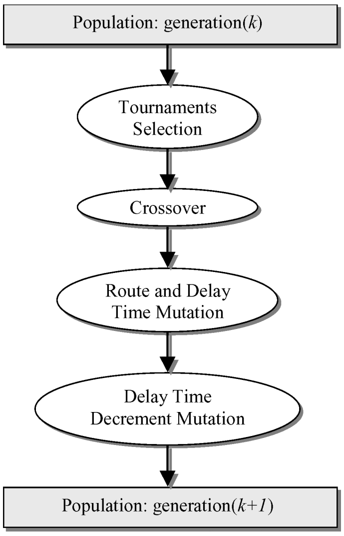


Figure 11. Steps of GA algorithm.

analyze the planning problem and is able to simulate representative airport scenarios, although the system is conceived to be finally connected with real airport data networks, access the necessary information, and then provide solutions to real scenarios. In this case, simulated departure operations have been used by both techniques to find solutions containing the best sequence, routes and schedule for the operations. As mentioned above, MCMF and GA algorithms are not always comparable, because MCMF provides a benchmark for situations where operations can be indistinguishable. Only the GA technique can deal with flights where limitations, such as pre-assigned runways, apply. The results discussed in this section are mainly concerned with situations where both algorithms can be applied to generate the solutions.

The graph representing Madrid Barajas International Airport (Fig. 4) was considered, including 24 nodes and 29 arcs. The capacities and costs of the constrained arcs linking the nodes were adjusted assuming the following parameters for motion on the ground and operational procedures:

- *Taxiing:* average speed of 10 m/s and minimum longitudinal separation of 200 m. The junction areas of crossing taxiways have also been characterized with a minimum spatial separation of 200 m between aircraft by entering additional arcs in the extended model for network flow algorithms (Section 4.3.3).
- *Runways*. The runways are the ending points of departure operations (flow sinks) and require a one-minute separation between consecutive takeoffs.

With these parameters, the costs of all arcs located between gates and runways, assessing the time needed to cross them, are directly computed as their spatial length divided by average speed. This can be done by assuming that aircraft move at constant speed and do not stop once they have started their trajectories (as indicated in the model assumptions listed in Section 3.2). Regarding arc capacity, this depends on the type of area they represent. For runways, due to the time constraint between consecutive takeoffs, one flow unit can go across a runway node per minute. Since we have selected a time unit of one minute to represent flow and capacity vectors, the maximum capacity of an arc linking a runway and the sink node is 1. For taxiways, the maximum number of operations, moving with an average speed of 10 m/s, that can cross a node respecting the specification of longitudinal separation of 200 m is 10/200 operations per second. With the selected time unit of one minute, the capacity of arcs linking taxiway segments is 3 operations per minute.

For example, the window at the bottom of Fig. 12 shows the nodes connected to node $F(C, E, P$ and $G)$, and the available capacity along each path during the next 30 minutes.

Capacity is represented by a gray scale in the graph. By way of an example, ithere are some dark segments in the arcs representing access to runways, RUNWAY1 and RUNWAY2, in Fig. 4. They indicate that some time slots have already been assigned for landing operations to be carried out in the following 30 minutes. Therefore, these runways cannot be used for the takeoff operations to be planned in this interval. Once the solutions have been generated for the demanded operations, the controller can select the operation to display detailed information about a specific route. This information with routes and schedules for each individual demanded operation is extracted from the solutions generated by the algorithms and then presented to controllers in the DSS interface.

In the following sections, both schemes are first applied to different scenarios with an increasing number of departure operations for allocation, all of which can be allocated to any runway. The solutions proposed by both techniques are compared by means of the details of space-time flow distribution. Performance is summarized as the ratio between total delay and number of

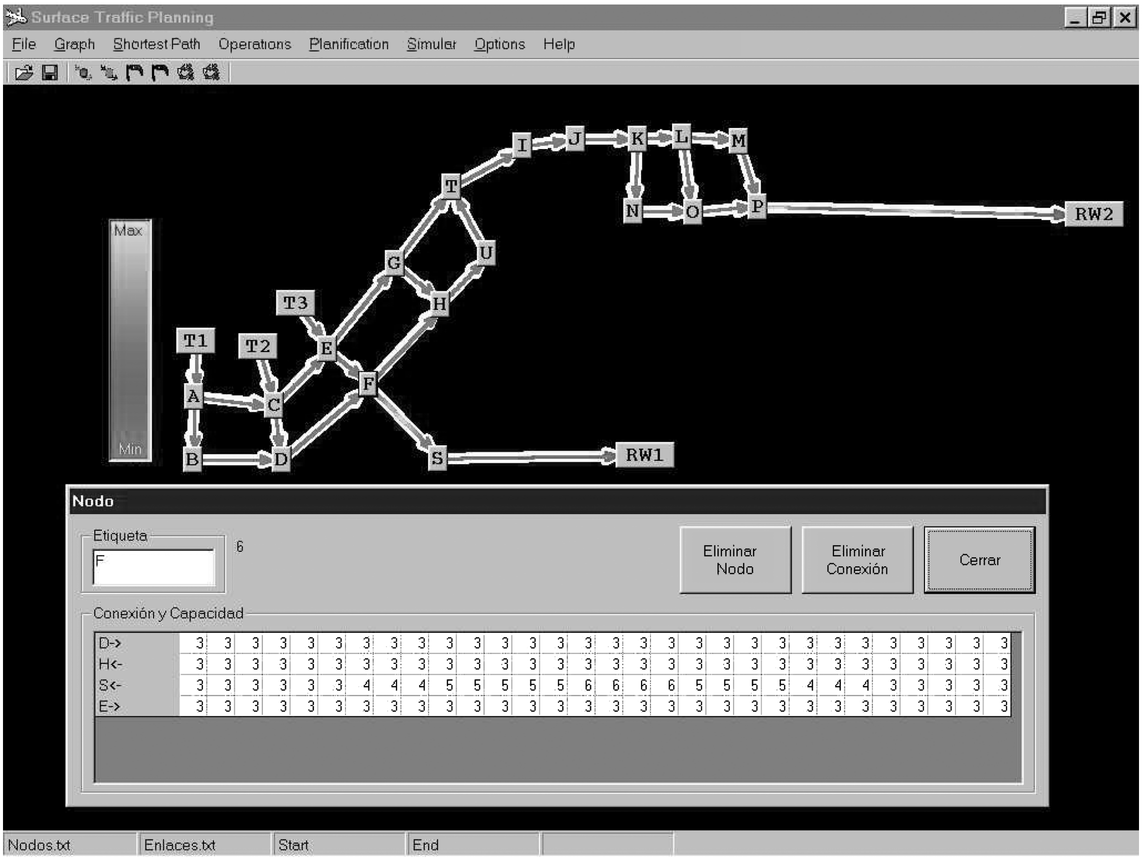


Figure 12. Capacity of arcs.

departures demanded for both systems. Finally, an analysis is presented comparing solutions achieved with automatic planning and solutions generated using a fixed-routes procedure, simulating planning similar to what human controllers currently produce using operator rules.

5.1. *Capacity and Flow Analysis*

Both the flow allocation and GA algorithms were run under several conditions, varying the number of demanded operations and which gates were used to start departures (sources of demanded departure operations). In these experiments, the airport was first taken to be empty (all arcs with full capacity available) and flow units were allocated in a single planning horizon of 20 minutes.

Figure 13 illustrates a situation with six demanded departures, two from each airport terminal (T1, T2, T3). The time distribution of solutions provided by both techniques is shown. The MCMF solutions are shown on the left and the GA solutions on the right. The time distribution of operations as they cross the segments in the airport graph is indicated for the 20 minutes considered for planning. Only the nodes in the airport graph representing the sources (airport terminals) and sinks (runways) have been depicted. From Fig. 13, we can see that both the algorithms decide to start the two operations from the nearest terminal (T3) in the first minute (although they must be at least 20 seconds apart to guarantee a distance of 200 m). MCMF chooses to start an operation from T2 at the same time. One and two minutes later, it starts an operation from terminal T1 and another from terminal T2, respectively. Finally the second operation from terminal T1 is delayed by at least two minutes. The GA starts operations at terminal T1 at minutes 1 and 3, while operations from terminal T2 start at minutes 2 and 4. As we can see on the left-hand side (MCMF solutions), five operations are routed to

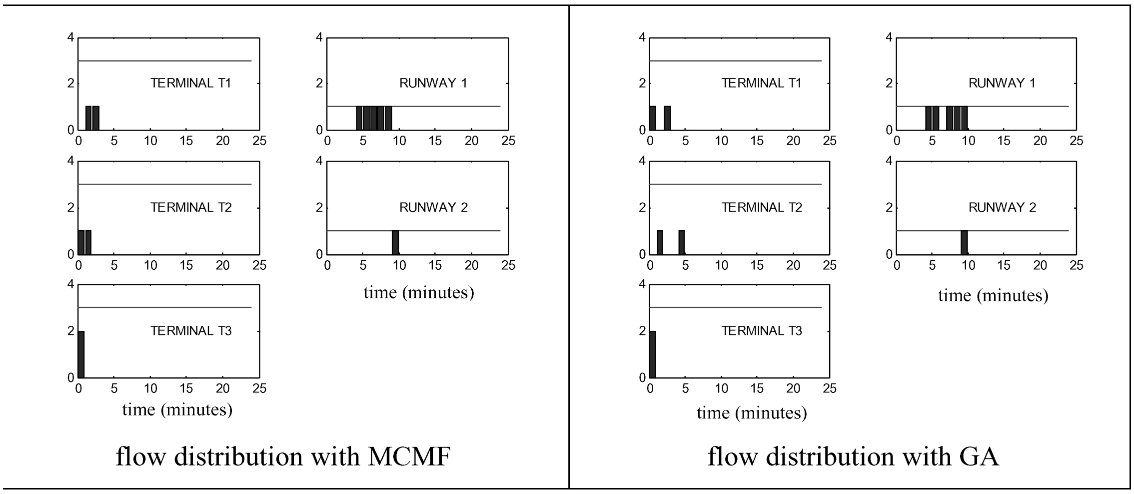


Figure 13. Flows with six allocated operations (terminal and runway nodes).

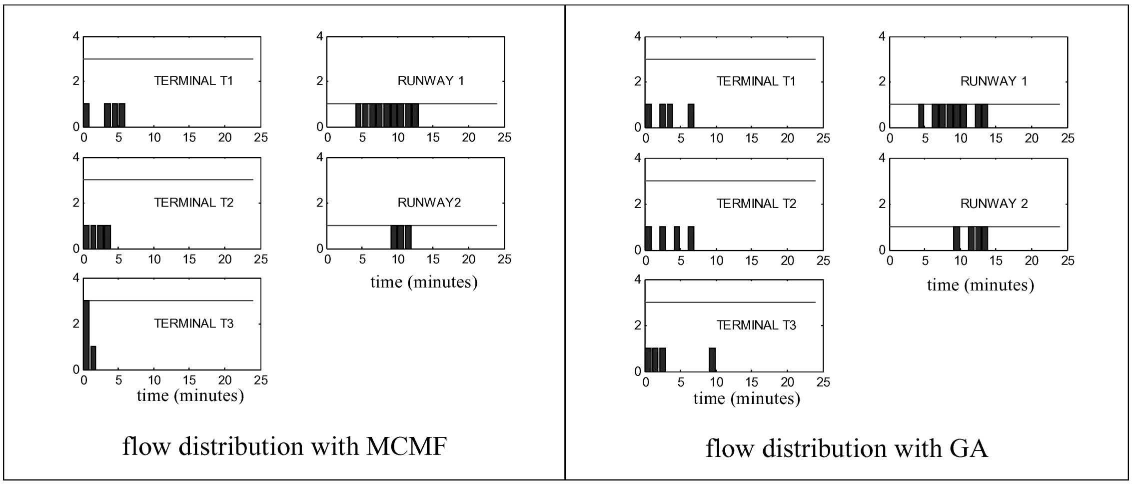


Figure 14. Flows with 12 operations assigned (terminal and runway nodes).

runway RW1 and one to RW2 to guarantee that the objective of minimum total time (period within which all demanded departures are completed) is achieved. In this case, the six operations takeoff within ten minutes after the starting time of planning. The first takeoff at RW1 starts at minute 5 and the first at RW2 at minute 10. These are the minimum time intervals needed to arrive at these nodes with the assumed aircraft groundspeeds. Looking at the GA solutions (right-hand side), they also end at minute 10, although there is an extra one-minute delay for the last three operations at runway RW1.

Figure 14 shows the case with 12 demanded operations, four from each terminal T1, T2, T3. The MCMF system again first serves all the operations from terminal T3, closest to runway RW1, and selectively delays the rest. The objective is again accomplished, as shown by the runway occupation figures, with a compact sequence of takeoffs achieving efficiency in the use of airport capacity. All operations are served within 13 minutes from the start of the planning interval. The GA solution, on the right-hand side, is a bit sparser, with some unused capacity and achieving a total time of 14 minutes. Occupation appears to be a bit tighter “packed” in the flow algorithm. This result is to be expected, as the GA provides a non-optimal solution.

Finally, Fig. 15 illustrates saturation cases, where there are many operations from all terminals and the

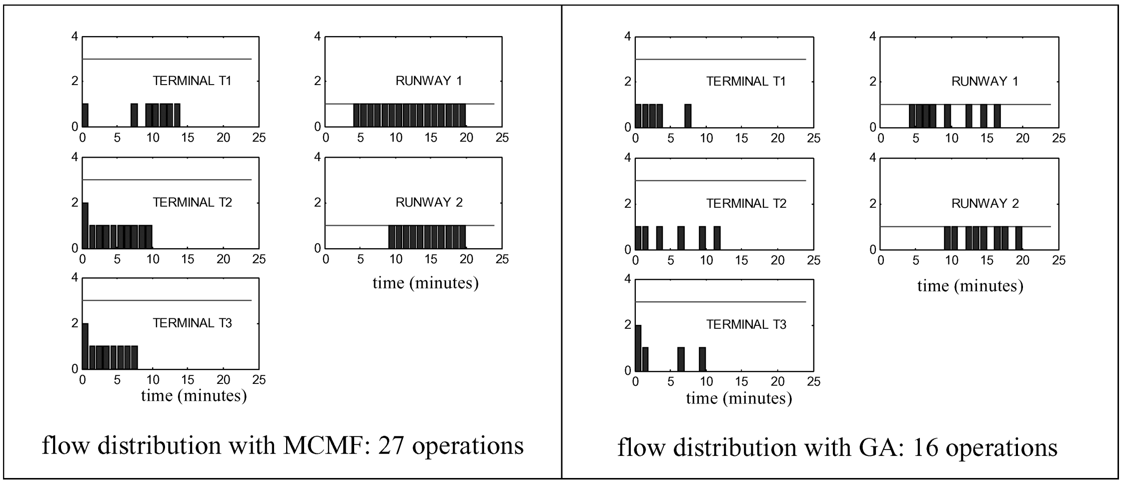


Figure 15. Flows with maximum operations allocated under saturation (terminal and runway nodes).

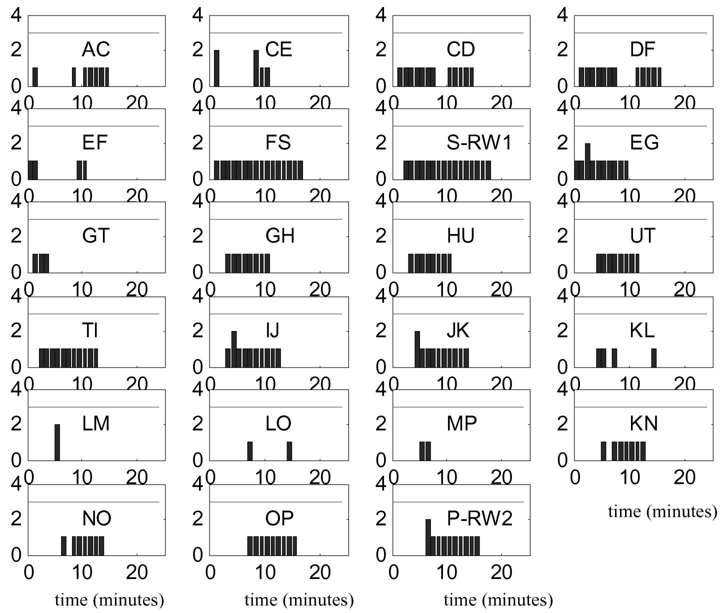


Figure 16. Flows with MCMF under saturation (detail of taxiway nodes).

system decides how many of them will be served in a 20-minute planning time period to minimize global delay. As we can see on the left-hand side, the MCMF algorithm decides to start 9 operations from terminal T3, 11 from T2 and 7 from T1. The system again achieves optimum usage of runway capacities, with a continuous occupation for each runway starting at the minimum time instants needed to reach the runways from the gates. Therefore, in this situation and assuming the above-mentioned conditions, the maximum airport capacity is 27 operations in 20 minutes. In the case of GA algorithm, it delivers 16 operations within the 20 minutes, that is, 11 operations fewer than the MCMF. Obviously, if the situation were to be prolonged after the transient period, the maximum theoretical capacity under these conditions would be 40 operations per 20-minute time interval (120 operations per hour), with full usage of both runways.

The scattering in the use of resources is shown in the Fig. 15 for sink and source nodes (airport terminals and runways) and in Figs. 16 and 17 for the other

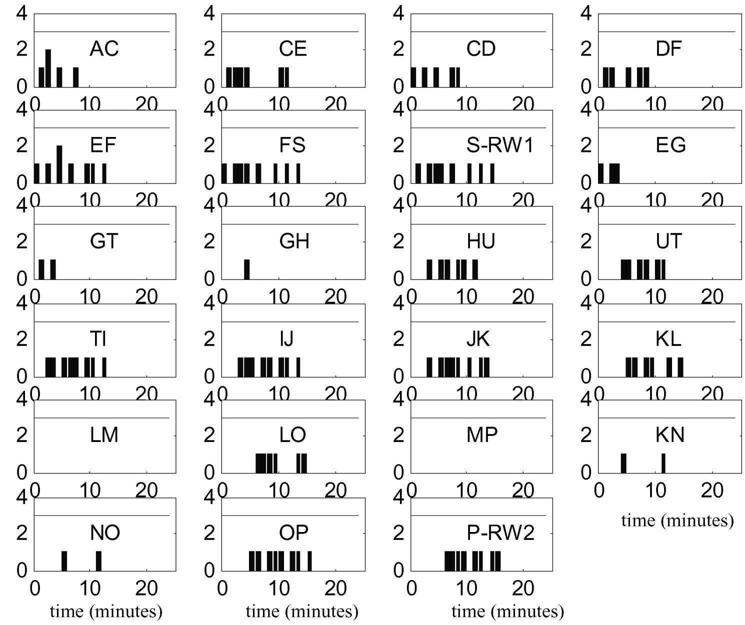


Figure 17. Flows with GA under saturation (detail of taxiway nodes).

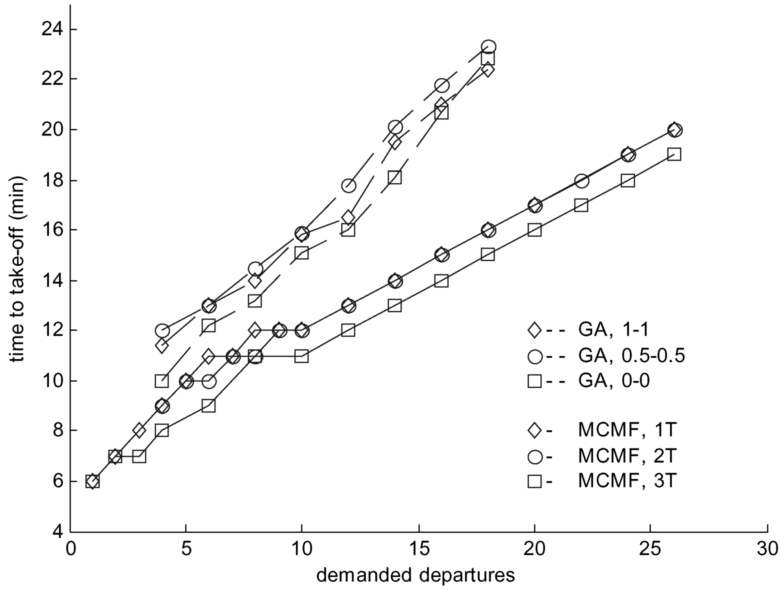


Figure 18. Time-to-take-off with MCMF and GA algorithms.

airport segments (inner nodes represent taxiway segments). As we can see, none of the taxiway arcs is filled to maximum capacity, since the limiting factor is runway capacity, as runways are only able to serve one operation per minute, while the capacity constraint on taxiways for moving operations is a lot less restrictive.

The flow-allocation algorithm was run in a number of different situations and the results with time-to-takeoff versus demanded departures are shown in Fig. 18. The number of demanded operations for departure was varied from 1 to the maximum number of operations that can takeoff within 20 minutes for each technique and configuration. In the case of the MCMF algorithm,

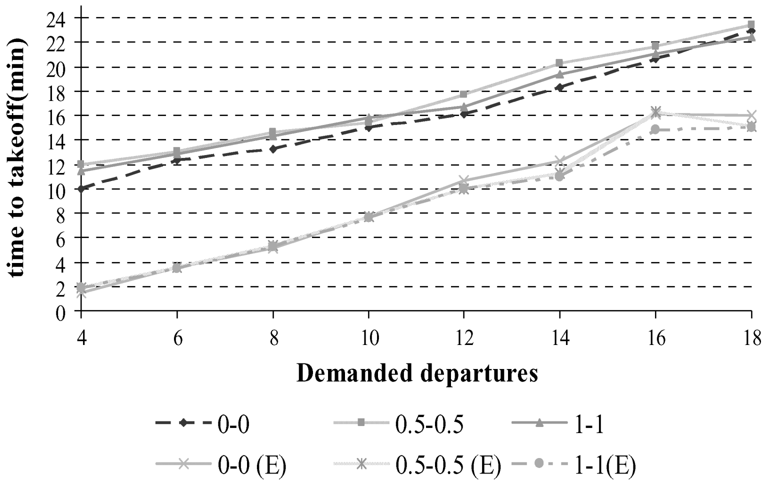


Figure 19. Comparison of training and takeoff times.

three different situations were considered: all operations coming from a single airport terminal, T1 and T2, and a situation in which demanded operations are interleaved according to the sequence T1-T2-T3. There are basically two regions in Fig. 18, corresponding to the usage of one or both runways. When there are few demanded operations, they are all sent to the closest runway RW1, with a slope of one minute per operation. As soon as total time is higher than needed to reach RW2, operations are sent to both runways, with a new slope of 0.5 minutes per operation. In the transition, the number of operations served depends on the specific time needed to access runways from terminals in each case. Minimum total time is achieved when all operations come from the closest terminal and two operations are lost in the additional time needed to reach runways in the cases of only operations from T2, T3. In the case of operations demanded from all terminals, the system decides the flow distribution with optimal usage of capacity.

Regarding the GA technique, we find that the time to takeoff grows linearly with the number of demanded departures (in the evaluated range). This behavior is a consequence of graph topography. As mentioned earlier, in the case of the GA technique, each operation may have a predetermined or unspecified destination runway. Three different lists of operations considering this possibility have been designed. The fraction of operation with the predefined destination runway has been set at 0, 0.5 and 1.0 for both runways. As expected, if all operations have a fixed destination runway, the total sum of delay increases. When the destination runway is not fixed, the algorithm searches for the best allocation to get the lowest average delay per operation.

With respect to the time consumed, the results presented in this section were achieved in runs on a Pentium IV 1.5 Ghz. The highest demand for computation was for the GA technique. We therefore assessed the training time required, illustrated in Fig. 19. Like the time-to-takeoff, the time to run the GA grows more or less linearly, with a steeper gradient than time-to-takeoff. When the training time cuts the time of takeoff, the projection over the demanded departures axis will provide the maximum number of takeoffs that the GA can schedule. As Fig. 9 shows, the values for this time are always lower than required for takeoff in the range of the assessed operations. This result is very important because it means that the GA can be used to output schedules in 30-minute time stages.

5.2. *Comparison with Manual Planning*

Finally, this section presents, for illustrative purposes, some performance figures for the planning techniques compared against a manual system with fixed routes. This reference procedure considers each terminal with a predetermined route to each runway (the runway with the shortest distance, generated by an off-line Dijkstra's algorithm run). If an operation is allocated to a fixed route, and this route is occupied at that time, the operation is delayed. Besides, the assumed operation mode classes landings as top-priority operations, and departures are assigned in the remaining space. Results were obtained by simulating sequences of 15 30-minute intervals (7.5 hours) with different loads for demanded operations and landings, simulated as uniform discrete events. The figures presented show the average time-to-takeoff versus the average load of demanded departure operations, taking the average demands for landing at the airport as a parameter.

The average demands for operations were:

- departures: 14, 27, 40, 63, 86, 120
- landings: 0, 10, 20, 30, 40, 60

The maximum capacity of both runways, with the simplified model taken, is $60 + 60 = 120$ operations per hour. Therefore, the value simulated last is a situation of congestion, which is worse when there are also landings using the runways. Under these conditions, operations cannot be served and are accumulated in the queue throughout the simulated time interval. The results are given in Fig. 20, which shows that maximum advantage is gained from dynamic and automatic planning when the situation is close to congestion. The definition of practical airport capacity [40] considers a maximum delay for the number of operations. Therefore, this procedure has an important advantage. The quantitative numbers are not completely representative of real capacity, since it is a simplified model under ideal conditions.

The right-hand side of Fig. 20 shows behavior when there are some segments in the airport that cannot be used (temporally restricted, special configurations, other operations in progress, etc.). As usual, it has been assumed that this information is known in advance by the planning function and it is input in the modeled capacity vectors. The probabilities of non-closed segments were set at 0.1, 0.2, 0.3, while the operation loads were:

- takeoffs: 14, 27, 40, 63, 86, 120
- landings: 0, 20

The same observations apply. The advantage of using information about available resources allows alternative routes to be computed, minimizing the impact of final delays compared with a rigid procedure.

6. Conclusion and Further Work

This paper applies two different approaches to show that the possibility of optimizing aircraft ground traffic at an airport is a useful support tool for assisting controllers. The results indicate promising performance, where the optimal capacity and flow distribution is calculated by means of a network flow algorithm and the departure schedule and aircraft routes have been computed by means of a genetic algorithm.

These techniques were integrated in a prototype decision support system, IPAGO, supporting the A-SMGCS concept. This integration improved the efficiency of the planning process, providing controllers with a means of automatically searching for adequate solutions, helping them to handle highly complex situations and assisting them with ground traffic management. Several planning techniques solve the problem of allocating taxiing routes to operations. Assuring the safety of operations while minimizing their average delay is, generally, the condition imposed on planning procedures, reflected as constraints taken into account in the search. The global system provides a standard representation of the planning problem, including both the routing and scheduling aspects, to be automatically processed by alternative techniques. This representation of solutions is useful for handling by other A-SMGCS functions, especially guidance, intended to help pilots to move on the surface following the routes allocated by controllers. Also, we mentioned interaction with the control function for checking that aircraft correctly follow routes.

Regarding the relation between the two specific techniques explored, the first strategy deals the planning problem as a network with timely constrained arcs to obtain the solutions with an optimal time-space distribution. It uses a simplified representation of the problem, where the time dimension has been discretized to produce vectors for decision variables and constraints, and specific elements have been proposed to include particular conditions of the airport ground planning problem in the modelled graph. The second strategy was the use of a genetic algorithm to directly search the combination of individual routes and schedules minimizing the time required to carry out all operations. A specific encoding has been designed to have a full and flexible representation of the problem.

Both techniques were compared only in simplified conditions. This was due to the fact that the flow distribution provided by the first technique provides

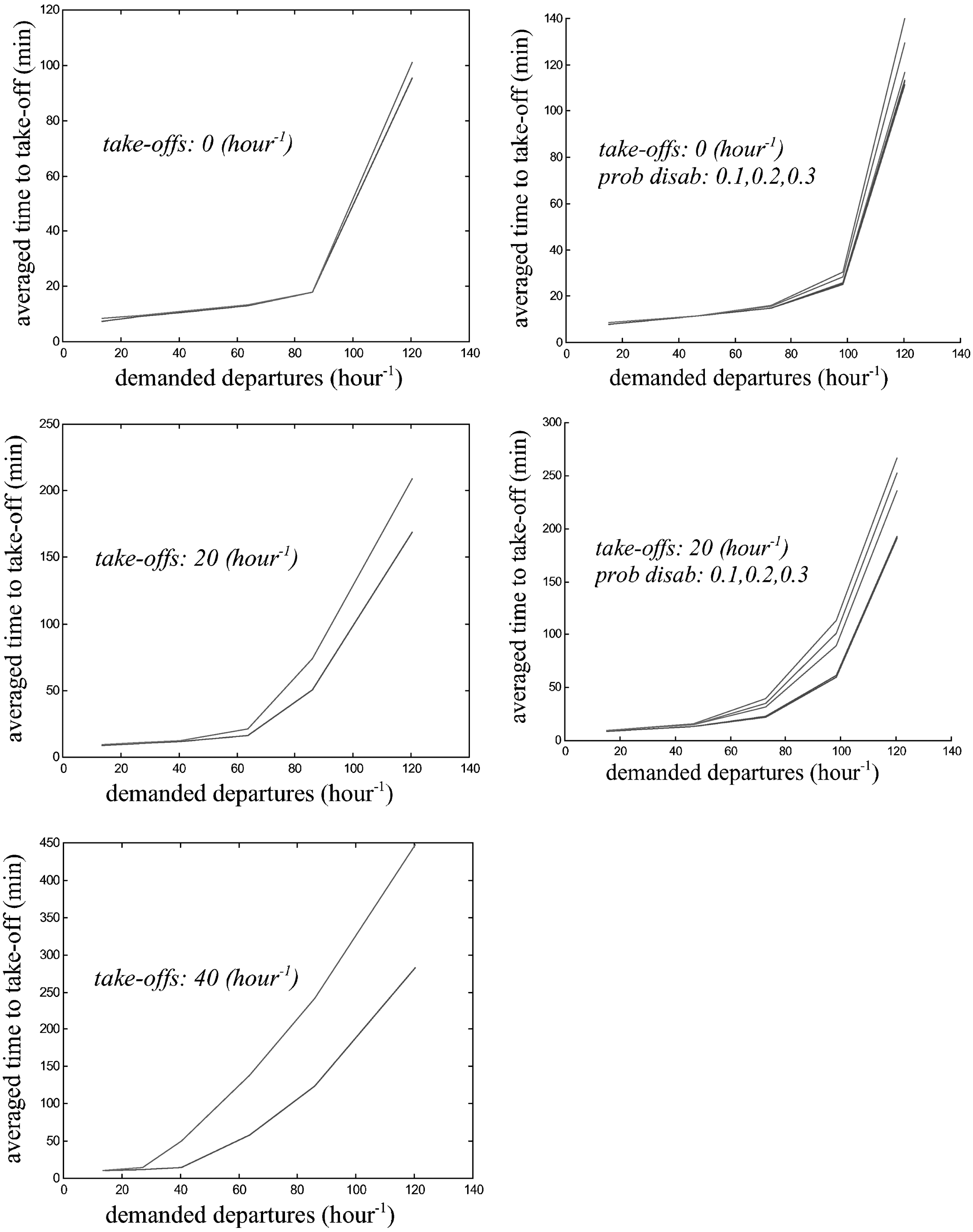


Figure 20. Average delays vs. demanded operations with automatic and manual planning.

indistinguishable flow units, losing the route plans and schedules for each individual demanded operation. In those cases where all demanded operations are equivalent and there are no individual constraints for operations, the solution directly extracted from the flow distribution is the optimum one and so has always superior performance. On the other hand, the GA technique is able to include specific considerations for individual operations such as assigned runway or separations depending on weight category, so it has capacity to deal with more realistic problems. The application of GA's requires from ad hoc adjustment, such as a careful design of fitness function or refinements in genetic operators (selection, combination, mutation). The ability to obtain effective solutions has been illustrated in the results, although the complexity of search space makes difficult the achievement of optimal solutions.

So, both techniques have a complementary relation, and it is open for future work the development of solutions by the hybridization of GA paradigm with the obtained flow distribution. Besides, the simplified model could be improved to incorporate more realistic procedures, such as acceleration and uncertainties on aircraft speeds, holding nodes, deviations from plans, etc., and the possibility of re-building plans considering the real trajectories observed and maneuvers performed. The genetic algorithm could be extended by incorporating new genetic operators and parameters into the fitness function but the problem representation as a transformed, flow-management technique-solvable problem is a more restricted approach. Although several transformations were proposed to represent the problem as a constrained-capacity-arcs directed graph, there are probably other simplifications (variations in speed, different constraints for individual operations) that cannot be removed without giving up the flow-distribution approach. In these conditions, flexible problem solvers with explicit representation, such as GA techniques, are the next step towards a full real-world representation, and here the use of an initial global solution, although over-simplified, may help to achieve an effective search.

References

1. ECAC and EC Workshop on A-SMGCS. Frankfurt, April 1994.
2. "European Manual on Advanced Surface Movement Guidance and Control Systems (A-SMGCS)". ICAO. Draft 0.1 04/11/1997.
3. Federal Aviation Administration. "The future airport SMGCS: A vision for transition into the 21st century," Nov. 1993.
4. J. García, J.A. Besada, and J.R. Casar, "Use of map information for tracking targets on airport surface," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 2. 2003.
5. J.A. Besada, J. Portillo, J. García, and J.M. Molina, "Image-based automatic surveillance for airport surface" FUSION 2001, Internacional, ISIF, Montreal, Canada, August 2001.
6. J. García, J.A. Besada, G. de Miguel, and J. Portillo, "Data processing techniques for conflict detection on airport surface," Fifth EUROCONTROL—FAA ATM R&D Seminar (ATM 2003). Budapest, Hungary, June 2003.
7. J. García, J.A. Besada, A. Berlanga, J.M. Molina, and J.R. Casar, "Planning techniques for airport ground operations," in *21st Digital Avionics Systems Conference 2002*, Irvine, October 2002.
8. G.F. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem-Solving*, 4th edition, Addison-Wesley Publishing, January 2002.
9. V. Tosic, O. Babic, M. Cangalovic, and D. Hohlacov, "A model to solve en-route air traffic flow management problem," Air Traffic Management 1997 Saclay, France, 2003.
10. A.R. Odoni, "The flow management problem in air traffic control," *Flow Control of Congested Network*, ASI series, vol. F38, pp. 269–288, 1987.
11. S. Luo and G. Yu, "On the airline schedule perturbation problem caused by the ground delay program," *Transportation Science*, vol 31, no. 4, pp. 298–311, 1997.
12. G.H. Tzeng, W.H. Huang, and D. Teodorovic, "A spatial and temporal bi-criteria parallel-savings-based heuristic algorithm for solving vehicle routing problems with time windows," *Transportation Planning and Technology*, vol. 20, 1997, pp. 163–181.
13. J. Milan, "Flow management problem in air traffic control: A model of assigning priorities for landings at a congested airport," *Transportation Planning and Technology*, vol. 20, no 2, pp. 131–162, 1997.
14. Daniel Delahaye, Jean-Marc Alliot, Marc Schoenauer, and Jean-Loup Farges. "Genetic algorithms for air traffic assignment," in *11th European Conference on Artificial Intelligence Applications (ECAI 94)*. Amsterdam, The Netherlands, Aug. 8–12, 1994.
15. S. Oussedik and D. Delahaye, "Reducing air traffic congestion by genetic algorithms," in *Proceedings of the Fifth International Conference on Parallel Problems Solving from Nature*, 1998.
16. David Gianazza and Jean-Marc Alliot, "Optimization of air traffic control sector configurations using tree search methods and genetic algorithms," in *21st Digital Avionics Systems Conference*, 2002.
17. Nicolas Durand, Jean-Marc Alliot, and Frederic Medioni, "Neural nets trained by genetic algorithms for collision avoidance," *Applied Artificial Intelligence*, vol. 13, no. 3, 2000.
18. F.R. Polak, "National Aerospace Laboratory NLR airport modelling: Capacity analysis of Schiphol Airport in 2015," Air Traffic Management 1997 Saclay, France, 2003.
19. Kurt Klein, "The DLR resarch programme on an integrated multisensor system for surface movement guidance and control," in *59th Sysmposium on Dual Usage in Military and Commercial Technology in Guidance and Control*, Guidance and Control Panel, 1994, pp. 9.1–9.8.
20. J.P.Gotteland, N. Durand, J.M. Alliot, and E.Page, "Aircraft Ground Traffic Optimization," in *Fourth International Air Traffic Management R&D Seminar ATM-2001*, Santa Fe, New-Mexico, USA, 2001
21. E. Gilbo, "Arrival/Departure capacity tradeoff optimization: A case study at the St. Louis Lambert International Airport (STL)," Air Traffic Management, Budapest, Hungary, June 2003.
22. M. Hansen, J. Tsao, and Sheng-Chen Alex Huang, "Empirical analysis of airport capacity enhancement impacts: A case study of DFW airport," University of California, Berkeley, July 1998, –NR-1998-015.
23. U.S. Dpt. of Transportation, Federal Aviation Administration, "Newark International Airport Capacity Enhancement Plan," May 2000. http://www.tc.faa.gov

24. J.E. Beasley, J. Sonander, and P. Havelock, “Scheduling aircraft landings at London Heathrow using a population heuristic,” *Journal of the Operational Research Society*, vol. 52, pp. 483–493, 2001.
25. H. Vandevenne and M.A. Lippert, “Evaluation of runway-assignment and aircraft-sequencing algorithms in terminal area automation,” *The Lincoln Laboratory Journal*. vol. 7, no. 2, pp. 215–238, 1994.
26. E.P. Gilbo, “Airport capacity: Representation, estimation, optimization,” *IEEE Transactions on Control Systems Technology*. vol. 1, no. 3, pp. 144–153, 1993.
27. E.P. Gilbo, “Optimizing airport capacity utilization in air traffic flow management subject to constraints at arrival and departure fixes,” *IEEE Transactions on Control Systems Technology*, vol. 5, no. 5, pp. 490–503, 1997.
28. E. Gilbo and K.W Howard, “Collaborative optimization of airport arrival and departure traffic flow management strategies for CDM,” Third EUROCONTROL—FAA ATM R&D Seminar (ATM 2000). Naples, Italy, June 2000.
29. G.D. Glockener, “Effects of air traffic congestion delays under several flow-management policies,” *Transportation Research Record*. N 1517, July 1996. pp. 29–36.
30. L. Navazio and G. Romanin, “Multiple connections multi-airport ground holding problem: Models and algorithms,” *Transportation Science*, vol. 32, no. 3 pp. 268–276, 1998.
31. G. Andreatta, A.R. Odoni, and O. Richetta, “Models for the Ground-Holding Problem,” in *Large-Scale Computation and Information Processing in Air Traffic Control*, edited by L. Bianco and A.R. Odoni, Springer-Verlag: Berlin, 1993, pp. 125–168.
32. E. Piazza, “A-SMGCS routing and guidance functions,” *IEEE Aerospace and Electronic Systems Magazine*, pp. 15–23, 2000.
33. B. Pesic, N. Durand, and J.M. Alliot, “Aircraft ground traffic optimisation using a Genetic Algorithm,” in *Proceedings of the Genetic and Evolutionary Computation Conference, Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Francisco (USA), July 2001.
34. J.B. Gotteland, N. Durand, and J.M. Alliot “Handling CFMU slots in busy airports,” Air Traffic Management, Budapest, Hungary, June 2003.
35. R.W. Hall, *Handbook of Transportation Science*, Kluwer Academic: Boston, 1999.
36. A. Dolan and J. Aldous, *Networks and Algorithms*, Jown Willey and Sons, 1993.
37. M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali, *Linear Programming and Network Flows*, John Willey and Sons, 1990.
38. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley: Reading MA, 1989.
39. J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press: Ann Arbor (2nd edition, 1992, MIT Press), 1975.
40. N. Ashford and P.H. Wright, *Airport Engineering*, 2nd edition, John Wiley & Sons, 1984.