Optimization of Airport Ground Operations Integrating Genetic and Dynamic Flow Management Algorithms

Jesús García\textsuperscript{a}, Antonio Berlanga\textsuperscript{a}, José M. Molina\textsuperscript{a} and José R. Casar \textsuperscript{b}

\textsuperscript{a} Dpto. Informatica
Univ. Carlos III de Madrid
e-mails: \{jgherrer@inf, aberlan@ia, molina@ia\}.uc3m.es

\textsuperscript{b} Dpto. Senyales, Sistemas y Radiocomunicaciones
Univ. Politécnica de Madrid
e-mail: jramon@grpss.ssr.upm.es

This work \textsuperscript{*} presents a new method to automatically search the best routes and schedules for airport ground operations, within a decision support system for tower controllers, a hard real-world application. It explores the potential advantages of hybridizing two complementary types of algorithmic approaches to find solutions with minimum delay: a genetic algorithm and a time-space dynamic flow management algorithm. An integration scheme to combine the strengths of each one and exploit their complementary nature has been analyzed. The proposed flow-management algorithm deterministically optimizes an over-simplified problem, while the genetic algorithm is able to search within a more realistic representation of the real problem, but success is not always guaranteed if search space grows. The performance of this combination is illustrated with simulated samples of a real-world scenario: ground operations in the Madrid-Barajas International Airport.

Keywords: Network Flow Management, Hybrid Genetic Algorithms, Scheduling, Routing, Airport Ground Operations

1. Introduction

The automatic planning function for airport ground operations will assist tower controllers to improve surface operations and safety in the process of traffic flow management. During peak periods of traffic flow or during capacity drops due to weather deterioration, demand temporarily exceeds the available operational capacity, and severe congestion appears resulting in expensive delays for users and airlines. Research on new surface procedures has been carried out to increase efficiency in the usage of current resources, in order to improve the current methods based on fixed routing schemes and pure mental processes \cite{2}. This set of new technologies and procedures supporting the future airport traffic management constitute the A-SMGCS (Advanced Surface Movement, Guidance and Control Systems, \cite{9},\cite{10},\cite{29}) concept, whose development will increase safety and efficiency in operations. Planning, the least mature of A-SMGCS functions, is conceived as a supporting tool to help in the selection of appropriate sequences and ground routes for demanded operations. The goal is the integration of the available information in A-SMGCS to automatically provide controllers with appropriate suggestions for complex situations.

A prototype for A-SMGCS system is currently being implemented at Barajas-Madrid international airport, the busiest airport in Spain \cite{5}\cite{14}. There, the current mode of operation is a segregated scheme, with one runway exclusively used for landings and the other for takeoffs, because of the simple advantage of direct management and fixed configurations. However, a mixed exploitation, with runways used both for landings and takeoffs may potentially increase available capacity, since the en-route separation between aircraft imposes lower utilization of runways when they are only devoted to landing. Besides, the flight-plans currently have pre-assigned gates and runways, with fixed routes from gates to runways depend-
ing on airport configuration, so that ground controller selects the starting time to meet the time slots delivered from European Central Flow Management Unit (CFMU). An automatic scheme dynamically selecting appropriate routes and schedules for demanded departures to obtain minimum ground delay is currently open to medium-term research, considering also the analysis of potential advantages with respect to current modes of operation. Some previous techniques have been studied in the authors’ research group, developed and integrated in a prototype system, IPAGO [13] (Intelligent Planning for Airport Ground Operations). A full description of planning problem details and solving strategies is presented in [15], together with the implications to the real problem of tower controllers handling dynamic situations. Several techniques were analyzed to improve the performance currently achieved with conventional procedures (a pure mental process performed by human controllers). The continuation developed in this current work explores the advantages of hybridizing complementary algorithmic approaches to find better solutions, combining their strengths and complementary nature. The information provided by a deterministic flow-management algorithm, referred to an over-simplified problem, is applied here to guide a genetic algorithm, potentially able to search within a more realistic representation of the real problem.

The planning problem of searching optimal time-space assignments has a NP-hard complexity [6], due to the complexity induced by dynamic traffic assignment when route and time of departures are simultaneously optimized, and usually have several optima. The search has a double nature of combinatorial space (the selection of the set of routes for all individual demands) and continuous adjustment (time intervals for operations kept in holding areas), with constraints to satisfy the safety requirements. Due to this complexity, the problem, represented with an appropriate formulation, is suitable to apply Artificial Intelligence techniques such as planning or stochastic optimization [31].

This particular problem of airport traffic management can be regarded as a special case of Traffic Flow Management, appearing both in communication and transport networks fields. In the case of airspace traffic, we can mention approaches based on temporal and spatial Operations Research techniques complemented with heuristics [41], [35], [32], [43], dynamic programming [33], and evolutionary algorithms for different levels of Air Traffic Control, such as traffic assignment [6], [36], design of airspace sectors [16] or en-route conflict resolution [8]. In the special case of traffic flow management at airports, there has been a strong interest to improve the use of available capacity. Simulation tools modeling airport operations, such as TAAM [39], SIMMOD (FAA) or TARMAC (DLR) [30] have been applied to analyze alternative configurations and bottlenecks in airports such as Schiphol [39], Orly, C. De Gaulle [24] or St. Louis [20]. Simulation has been complemented with data analysis to study the capacity enhancement derived from expansions or reconfigurations in airports such as DWF [27] or Newark [11].

With respect to specific techniques for planning airport operations, several problems with different levels of detail have been addressed. Most approaches are oriented to optimize the global use of runways or minimize congestion at destination airports. So, there are techniques aimed at computing appropriate landing sequences [4] and scheduling to assign multiple runways to landings (segregated mode) [44]. Integer programming has been also applied to on-line optimize the mixed assignment of takeoffs and landings to runways depending on demand [18], [19], and recently expanded to include collaborative decision-making paradigms [17], [20]. Other approaches decide the delays on ground to solve future problems at arrival on destination airports [21], [34], [1]. The major challenge is to address the details of planning ground operations, considering alternative surface routes for taxiing. Some suboptimal approaches search for solutions by considering individual operations one by one and the previously assigned traffic as constraints [38], while only a few works address the search of global solutions [38], [24], [25]. They are based on heuristics and genetic algorithms to explore appropriate decisions.

In this work, a new approach is applied to the airport-surface planning problem in the highest detail. Two types of previous approaches for solving time-space planning (presented in [15]) are now hybridized to improve efficiency in the search. A modified version of the Minimum-Cost Maximum-
Flow algorithm (MCMF [7]) is used to seed the initial population of a genetic algorithm (GA), and to penalize the fitness function. A direct strategy is the use of GAs to search routes and schedules minimizing the time required to carry out all operations. A specific encoding is needed to have a full and flexible representation of the problem and include specific considerations for individual operations such as assigned runway or weight category. In order to achieve a more effective search, GA is combined with a modified flow algorithm developed to obtain the optimal flow distribution over airport segments, starting from gates and ending in runways.

A modified algorithm derived from MCMF computes first 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, taking into account the dynamic conditions during the planning interval. To obtain this distribution, the planning problem is modeled with a network with timely constrained arcs, so flow distribution techniques are applied to dynamically obtain solutions with a deterministic optimal scheme. Besides, the time dimension has been discretized and only indistinguishable flow units are handled, losing the individual route plans for each individual demanded operation. Then, the GA exploits this previous information to obtain better solutions taking into account more details and constraints of the problem to derive the real solutions.

Genetic Algorithms have been extensively applied to discrete and continuous optimization problems in Operational Research fields. When complexity of search space increases, due to the combinatorial dependence on number of dimensions, a naive direct application will probably achieve poor performance. The application of GAs usually requires some ad hoc adjustments, such as a careful design of fitness function or refinements in genetic operators (selection, combination, mutation). Other alternative commonly used in previous works is the hybridization of GA paradigm with other specific optimization procedures, such as local greedy search, tabu search, or other domain-specific operators. For instance, we can refer to the combination of GA with the greedy heuristic of C4.5 algorithm for inducting decision trees [42], with local operators for bin-packing [40] quadratic assignment [12] or asymmetric traveller salesman [23] problems. All these proposals share that a local search technique is embedded into the genetic algorithm phases to improve existing solutions. The approach proposed in this work is different in the sense that the genetic algorithm is combined with a global algorithm providing an initial solution for a simplified problem. This solution defines the initial seed for searching better solutions.

The rest of paper is organized as follows. Section 2 presents a brief description of the problem to be solved by the airport planning function, indicating the specific features of the problem compared with other traffic flow management situations. In section 3 several possible solutions are described until arriving to the hybrid proposal. First, a formulation of a pure genetic algorithm (GA) is indicated, with the adaptations and enhancements to apply it to the problem described. Then, the mechanism for combining this GA with flow distribution is presented, indicating several alternatives. Finally, the algorithm coming from network flow techniques is summarized, indicating the proposed modifications to dynamically handle airport traffic units and provide simultaneously routing and scheduling. Section 4 presents some experimental results obtained for some representative scenarios in a simulation platform, and a discussion of results. Finally, summarizing conclusions and further work is presented in section 5.

2. The problem of planning ground operations

The planning function is intended to help for an efficient management of airport ground traffic, reducing the operations delays through a suitable assignment of resources. Airport resources can be represented as a set of space-time positions (4-D trajectories) so that the flow assignation problem is the design and assignment of these space-time trajectories to demanded traffic, with an optimization criterion and constraints to be satisfied by the solutions found.

The planning output should provide routes for each aircraft, considering the operations demanded both for landing and take-off. These operations share the airport resources: runways, taxiways and gates to aprons (airport areas containing parking positions). Therefore, the planning system must sequence and timely assign operations minimizing a global cost function, the sum of all taxiing and waiting times. Thus, the problem to solve presents basically two aspects:
– Find routes for operations with minimum delays (taxiing length).
– Find a sequence of operations and time schedule (assignment of time delays) to achieve optimal use of capacity.

The selected representation of airport resources (runways, taxiways and aprons), is a directed graph containing constrained capacity arcs, transit nodes and flow-source nodes. All the constraints to be considered for each interval planned should be collected and reflected in this graph. Constraints cover the runways state, considering time slots previously allocated for other operations, the current and predicted surface-traffic situation, safety alerts and modifications placed by controller. Source nodes in the graph represent the origin of demanded operations, basically airport passenger terminals for take-offs and landings fixes from close airspace. The rest of nodes in the graph are the reference places in the airport layout where an aircraft can be located, representing both waypoints in trajectory, generally junctions between runways and taxiways, or hold-on areas before accessing runways. An aircraft path, or route, is defined as a sequence of nodes, each one associated besides with an estimated time of arrival. Nodes in the graph are linked by means of arcs. An arc has three attributes: direction, cost, representing the time needed to cover it, and capacity. The available capacity of each arc represents the free space where flow units (operations) may be assigned along time for each planned interval, and so represents the real resources to be managed by the system. They have been represented as capacity vectors for each arc, which will reflect the difference between maximum capacity and the already planned operations. When a planned route occupies a certain segment, its capacity is decreased one unit for those intervals while the aircraft is supposed to be traversing that specific segment.

The maximum available capacity for each arc, defined as the number of operations that can enter by time unit, depends basically on the safe minimum longitudinal separation between operations and on the aircraft groundspeeds. Specific values for costs and capacities of the arcs in the graph representing Madrid-Barajas airport, depending on these characteristics, will be detailed in section 4.

The two basic requirements to the solutions searched, in order to be useful, are the generation from a global and dynamic point of view:
– Generation of global solutions imply the consideration at the same time of all operations to be served and the state of all resources, avoiding the generation of particular solutions useful only for individual interests. So, a global cost function such as the sum of all delays resulting from a certain solution must be evaluated to decide the most profitable actions.
– The scheme must integrate dynamically the information obtained about the current state of traffic, operations served, and other events such as indications from controllers or conflicts. So, it should be reactive to the evolution of global state and select the most adequate solution at each time. In the case that anomalous or hazardous situations are detected, or modifications are introduced by controllers, the flow management system should dynamically adapt and find the most adequate solution.

A simple and incremental illustration can be useful for explaining what means “global planning”, compared with other approaches oriented to one-to-one assignment, searching the shortest not-constrained path for each individual operation. In the case that a single departure operation in the queue has to be assigned, considering an empty airport, the system would obviously provide the shortest path to the closest runway, directly obtained with a shortest-path algorithm such as Dijkstra’s or A* algorithms [25]. However, if this shortest path currently is already occupied with other operation, considering a non-empty but pre-assigned airport situation, the system must decide now between two basic alternatives: delaying the starting time until the resources get free or selecting an alternative route to follow at the same time by both operations. Finally, situations will involve several competing operations simultaneously demanded for the same planning interval while at the same time resources will be shared by other operations in progress. The system will have to decide now their sequence, scheduled timetable, and routes assigned to each, in order to achieve the final goal of global minimum delay.

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 serve, being the control tower a centralized position. It must take into account constraints among operations, such as aircraft separation to guaran-
tee safety or minimum time intervals in the use of runways, and constraints on available resources, since they can be occupied by other pre-assigned operations. As an example of constraint, landings delivered by Air Traffic Control (ATC) centers at close airspace centers will have higher priority that departures authorized on surface. If all possible maneuvers for individual trajectories were considered, the decision variables for planning would be certainly complex. As a simplification, the system will decide only about routes and initial delays, supposing than each operation spends all necessary waiting time to meet its assigned slot stopped in the gate, situation preferred from the points of view of safety and manageability.

3. Solvers for Ground Planning

Once the airport-planning problem has been represented with a directed-graph format in the previous section, here we propose a hybrid approach, accordingly to the desired goals of minimizing times for demanded operations, satisfying the constraints.

Firstly, we detail a direct GA approach performing an explicit search using an Artificial Intelligence technique based on stochastic optimization. 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. The search is performed within the whole space of routes and time schedules.

In the second place, a hybrid strategy to integrate more information in the GA is presented, as a sequential application of two phases. First, a deterministic flow algorithm provides an initial flow distribution optimized for a simplified problem. Then, a refinement is left as a task to perform by means of GA with specific fitness function, taking the initial solution as starting point to search better solutions considering the individual constraints and detailed scheduling. The flow algorithm handles a simplified airport-planning problem represented as a flow-management one, in order to apply network flow algorithms extended to consider also time assignments. It represents the demanded operations as flow units, and determine the paths and time
intervals in the graph to achieve a maximum flow with minimum transit delays. The required minimum separations and assumed aircraft's groundspeeds have been translated to time-varying capacities of arcs.

Before explaining the algorithms details, it is important to state here that they will deal with a simplified model of airport conditions and aircraft motion on the ground. The main assumptions are (i) aircraft move on ground with uniform motion between the gates and runways; (ii) once the ground movement plan is delivered for an aircraft, there is no uncertainty about the trajectory it will follow; (iii) 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. Therefore, 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.

Under these conditions, the algorithms suggest the appropriate solutions, which can be helpful to the ground controllers to select the alternatives. To be a completely useful tool in operational conditions, some further steps should be addressed. The most important aspect would be a flexible interface to continuously re-define the problem, taking into account modifications made by human 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. 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 could be included in the models, such as holding positions at taxiing or before takeoff, variations in groundspeed, variable time separations in runways depending on weight categories, and uncertainty in maneuvers and speed throughout the planned time could be considered in the generated plans.

3.1. A Genetic Algorithm for Operation Planning

Network problems are one of the earliest applications of a kind of stochastic global optimization techniques labeled as Evolutionary Computation [6]. Genetic Algorithms search in the space of combinations of input parameters, providing fast and accurate solutions. In the field of transportation management, in the particular case of Air Traffic Management (ATM), the work developed by [6], [36], [16], [25], [4], [37] proposed GA solutions in order to improve some aspects of ATM.

We have developed a GA inspired in the previous mentioned works, incorporating ad hoc the mutation operator and fitness function to schedule the demanded operations. The algorithm described in [22], namely Canonical Genetic Algorithm (CGA), was applied in order to obtain the surface movements plan. We defined a plan as the departure schedule and the paths that a set of aircraft follows from gates to takeoff runways. The objective is to find the plan that reduces the average delay per operation, subject to the restriction of no conflict between operations.

J. Holland formally introduced the Genetic Algorithms (GA) [28] and, since then, their characteristics have made them widely applied to optimization problems, especially of combinatorial type. Their main characteristics are robustness and parallelism in the search, although optimality is not always guaranteed. The airport surface operation planning, formulated as a combinatorial problem, has a very large search space and an approximate solution could be satisfactory. Thus, the use of GA paradigm is justified since the trade-off between qualities of solutions and processing times is advantageous.

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

- Genetic representation. Each solution is coded as an instance of the vector with the decision variables. This codification is called the "genotype" of a solution.
- Genetic operators. The exploration and exploitation of the search space are performed applying three operators that produce new solutions from preexisting ones: selection, crossover (genes re-combination) and mutation.
Objective function. The criteria to measure "the goodness" of a solution are typically implemented by means of an evaluation function, generally referred to as "fitness function". Using the biological analogy, the genotype of a solution is expressed as the phenotype of one individual, and over this individual is applied the fitness function.

3.1.1. Genetic representation

For the case of air traffic ground plans encoding in a GA, they are codified with two sequences of numbers with length equals to the demanded departures, \(d\).

\[
\vec{r} = (r_1, r_2, ..., r_d) \quad \vec{t} = (t_1, t_2, ..., t_d)
\]

(1)

For each i-th operation, a plan assigns a route, \(r_i\), selected from a predefined set with all possible routes, and the time, \(t_i\), that the aircraft will delay its departure from the gate. This special codification allow us an easily implementation of the crossover and mutation operators, adapted to problem characteristic. Restriction of infeasible solutions, such as those which have operations assigned to the same route at the same time, cannot be taken into account in the codification so fitness function will penalize solutions that violate the restrictions. The codification only restricts \(r_i\) and \(t_i\) to be valid values; \(r_i\) must be within the range of possible operations from a set of fixed routes, and \(t_i\) must be an integer value between 0 and the maximum allowed delay.

3.1.2. Genetic operators

The main idea behind the GA performance is the "cumulative selection", although this term is not enough to provide the definite answer to the question of why it works. Cumulative selection is not a new concept at all, appears in stochastic optimization and other methods like descent gradient. The GA innovation is the incorporation of characteristics inheritance and variation trials in the search. These features, in a simplified form, resemble to the biological natural selection and are implemented through the genetic operators: selection, crossover and mutation.

Because the operator must be adapted to a particular problem, there are many genetic operators reported in the literature. In this work, the tournaments selection [22] was the selection scheme chosen for selecting the parent individuals in the population to generate following offspring. The crossover operator produces new solutions recombining the existing ones. In this work, a single-point crossover has been used. The crossover operator must be modified to consider the codification of plans as two sequences of numbers. The same crossover point is applied, to both part of two parent plans, in order to obtain the offspring.

Two mutation operators are used in this work. One is the traditional mutation described in CGA and the other one is a new operator included to speed up the appearance of small delay-time solutions. A random variation of the delay time, uniformly distributed in the range \([-8, 2]\), is applied with probability 5%.

Figure 2 describes the main steps of the algorithm in order to obtain a new population of solutions.

3.1.3. Objective function

Finally, the fitness function, which measures "the goodness" of solutions, is generically described here. The description is general to allow both a direct GA application and the proposed hybrid approach detailed in next section, where extra information obtained with a flow-assignment algorithm is included. The fitness value measures how a solution (a plan of operations) solves a problem, in this case represented as a series of demanded departures registered in a list of two-dimensional vectors, \((G, R)\), containing the departure gates and the takeoff runways. These specifications may be included for each individual op-
eration in the problem formulation, they may be omitted, or they might be the result of applying the flow-assignment algorithm.

In order to calculate the fitness values of a plan, $f_p$, a monitoring of the surface movements is performed with the assumptions enumerated before. The following quality measures are assessed, used as parameters of the following fitness function:

$$f_p = o \sum_i l_i + w \sum_i l_i + t_{plan} + 50c - 50k + r$$

The terms are defined as follows:

- When a plan contains an operation using a gate $G$ (source) or runway $R$ (sink) different to those specified, this plan is penalized with a value equal to the time that takes the wrong operation, $l_i$, multiplied by a weight. For the $i$-th operation, the time elapsed since the engine starts at the gate until the takeoff at runway, is represented as $l_i$. The weights, $o$ and $w$, are zero when the gate and the runway are right, respectively. And both weights have been fixed to 1 for the case of operations with errors. The reason to enable errors in the assignment of gates and takeoff runway is justified as an improvement of the search in amplitude. Is a consequence of that for this problem, the valid solutions are connected through solutions that violate some restriction, and the genetic search must be produced by means of little modifications over initially feasible solutions.

- Time to carry out the whole plan, $t_{plan}$. The simulation finishes when the last aircraft has taken off. This time, similar to the makespan of workflow problems, represents the total amount of time needed to carry out all required operations on surface, and it is so a global target to be reduced as much as possible.

- Number of conflicts, $c$. When two aircrafts violate the security distance, a conflict is reported. Obviously, a plan containing only a conflict is unacceptable; therefore, this parameter is strongly weighted. In the experiments, the best plan never has any conflict after the 20th generation.

- Number of take-offs, $k$. The objective is to obtain plans that process all demanded departures.

- $r$ is the sum of the operations delays, normalized by the number of operations that effectively finalize. The delay of an operation, $t_i$, explicitly codified for each operation as indicated in section 3.1, is measured as the time elapsed since the beginning of operating plan until the operation is authorized to depart from the gate. The maximum time of simulation has been fixed equal to 5000 seconds. This means that all operations should have finished within this time interval.

Regarding the coefficients in fitness function (50x factors for $c$ and $k$), they were adjusted to obtain appropriate tradeoffs among convergence time, quality and feasibility of solutions, after repeating some experiments with different seeds in the process. The design of the fitness function to achieve the best solutions followed a quite heuristic approach. Initial values were derived from preliminary runs and then they were experimentally tuned.

### 3.2. A Hybrid strategy to search for planning solutions

The previous general GA approach directly included the individual operations in the encoded problem, so the solution is referred to each operation: route assigned and time schedule. Our hybrid approach incorporates information provided by a dynamic flow management algorithm, Dynamic MCMF (D-MCMF). This algorithm was initially proposed by the authors in [15] as an extension of classical Minimum-Cost Maximum-Flow algorithm (MCMF) [7], and it will be briefly summarized later. In this algorithm, the operations to assign are first abstracted as (undistinguishable) flow units to compute the optimum flow distribution. It takes the number of operations from each terminal and selects how and when to deliver them to the available runways. In this approach the safety requirements on separation between operations are translated in the flow approach as capacity constraints. However, this is considered only as the maximum number of operations that can be sent along certain arcs to keep enough separations, but a fine adjustment of schedules is needed to satisfy the constraints. Besides, D-MCMF can only handle longitudinal separations for operations entering the same arcs, but no constraints among close
operations in different arcs, which must be explicitly checked. Other limitation of this flow approach is that some limitations refer to individual operations and should be explicitly considered. For instance, some operations can be constrained to depart only from a certain runway, so only routes ending in that runway from the departing terminal should be considered.

3.2.1. Integration of external flow distributions in the GA

The hybrid approach starts with the computation of flow distribution from the D-MCMF algorithm. This solution provides the flow units, \( x_l[k] \), for all arcs in the directed graph (\( l \in E \)) and all time slots considered (\( k=1,...,N \)). This information is taken by the GA to search the solutions in two aspects: initial population and fitness, modifying the pure approach where solutions in the initial population would be uniformly generated. The flow distribution will be used in the initialization population process of GA. The starting population of operations plans is filled with a random selection of operations matching with the distribution of flow units in the terminals and runways, with starting times randomized within the minute interval indicated in this distribution.

The information of the D-MCMF flow algorithm, summarized below, has been incorporated in the fitness function with four variations, to explore the different behaviors, all of them considering the first term in the expression above. The first one is labeled as "Pure-GA", applying the CGA explained in section 3.1, with a random initial population and the fitness function not including restrictions about flows calculated with the optimal-flow algorithm (parameters \( o, w \) depend only on external constraints for individual operations, if there exists any). The other three variant hybrid algorithms take the supplied flow distribution, all of them "injecting" the initial population as indicated above. The first one, labeled as "GA+Teminal-Runway Flow", incorporates in the fitness function the likelihood of terminal and runways distribution with the flows provided. So, the term \( o \), "number of incorrect origin gates", takes into account the departures from the terminal gates and the differences with the proposed flow distribution, and penalizes the difference. The same strategy is applied regarding destination runways, the term \( w \). The other two variants, labeled as "GA+Runway Flow", and "GA+Teminal Flow", remove constraints on one of both distributions, which is equivalent to setting constants \( o, w \), respectively to zero in fitness computation.

3.2.2. Dynamic flow management algorithms for optimal distribution

This algorithm handles a simplified representation of the airport planning problem, with a directed graph with constrained capacity arcs. This fact allows it compute in a deterministic way an optimum solution. D-MCMF extends the classical network algorithms to include dynamic variation in capacity along time, extensions in the number of sources and sinks and constraints in nodes. The problem addressed under this perspective is simplified and open to the further search to address all the relevant real-world constraints.

Classical algorithms for stationary conditions

Algorithms for flow management on networks come from Operational Research field [26,7], specifically from optimization techniques applied to integer-constrained linear programming. They are well-known methods to provide optimal routes and flow distributions in networks under stationary conditions (all flows are characterized with constant values or long-term statistics). A directed graph \( (V, E) \) is the data structure handled, being \( V \) a set of nodes and \( E \) a set of directed arcs or edges linking the nodes. The network is able to move some commodity along the arcs, being defined the flow as the quantity of commodity moved per time unit. The decision variables are positive-valued real variables \( x_l \), containing the flow distribution for all arcs in the network, \( l \in E \), according to the direction defined by each arc. Each node \( N \) in the graph is classified into one of three possible types, depending on the flow balance of arcs leaving the node and arcs arriving to it, \( b_N : \text{source}, \text{if } b_N > 0, \text{sink, when } b_N < 0 \text{ and transit, if } b_N = 0 \). 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, \( c_l \), defining the maximum flow that can be assigned (\( x_l \leq c_l \)).

Besides, when there are defined costs per flow unit for each arc, \( d_l \), we have a weighted basic net-
work. As it will be seen later, assumption of basic networks is not a severe restriction, since simple transformations can be applied to more generic networks and converts them into basic ones [26]. The three main results from network flow algorithms taken as starting point are briefly summarized next.

**Maximum Flow Algorithm**

This algorithm addresses the basic problem of searching the flow distribution in the network arcs, $x_i$, that maximizes the flow between S and T nodes, and besides satisfies all network constraints: the same flow leaving S arrives to T; in transit nodes the flow balance is zero; and the conditions $0 \leq x_i \leq c_i$ are attained by all arcs $i \in E$. A classic algorithm, due to Ford and Fulkeson [7], computes the maximum flow in the network by incrementally increasing the flow along augmenting paths while there is possibility to do that. It works with an extension of the edge set, defining for each occupied arc other in the opposite direction with as much capacity as the assigned flow in the direct direction. That allows a backtracking mechanism to find new flow-augmenting paths through re-assignment of previous routes and increasing the flow in the network.

**Minimum Cost Path**

This algorithms searches the route with minimum cost between S and T to send a flow unit: $\min \{ \sum_{l \in E} d_l x_l \}$, accomplishing the same network constraints as above. An optimal and efficient algorithm for this problem is the Dijkstra’s algorithm [7], which computes the solution in $O(m \log n)$ time, being n, m the number of nodes and arcs, respectively. Unfortunately, Dijkstra’s algorithm is only applicable when all costs $d_l$ are positive. A solution for the general case with positive and negative costs is the Bellman-Ford algorithm [3], finding the solution in $O(nm)$ time. As we will see next, although the original airport graph has positive costs (the time needed to traverse each edge), the transformation of Ford-Fulkerson’s method to discover flow-augmenting paths introduces arcs with negative costs, precluding the application of Dijkstra algorithm.

**Minimum-Cost Maximum-Flow (MCMF) Algorithm**

Finally, a combination of the algorithms solving the two problems mentioned above, maximum flow and minimum cost, allows to send a certain amount of flow, F, between source and sink nodes in a basic network, with the minimum cost. Besides, if the flow quantity F is increased until full saturation of the network appears, the problem addressed is then the delivery of maximum flow between source and sink with a minimum cost (Minimum-Cost Maximum Flow, MCMF [7], algorithm). The MCMF steps are the following:

- **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. Not saturated arcs have the original cost, saturated arcs have infinite cost, and for each arc with flow higher than zero, a fictitious arc in opposite direction and 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 assigned flow will be subtracted to the corresponding original arcs in the direct directions.
- **Step 3.** Repeat steps 1, 2 until there are no more unsaturated arcs available to find new paths.

3.2.3. Extensions of MCMF algorithm for dynamic management: D-MCMF

The classical network-flow algorithms described above have been extended and adapted to the airport problem representation: finding routes and schedules for the required operations (flow units) achieving an optimum usage of available capacity. The enhancements proposed to do that are two-fold. First, the introduction of time scheduling for operations (decision of time slots) in the search space variables of flow management algorithms, extending the dimensions of variables usually handled (basically assigned flows and capacities), to take into account the dynamic conditions during the planning interval. Secondly, the application of some basic transformations to the graph representing the problem in order to address important practical issues such as deciding the initial operation delays, assignation of multiple sources to multiple sinks and explicitly considering inter-
sections. The decided plans assigned to the demanded operations will dynamically depend on the airport conditions, such as available capabilities of runways and taxiways during planning period and other required operations for the same period.

So, the decision variables (flows for each arc) will consider the time dimension. To do that, both assigned flows and available capacities have been now represented with vectors, with as many components as time units considered for the planning interval. The MCMF algorithm has been reformulated with a representation of flows and capacities with N components, corresponding to N time intervals considered for planning: $x_l[k], c_l[k], k=1,...,N$, so the output depends on dynamic conditions (dynamic MCMF). This means only that the capacity available is not fixed, but varying, and divided in a finite number of time intervals (the unit is one minute). So, the flow units are assigned considering the available capacity depending on time. For instance, a taxiway is occupied when other operations are predicted to move there at a certain time, but then it is freed again and available for assignment.

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

This extension of MCMF algorithm, together with initial delays explicitly represented as additional nodes in the graph, will allow finding flow vectors to assign, which naturally include the sequence and schedule of operations. Other practical aspects covered before applying it to airport planning are the mentioned transformations performed on the graph representing the airport to address specific conditions of this problem: multiple sources and links, delay nodes and intersection nodes. They are simple transformations applied in the graph representing the airport [15], and the same strategy could be applied to extend the model further. For instance, we can identify some nodes in a path with special capacities, such as holding areas at the end of the taxiways, or areas in front of runway departing zones.

4. Experimental results

In this final section, the results obtained with the proposed planning methods applied to scenarios generated by simulation, are compared and analyzed. These scenarios were defined over the simplified representation of Madrid-Barajas airport with a directed graph (figure 1), composed by 24 nodes and 29 arcs. Simulated departure operations were generated to search for the best sequence, routes and schedules. For all the experiments, the pure GA used alone was considered as a benchmark to compare with the proposed hybrid combinations of GA and D-MCMF, considering the three variations described in section 3.2.2. Parameters of GA for all cases are summarized in table 1.

<table>
<thead>
<tr>
<th>GA parameters</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>200</td>
</tr>
<tr>
<td>Ending criteria</td>
<td>Generations = 200</td>
</tr>
<tr>
<td>Selection</td>
<td>Tournaments of size 4</td>
</tr>
<tr>
<td>Crossover Rate</td>
<td>100%</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>1%</td>
</tr>
<tr>
<td>Time Decrement Mutation Rate</td>
<td>5%</td>
</tr>
</tbody>
</table>

The capacities and costs of constrained arcs linking the nodes were adjusted assuming the following parameters for motion on ground and operational procedures: "Taxiing: average speed of 10 m/s and minimum longitudinal separation of 200 m. The junction areas of crossing taxiways were also characterized with a minimum spatial separation of 200 m between aircraft pairs. " Runways. The runways are the ending points of departure operations (flow sinks), and a requirement in their usage is a minimum time separation between consecutive take-offs of at least one minute.

With these parameters, the costs of all arcs located between gates and runways, assessing the time needed to cross them, were directly computed as their spatial length divided by the average speed on ground. Capacity of arcs depends on the type of movement area represented. For runways, due to the time constraint between consecutive take-offs, a single flow unit can go across a runway node per minute. Since a minute is the selected unit for time in the representation of 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 that can go across an arc per time unit depends on specification for longitudinal separation and speed. With an average speed of 10 m/s and of 200 m of minimum separation, maximum flow is 10/200 operations per second or, expressed in the selected time unit, 3 operations per minute.

Three scenarios have been defined to evaluate the planning algorithms, increasing the number of departure operations to assign: 6, 12 and saturation. In these experiments, the initial state of airport is taken as empty (all arcs with full capacity available) and operations are assigned considering a planning horizon of 20 minutes. In all scenarios, the “optimal” flow distribution computed with the D-MCMF algorithm, probably containing conflicts, is presented as reference. It is also the seed information considered by the hybrid algorithm. The solutions obtained by the four different alternatives described in section 3.2.2 are presented. All solutions derived satisfy the constraint of zero number of conflicts (separations below 200 m), due to the high weight in final fitness. They are compared in terms of the time needed to serve all operations. Besides, the space-time flow distribution of plans and schedules computed by different algorithms are depicted, so that they can be compared to ideal D-MCMF distribution and present the deviations. Since the evolutionary algorithms may present certain variations in the solutions achieved, 10 runs were performed for each algorithm. The flow distribution has been detailed for the two best runs, and the histograms of the quality achieved by each one of the ten solutions is also presented, assessed as the sum of departing times for each operation served. Finally, the convergence of population in the GA search is included too, presenting the evolution of fitness function and details of two components during optimization process: time to serve last operation, and number of violations in separation.

4.1. Planning 6 and 12 operations

The first experiment simulates a situation with six demanded departures, two from each airport terminal (T1, T2, T3), all of them assignable to any of the available runways, RW1, RW2. After executing ten runs of each algorithm, the two best solutions for each one are presented in figures 3-6. In all figures, are represented the distribution of operations in starting times at terminals and take-off at runways, indicating the specific placements and minutes used. The values corresponding to demanded operations, final solutions, appear as filled black bars, while the initial flow distribution computed by the D-MCMF algorithm is indicated with attached empty bars. Only the external nodes in the airport graph representing the sources (airport terminals) and sinks (runways) have been depicted, not including the rest of internal arcs in the graph. From the figures, we can see that D-MCMF algorithm solutions start two operations from the nearest terminal (T3) in the first minute, and one operation at the same time from T2. Within one and two minutes later, it starts an operation from terminal T1 and other from terminal T2 start, and finally, at least delayed two minutes, the second operation from terminal T1. Looking at the distribution in runways, five flow units are directed towards runway RW1 and one to RW2, achieving the objective of minimum total time needed to serve all demanded departures. First take-off at RW1 could start within fifth minute and first at RW2 within 10-th minute, which are the minimum time intervals needed to arrive at these nodes with the assumed ground speed.

Regarding the solutions searched by GA for individual operations (black bars), the best one is found when the GA algorithm includes the distribution of operations in the starting terminals in the fitness function (figure 6). In this solution, we can see that the take-off distribution is the same as the one computed by the D-MCMF algorithm. Since the solution cost is the sum of times needed to make all operations arrive to runways and depart, the best solution is the one achieving the lowest times for take-offs at runways. When any interval is left empty (holes in the runways occupations such as the second solution at the right-hand side of figure 6), the solution is not using all available resources. However, it cannot be generically said that solutions with empty slots are always sub-optimal, since the separation constraints to be taken into account make many configurations unfeasible. If we consider the sum of all departing times for served operations to summarize each solution, the distribution of frequencies in figure 7 (histogram of accumulated times) presents the distribution for all optimization runs of the different algorithms described. As we can see in this figure, the best solution was achieved with the GA
Fig. 3. Distribution of 6 operations with a pure GA

Fig. 4. Distribution of 6 operations with a hybrid GA using both terminal and runway flows

Fig. 5. Distribution of 6 operations with a hybrid GA using runway flows
using distribution at terminals, but it was found only once in ten executions. It is more significant that solutions using the distribution of operations in runways achieved in all the executions a solution better than any of those achieved by the pure Genetic Algorithm, without the flow information. It is noticeable too the phenomenon of achieving the same solution in all the executions for two versions: taking distribution on sinks (runways), and taking distribution both on sinks and sources (terminals). This contrasts with the scattered results obtained with the blind search of pure GA. Figure 8 presents the evolution in the optimization process in all versions, considering the best individual in the population for each generation, and averaged in the 10 runs. It is detailed the global fitness function and two representative indicators: time of last operation for each solution and number of separation failures (violations of rules). It can be seen that all solutions satisfy the restrictions (its value is quickly set to zero). Besides, a significant advantage appears in the hybrid algorithms using
the information with flow distribution for the simplified problem represented by the D-MCMF algorithm, achieving besides a faster convergence to stable solutions.

Next experiment is similar to the previous one, but now there were 12 operations to serve, 4 from each terminal. The D-MCMF suggests (empty bars in figures 9-12) an optimum flow distribution if system again serves first all the operations from terminal T3, closest to the runway RW1, and selectively delays the rest. This would allow a theoretical optimum runway occupation, with a compact sequence of take-offs in both runways. Searching solutions for real operations (filled bars), the best one is again found for the GA using terminal distribution in the fitness function, as indicated in the histograms of figure 13. From the figure, the solution was able to serve 11 operations in the optimum usage of runways, leaving only 1 operation delayed (hole).

Figure 14 presents again the evolution in the optimization process in all algorithms, considering the best individual in the population for each generation, and averaged in the 10 runs. A certain advantage was also obtained in this experiment with respect to the pure GA, as indicated in the distribution of final solutions depicted in 13.

4.2. Saturation situation

Finally, a saturation scenario was analyzed, simulating the situation where there are many operations from all terminals and the planner system decides which ones will be served within the horizon planning time, in this case of 20 minutes. The objective is again to minimize the time used to carry out conflict-free operations. However, since the maximum time is constrained to 20 minutes, the evaluated figure in this case is the number of operations that can be served. As we can see in figures 15-18 the D-MCMF algorithm found a distribution which starts 9 operations from terminal T3, 11 from T2 and 7 from T1, the terminal with higher delays. In this case, the idealistic distrib-
Fig. 9. Distribution of 12 operations with a pure GA

Fig. 10. Distribution of 12 operations with a hybrid GA using both terminal and runway flows

Fig. 11. Distribution of 12 operations with a hybrid GA using runway flows
ution of runway capacities usage allows 27 operations in 20 minutes. Obviously, it is a transient situation considering the initial times needed to move from terminals to runways. After the transient period, the maximum theoretical capacity in ideal conditions would be 40 operations per time interval of 20 minutes, or 120 operations per hour.

If we analyze the real solutions found, we can see that none achieves more than 18 operations served in the 20-th minute, achieving the separation constraints. Looking at the histograms of solutions in all cases (figure 19), the solutions obtained with the pure GA for this situation were not significantly improved with those using D-MCMF flow distribution. As we can see, solutions obtained by each version of hybrid algorithm were the same in the ten runs. The best solution, slightly improving some of the solutions found by GA, was achieved by version using both terminal and runway flow information in the fitness function. In figure 20 is also displayed the number of operations served within the 20 minutes interval by each algorithm.
Fig. 14. Fitness convergence averaged in ten runs for the best individual in population. 12 operations

Fig. 15. Distribution of operations under saturation with a pure GA.
Fig. 16. Distribution of operations under saturation with a hybrid GA using both terminal and runway flows

Fig. 17. Distribution of operations under saturation with a hybrid GA using runway flows

Fig. 18. Distribution of operations under saturation with a hybrid GA using terminal flows
Fig. 19. Histograms of accumulated times of operations under saturation for 10 runs

Fig. 20. Histograms of served operation within 20 minutes for 10 runs
Finally, the fitness evolution is displayed in figure 21. There is a bias between solution with pure GA and the solutions with hybridized algorithms due to the fact that the first one used 18 operations to assign (it needs a fix number of operations to be assigned), while the hybrid ones tried to deliver the 27 operations suggested by D-MCMF. Again, the number of conflicts is the first component which is minimized to obtain the solutions.

5. Conclusion and further work

This contribution addressed the open problem of optimizing the traffic on airport surface, dynamically considering the current state and constrained to the safety requirements. First, a standard representation of the planning problem, including both the routing and scheduling aspects, is formulated to be processed by the explored techniques. The problem was encoded into a Genetic Algorithm, hybridized with a proposed dynamic flow management algorithm. This flow algorithm simplifies the problem to search in a constrained graph to suggest optimum flow distributions, and requires additional adjustments to satisfy the constraints on separation. The flow distribution is used to start the search of operations assignments to routes and schedules, considering different versions to feedback the distribution suggested to the solutions in the GA population.

The results showed significant advantages of the proposal in moderate situations against a pure GA formulation. However, when the problem grows to the hardest conditions, maximum number of operations until saturation, all solutions found are similar and the distribution for the simplified flow problem hardly helps to improve the quality solutions. So the hybrid approach is useful while the difference between the simplified flow problem and the real achievable optimum assignment is moderate. Other advantages of using the hybrid strategy is that convergence in search is faster, which may reduce the computation time, and that solutions found with different runs have lower variance.
The proposed models could be extended in order to incorporate more realistic procedures, such as acceleration and uncertainties on speeds of aircraft, holding nodes or deviations from plans. The possibility to re-build plans, considering the real trajectories observed and maneuvers performed would be also an interesting issue. Further work will focus on these extensions to modify the algorithms and incorporate new capabilities, considering besides other possibilities to incorporate new genetic operators and parameters in the fitness function.

References


