Transportation Research Part C 80 (2017) 424–438

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Solving the gate assignment problem through the Fuzzy Bee Colony Optimization

Mauro Dell'Orco *, Mario Marinelli, Maria Giovanna Altieri

D.I.C.A.T.E.Ch. – Technical University of Bari, via Orabona 4, 70125 Bari, Italy

A R T I C L E   I N F O

Article history:
Received 13 January 2015
Received in revised form 5 May 2016
Accepted 27 March 2017
Available online 5 April 2017

Keywords:
Bee Colony Optimization
Metaheuristics
Multicriteria analysis
Air transport
Flight gate assignment problem

A B S T R A C T

In the field of Swarm Intelligence, the Bee Colony Optimization (BCO) has proven to be capable of solving high-level combinatorial problems, like the Flight-Gate Assignment Problem (FGAP), with fast convergence performances. However, given that the FGAP can be often affected by uncertainty or approximation in data, in this paper we develop a new metaheuristic algorithm, based on the Fuzzy Bee Colony Optimization (FBCO), which integrates the concepts of BCO with a Fuzzy Inference System. The proposed method assigns, through the multicriteria analysis, airport gates to scheduled flights based on both passengers' total walking distance and use of remote gates, to find an optimal flight-to-gate assignment for a given schedule. Comparison of the results with the schedules of real airports has allowed us to show the characteristics of the proposed concepts and, at the same time, it stressed the effectiveness of the proposed method.

1. Introduction

Several serious problems are linked to airport operations, including the use of runways capacity, flight delays, environmental pollution due to noise and gas emissions, and safety. Many pieces of research have thus dealt with managing congestion and allocating scattered capacity, as well as mitigating the external costs related to noise, emissions, and to safety/risk. Among these problems, the so-called Flight-Gate Assignment Problem (FGAP) studies the assignment of the subset of available terminal or ramp positions, called gates, to each flight (aircraft), minimizing passengers' inconveniences and/or maximizing the operational efficiency of the airport. The complexity of this task has continuously increased with the increase of civil air traffic so that in the FGAP modeling exact algorithms are rarely used while the use of metaheuristics has now become increasingly widespread.

In this paper, we consider a Flight Gate Assignment Problem that assigns airport gates to scheduled flights based on passengers' daily flow data. The objective of the problem is to minimize (i) the total walking distance that passengers walk to catch their connection, and (ii) the number of flights assigned to remote terminal gates. We formulate this problem as a bi-criteria optimization problem. Since the analytical solution of this problem is rather complex, we design a metaheuristic approach, based on the Bee Colony Optimization. A computational experiment has been conducted and the results are presented and analyzed.

The paper is organized as follows: in the next section, a synthetic survey of previous works presented in the literature is proposed. In Section 2, we describe the mathematical formulation of GAP and the relevant objective functions. In Section 3,

* Corresponding author.
E-mail address: mdellorco@gmail.com (M. Dell'Orco).

http://dx.doi.org/10.1016/j.trc.2017.03.019
0968-090X/

we explain the Bee Colony-based Optimization method (BCO), and verify its effectiveness in solving the FGAP. In Section 4, we consider two cases of Italian international airports as a test bed for the proposed method, reporting results of the sensitivity analysis. Finally, in Section 5 some concluding remarks are given.

2. A synthetic survey of models for the flight-gate assignment problem

Models for FGAP have been extensively studied mainly from two different points of view: the airport operators and the passengers' points of view. Following the first one, Hassounah and Steuart (1993) took into account stochastic flight delays to improve the performance of static gate assignment; Yan and Chang (1998) and Yan and Huo (2001) used a fixed buffer time between two contiguous flights assigned to the same gate, in order to absorb the stochastic flight delays. Gu and Chung (1999) designed a genetic algorithm, which generates effective solutions by calculating minimum extra delayed time schedules. Bolat (2000) proposed mathematical models and heuristic procedures to provide solutions with the minimum dispersion of idle periods for the FGAP. Yan et al. (2002) proposed a simulation framework, which can not only analyze the effects of stochastic flight delays on static gate assignments but also evaluate flexible buffer times and real-time gate assignment rules. Li and Xu (2012) maximized the utilization of the available gates and terminal through an immune genetic algorithm; Dorndorf et al. (2008, 2012) modeled the flight-gate assignment problem as a clique-partitioning problem. Şeker and Noyan (2012) considered the gate assignment problem under uncertainty in flight arrival and departure times and developed stochastic programming models. Yan and Tang (2007) minimized the flights delay through a heuristic approach. According to the passengers' point of view, Babic et al. (1984) minimized the walking distance of passengers through the branch and bound algorithm. Mangoubi and Mathaisel (1985) took into account transfer passengers, using the LP relaxation and greedy heuristics. Bihr (1990) proposed a simplified formulation of the gate assignment problem, minimizing the walking distance for fixed arrivals in a hub through a 0–1 integer program. Wirasinghe and Bandara (1990) integrate into terminal design process the cost of delays to minimize intra-terminal travel; Haghani and Chen (1998) tried to increase the customer satisfaction by minimizing the passenger walking distance between gates. Xu and Bailey (2001) used a Tabu Search algorithm for a single slot, minimizing the overall walking distances to get connecting flights. Ding et al. (2004) too used the Tabu Search to study the case in which the number of flights exceeds the number of gates. Instead, Lam et al. (2002) developed an intelligent agent to solve the FGAP; Drexl and Nikulin (2008) optimized their multicriteria objective using simulated annealing. Wei and Liu (2009) carried out a fuzzy model and an optimization algorithm; Prem Kumar and Bierlaire (2014) set up a mixed 0–1 integer program with a linear objective function and constraints.

As for the types of approach, the FGAP has been formulated through a wide variety of models: Bard et al. (2001) introduced an integral minimum cost network flow model, which reconstructs the operating programs of airlines in response to delays, transforming the routing problem in a time - based network, where the total time horizon is divided into distinct periods. Under some slight conditions, an optimum of the new model corresponds to the optimal solution of the original problem.

Another popular approach uses the rule-based expert systems: these systems assign flights to gates based on specific rules. For this kind of approach, it is necessary to identify all the rules, ordered by importance and listed appropriately. Initially, Baron (1969) combined optimization and rule-based approaches; then, Hamzwawi (1986) introduced a rule-based system for simulating the assignment of gates to flights, and evaluated the effects of particular rules on gate utilization. Srihari and Muthukrishnan (1991) used a similar approach for solving the FGAP, even describing how to apply the sensitivity analysis. Cheng (1997) integrated mathematical programming techniques into a knowledge-based gate assignment system to provide partial parallel assignments with multiple objectives.

A class of methods can be broadly categorized as "heuristic" algorithms since they do not yield an "exact" optimal solution. Hu and Di Paolo (2009) proposed an improved Genetic Algorithm applied to the FGAP, considering a multi-objective function; Cheng et al. (2012) carried out a comparison of performances of different metaheuristics (Genetic Algorithm, Tabu Search, Simulated Annealing) applied to the FGAP. The drawback of these metaheuristics is that they could generate infeasible solutions that a carefully designed fitness function should properly penalize. The proposed BCO algorithm overcomes this drawback since it builds the final solution always generating partial feasible solutions and improving them over iterations. Thus, in this way, the efficiency of the optimization procedure can be increased.

3. Mathematical formulation

In this section, we establish the mathematical model for the FGAP. First, let us list the input data:

N is the number of flights arrived at or departed from the airport during the planning day;
M is the number of gates available at the airport, including remote gates;
$f_{j,o}$ is the number of passengers transferring from flight j to the baggage claim area;
$f_{o,j}$ is the number of passengers transferring from check-in area to flight j;
$f_{j,r}$ is the number of passengers transferring from flight j to flight r;
$w_{i,o}$ is the walking distance between gate i and baggage claim area;
$w_{o,I}$ is the walking distance between check-in area and gate i;

$w_{i,k}$ is the walking distance between gate i and gate k;
$Y_{i,j}$ is a binary decision variable, equal to 1 if and only if flight j is assigned to gate i, 0 otherwise.

Then, the mathematical formulation is as follows:
Objective 1

$$\min\left(\sum_{i=1}^{M}\sum_{j=1}^{N}f_{o,j}\cdot w_{o,i}\cdot Y_{i,j}+\sum_{i=1}^{M}\sum_{j=1}^{N}f_{j,o}\cdot w_{i,o}\cdot Y_{i,j}+\sum_{i=1}^{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\sum_{r=1}^{N}f_{j,r}\cdot w_{i,k}\cdot Y_{i,j}\cdot Y_{k,r}\right) \tag{1}$$

in which the first term represents the total distance that the passengers walk to departure gates; the second term is the total distance that the passengers walk to baggage claim area, and the third term is the total distance that the passengers walk to reach connecting flights. This objective corresponds to the minimization the total walking distance.
Objective 2

$$\min\sum_{i\in RG}\sum_{j=1}^{N}Y_{i,j} \tag{2}$$

The meaning of this objective function needs some explanations: when a plane arrives at an airport, it can be assigned to a gate near the terminal, called fixed gate (FG), either to a remote gate (RG). Usually, passengers from nearby gates reach the terminal through movable connectors, called jet bridge, jetway or, more officially, passengers boarding bridge (PBB); passengers from remote gates are brought to the terminal by transfer buses or people movers. Such gates may increase connection time and thus are hardly eligible when the goal is minimizing connection time, as well as total passenger walking distance. Therefore, Eq. (2) aims to minimize the number of flights assigned to remote gates.

The optimization problem (1) and (2) is subject to the following constraints:

$$\sum_{i=1}^{M}Y_{i,j}=1,\quad 1\leqslant j\leqslant N \tag{3}$$

$$t_{i,j}^{dep}<t_{i,z}^{arr}\quad \textit{if}\ \ Y_{i,j}=1\ \textit{and}\ Y_{i,z}=1,\quad 1\leqslant i\leqslant M,\ 1\leqslant j,z\leqslant N \tag{4}$$

where $t_{i,j}^{arr}$, $t_{i,z}^{dep}$ are the arrival and departure time of flights j and z associated to gate i, respectively.

The constraint (3) expresses the necessity that every flight must be assigned to one and only one gate; the constraint (4) prevents the schedule overlapping, in case two flights are assigned to the same gate.

Compatibility between stands and planes is usually provided by airport regulations: a small aircraft can be assigned to a large stand, but the vice versa is impossible. A large stand can accommodate various sizes of aircraft while the flexibility of a small stand is more limited. However, to minimize the space requirement, airport terminals are usually built with stands of different sizes, so that very large aircraft have their operations restricted to a few stands and, therefore, to a few gates.

The program (1)–(4) is evidently a multi-objective optimization problem, whose techniques of solution can vary widely, passing from outranking methods (Roy, 1968) to interactive programming (Benayoun et al., 1971; Geoffrion et al., 1972) or, more recently, to the Best - Worst method (Rezaei, 2015). Since a deep study of the multicriteria analysis is out of the aim of this work, for the sake of simplicity we have used the most popular method for multiobjective optimization, the weighted sum method (Stadler, 1979). The method transforms multiple objectives into an aggregated objective function by multiplying each normalized objective function $f_i(x)$ by a weighting factor and summing up all weighted objective functions:

$$V(x)=\sum_{i=1}^{\gamma}\delta_i\cdot f_i(x) \tag{5}$$

where the coefficient δ_I is the weighting factor for the i-th objective function and γ is the number of functions to be aggregated. If $\sum_{i=1}^{\gamma}\delta_i=1$ and $0\leq\delta_i\leq 1$, the weighted sum is said to be a convex combination of objectives.

In our case $\gamma = 2$, therefore, $\delta_2 = 1 - \delta_1$ and the combination of Eqs. (1) and (2) gives:

$$\min\left(\delta_1\cdot\left(\sum_{i=1}^{M}\sum_{j=1}^{N}f_{o,j}\cdot w_{o,i}\cdot Y_{i,j}+\sum_{i=1}^{M}\sum_{j=1}^{N}f_{j,o}\cdot w_{i,o}\cdot Y_{i,j}+\sum_{i=1}^{M}\sum_{k=1}^{M}\sum_{j=1}^{N}\sum_{r=1}^{N}f_{j,r}\cdot w_{i,k}\cdot Y_{i,j}\cdot Y_{k,r}\right)+(1-\delta_1)\cdot\sum_{i\in RG}\sum_{j=1}^{N}Y_{i,j}\right) \tag{6}$$

where β is the normalizing coefficient. The program (6) with the constraints (3) and (4) is hardly, if not at all, manageable by ordinary analytical tools; therefore, in this paper, we have proposed a metaheuristic, based on the collective intelligence concepts, called Bee Colony Optimization (BCO). Due to a possible uncertainty inherent in the input data, we have implemented a fuzzy version of the BCO.

4. The computational paradigm of the Bee Colony Optimization

The observation of various natural systems, like social insect colonies, shows that dynamic interaction among very simple individual organisms often create systems able to perform highly complex tasks, which original individuals cannot perform by themselves. These systems are usually characterized by autonomy, distributed functioning and self-organizing; their dynamics is a result of different actions and interactions of individuals with each other, as well as with their environment. Chemical and/or physical signals perform the interactions, which finally produce the behavior of the whole system. The communication systems contribute to the formation of the "collective intelligence", commonly known as "Swarm Intelligence".

In the last years, several attempts have been made to use the concepts of the Swarm Intelligence in the development of various Artificial Systems, able to solve complex combinatorial problems. The scientific community began to deal with complex problems through natural metaphors, like the behavior of bird flocks, fish schools, swarms of social insects as ants, termites, bees, etc.

In this paper, we explore a new direction in the field of Swarm Intelligence: the enhancement of the Bee Colony Optimization in solving complex problems characterized by uncertainty through the concepts of the Fuzzy Logic.

The behavior of honeybees in nature, when searching for the best site for nectar source, presents strong analogies with the search of optimal solutions in an optimization process: some "scouts" bees explore the area around the hive and, when they find a nectar source, return to the hive. After dropping the nectar, the scout bee can abandon the food source and become again a follower, or continue foraging at the food source, or dance, "advertising" different food areas, and thus recruiting the nest mates. The exchange information about both the location and the quality of the nectar sources takes place through the so-called "waggle dance", performed on the hive surface. The bees in the hive can either follow a scout bee or start a new search, becoming scouts. They can even remain in the hive, skipping the foraging at that moment. Camazine and Sneyd (1991) considered that "the recruitment among bees is always a function of the quality of the food source".

In this metaphor, foraging represents the solution generation phase, while the waggle dance is the information exchange phase, in which the quality of existing solutions is examined and the generation of the new ones is directed.

4.1. Brief literature overview

In last twenty years, a large quantity of papers proposing algorithms based on the behavior of honeybees swarms to solve combinatorial optimization problems has been published. In the following, we cite some of them, referring to the work of Davidović et al. (2014) for those who want a more in-depth literature examination. First, Sato and Hagiwara (1997) proposed the Bee System as an improvement of the genetic algorithm, employing some new operations such as the Pseudo-Simplex Method. Abbass (2001) presented a model that simulates the evolution of honey-bees starting with a solitary colony (single queen without a family) to the emergence of a eusocial colony. From the optimization point of view, the model evolves solutions using a committee of heuristics. Lučić and Teodorović (2001, 2003) introduced the Bee System (BS) as a system that uses the principles of collective bee intelligence when dealing with complex combinatorial optimization problems. The BS was tested in the case of Traveling Salesman Problem. Nakrani and Tovey (2004) proposed a new decentralized honeybee algorithm, which dynamically allocates servers among clients to satisfy request loads and to maximize the revenue. Wedde et al. (2004) presented BeeHive, a routing algorithm in which the bee agents exchange information on the network state for updating the local routing tables. Afterward, Teodorović and Dell'Orco (2005) evolved the BS in a metaheuristic, called Bee Colony Optimization (BCO), which they used to solve the Ride-matching problem (Teodorović et al., 2006). Drias et al. (2005) introduced a meta-heuristic named "Bees Swarm Optimization" (BSO), providing an overview of the results of empirical tests performed on the hard Johnson benchmark. At the same time, Yang (2005) developed a virtual bee algorithm (VBA) to solve the function optimizations. The solution for the optimization problem can be obtained from the intensity of interactions among virtual bees. The VBA was tested through the De Jong's function and Keane's multi-peaked bumpy function. Karaboğa (2005) presented the Artificial Bee Colony (ABC) algorithm, in which self-organization and division of labor are necessary and sufficient properties to obtain swarm intelligent behavior, such as distributed problem-solving systems. Pham et al. (2006) have presented another algorithm, called Bees Algorithm (BA), mimicking the food foraging behavior of honeybee swarms. The algorithm performs a kind of neighborhood search combined with random search and has been demonstrated to be efficient and robust fora number of benchmark problems. Afshar et al. (2007) used the honeybee mating optimization (HBMO) algorithm to handle the problems related to the optimization of single reservoir operation. They showed that the performance of the model is quite comparable with the results of the traditional linear programming solvers. Yang et al. (2007) proposed the Faster Marriage in Honey Bees Optimization (FMBO) algorithm, whose computation process becomes easier and faster through a random initialization of drones and a limitation of the iteration conditions. Narasimhan (2009) presented a version of the ABC algorithm for shared memory architectures: at the end of each cycle, the solutions obtained by single sets of bees are copied into the shared memory and made available for comparison to all other bees. The procedure achieves a substantial speedup. Fonseca et al. (2010) carried out an application to molecular biology of the Bee Colony Optimization, with the hill-climbing method as local search, to determine the three-dimensional structure from the one-dimensional amino acid sequence. Dimitrijević et al. (2011) applied the BCO to the anticovering location problem (ACLP), one of the fundamental problems in the area of discrete location, and found that the proposed algorithm can generate high-quality solutions in reasonable CPU times. Finally, Davidović et al. (2012)

studied the static scheduling of independent tasks on homogeneous multiprocessor systems by the Bee Colony Optimization, obtaining that the BCO is competitive with state-of-the-art methods for similar problems, with respect to both solution quality and running time.

4.2. Description of the metaheuristic

The basic idea is that a colony of artificial bees acts as a multi-agent system searching for good solutions of various combinatorial optimization problems. The agents explore the space of solutions according to the principles used by honeybees during nectar collection process, looking for feasible solutions. The artificial bees collaborate and exchange information, so that the collective knowledge thereby created allows concentrating the search on the more promising areas, abandoning the less promising ones.

The BCO search is running in iterations: when bees create a feasible solution, an iteration ends and the best-discovered solution is saved. Each single step of the algorithm is made of two alternating phases: the forward pass and the backward pass. In the forward pass, artificial bees create various partial solutions, as depicted in Fig. 1.

After that, they perform the backward pass, ideally returning to the hive, where they participate in a decision-making process. In this process, artificial bees exchange information about the quality of the partial solutions created; in practice, all generated partial solutions are compared, and each bee decides whether to remain loyal to its solution or not. The decision is made based on the goodness of the solution, with a certain probability: better solutions have a greater probability to be chosen in the further prosecution, and some worse solutions can be discarded (Fig. 2). In the next step, the bees perform again a forward pass; this time, according to decisions made in the previous backward pass, all artificial bees explore the search space. They apply a predefined number of moves, which construct the partial solutions, yielding the new partial solutions as an increase the partial solution with new components, creating new partial solutions (Fig. 3). Then, a new backward pass is performed and the cycle begins again.

In this way, the artificial bees incrementally construct a solution of the problem.

Now, let:

$ST = \{st_1, st_2, \ldots, st_m\}$ be a finite set of pre-selected stages, where m is the number of stages;

B be the number of bees participating in the search process;

I be the total number of iterations;

S_j ($j = 1, 2, \ldots, m$) be the set of partial solutions at stage st_j.

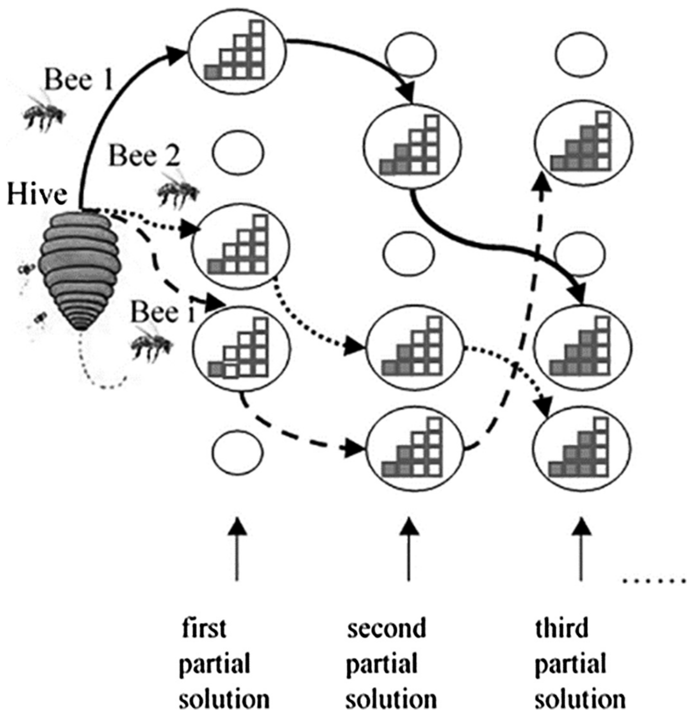

Fig. 1. Forward pass.

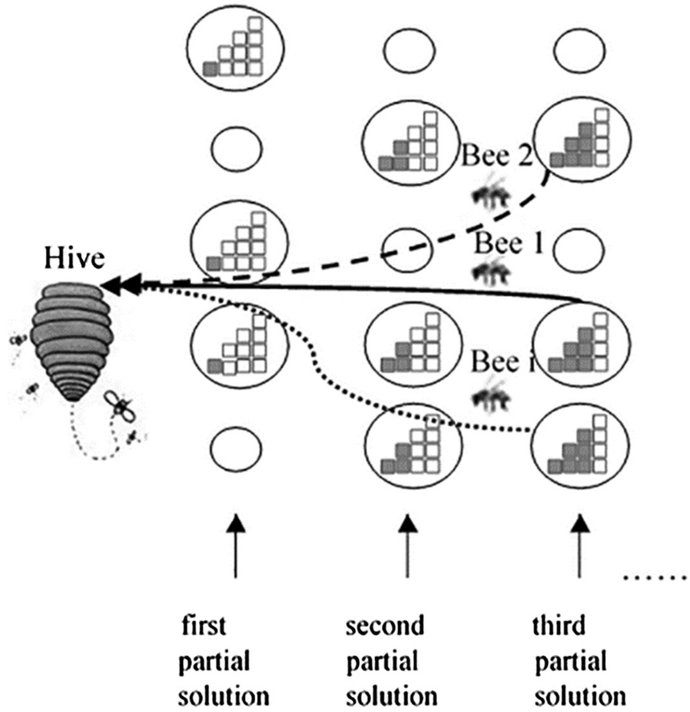

Fig. 2. Backward pass.

The following pseudo-code shows the Bee Colony Optimization algorithm.

1. *Initialization*. Set the number of bees B, and the number of iterations I.Select the set of stages ST = $\{st_1, st_2, \ldots, st_m\}$. Find a feasible solution x of the problem. This solution is the initial best solution.
2. Set i := 1. Until i = I, repeat the following pass:
3. Set j = 1. Until j = m, repeat the following pass:*Forward pass*: Allow bees flying from the hive and choosing B partial solutions from the set of partial solutions S_j stage st_j.*Backward pass*: Send all bees back to the hive. Compare the partial solutions created and decide, for each bee, whether
   (i) to abandon the created partial solution and become again uncommitted follower;
   (ii) to continue to expand the same partial solution without recruiting the nest mates;
   (iii) recruit the nestmates before returning to the created partial solution.Set j := j + 1.
4. If the best solution x_i obtained during the i-th iteration is better than the best-known solution, update the best-known solution ($x := x_i$).
5. Set i := i + 1.

For each single problem, we can set up different stopping criteria: maximum total number of forward/backward pass, maximum total number of the forward/backward pass between two value improvements of the objective function, etc. Additionally, various sub-models can be developed, describing how bees decide to abandon the created partial solution or to continue to expand the same partial solution without recruiting the nest mates, or to dance and recruit the nestmates before returning to the created partial solutions.

Although the BCO algorithm has proven to be effective in solving deterministic combinatorial problems, such problems are often affected by uncertain or approximation in the data. Therefore, in this paper we propose the Fuzzy Bee Colony Optimization (FBCO): in this method, the artificial bees use the rules of fuzzy logic to exchange information or make decisions. We have developed this heuristic algorithm for the Gate Assignment Problem as an illustrative example useful to show the characteristics of the proposed concepts.

4.2.1. The Fuzzy Bee Colony optimization

Choice/decision processes were usually modeled based on random utility modeling concepts. These models make the assumptions that decision-makers know perfectly all alternatives and behave always in a rational way, trying to maximize their utilities. However, frequently decision-makers have only approximate or imprecise information about the system,

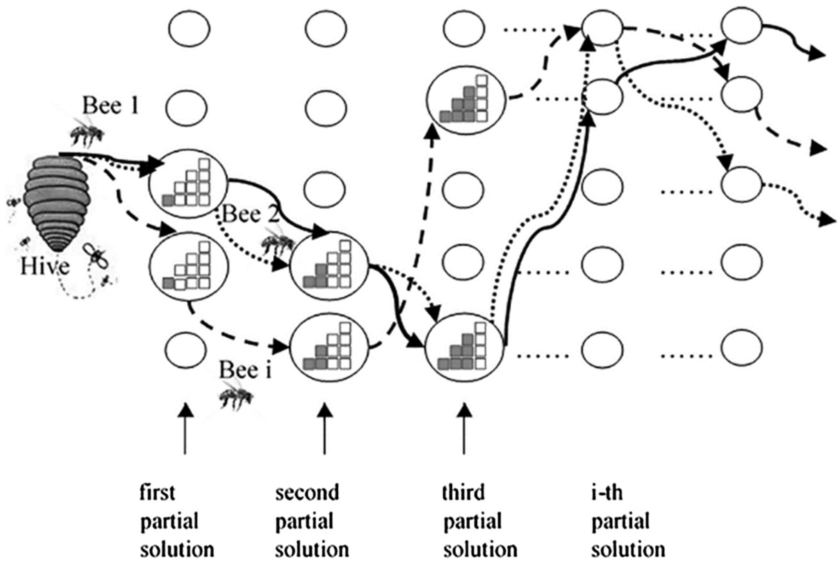

Fig. 3. Second forward pass.

and/or their behavior could be not perfectly rational (see, for example, Gigerenzer and Selten, 2002). Therefore, researchers felt the need of different modeling approaches, capable of more understanding for uncertainty, imprecision and verbal data, such as the Fuzzy Sets Theory (Zadeh, 1965, 1999). Fuzzy Sets are numerical sets capable of representing vague or verbal notions, like "close", "far", "slow" or "fast". A Fuzzy Set is defined through the membership function $\mu(x)$ having the form:

$$\mu(x) : X \rightarrow [0, 1], \quad x \in X \tag{7}$$

where X is the universe of the discourse. To each numerical value in the domain X, a specific "grade of membership" is assigned; 0 represents the smallest possible grade, and 1 is the largest possible grade. The most popular representation of fuzzy numbers is a triangle, in which three points, the lower boundary v_l, the center value v_c, which corresponds to the highest grade of the membership, and the upper boundary v_r represent a fuzzy number N:

$$N = (v_l, v_c, v_r) \tag{8}$$

For example, in the case of the variable "distance", possible attributes could be "short", "medium" and "long", represented as follows: short = $(0, 0, 1000)$; medium = $(0, 500, 1000)$; long = $(0, 1000, \infty)$.

A Fuzzy Inference System (FIS) is a system that uses the Fuzzy Set Theory to map inputs to outputs. The mapping provides, through some '*if...then*' fuzzy rules, a basis from which decisions can be made, or patterns discerned (Zimmermann, 1996). Usually, an FIS starts with a number, for example, the evaluation of a partial solution, as input and provides another number, like the attractiveness of the partial solution, as output. An interface called Fuzzy Inference Engine fuzzifies crisp inputs, and defuzzifies fuzzy outputs, as reported in Fig. 4, where a scheme of a generic Fuzzy Inference System is depicted.

In our model, we made the assumptions that artificial bees use verbal values in their communication and make their choices according to the rules of an FIS.

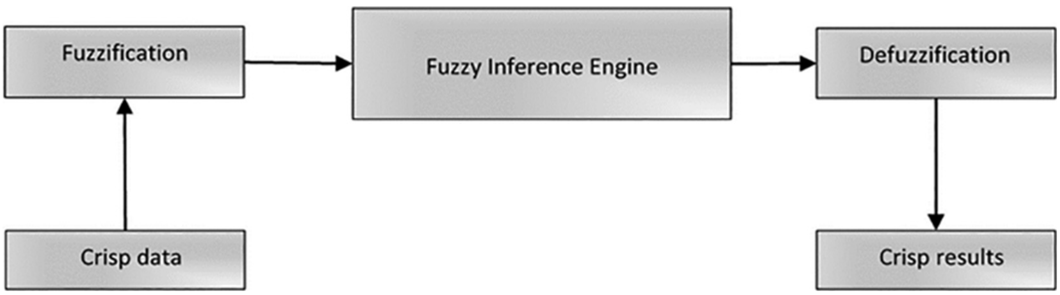

Fig. 4. The Fuzzy Inference System.

Table 1
Rules of the 'node-attractiveness' Fuzzy Inference System.

Value of objective function	Component attractiveness
Low	*High*
Medium	*Medium*
High	*Low*

Let us call '*node-attractiveness*' the fuzzy attribute of a specific partial solution component, calculated through a very simple FIS, represented in Table 1.

The table is read as for example:

If the normalized value of the objective function of the solution component is *Low*
Then the attractiveness of the considered solution component is *High*

If we denote by f_i the attractiveness calculated for the solution component i, the probability that the partial solution would be chosen by an uncommitted artificial bee is:

$$p_i = \frac{f_i}{\sum_{j=1}^{S} f_j} \tag{9}$$

where S is the total number of solution components added so far. The choice of the next solution component to add to the partial solution can be carried out by several different methods, like the "stochastic universal sampling" introduced by Baker (1987), or the "tournament selection" (Miller and Goldberg, 1995). In this work, we have used the well-known "roulette wheel" selection, inspired by a game-gambling roulette: the selections of a roulette wheel are proportional to probabilities p_i.

To give significant influence on the future search directions to "historical facts" discovered by *all members* of the colony, bees compare their discovered partial solutions with the best and the worst discovered partial solution from the *beginning* of the search process.

Every partial solution (partial path) that is being advertised in the dance area has two main attributes: (a) the objective function value, and (b) the number of bees B that are advertising the partial solution. The number of bees that are advertising the partial solution is a good indicator of a bees' collective knowledge. It shows how bee colony perceives specific partial solutions.

The approximate reasoning algorithm to determine the attractiveness a_i of the advertised partial solution i consists of the rules represented in Table 2.

As usual, the table is read, for example:

If the length of the advertised path is **Short** and the number of bees, advertising the path, is **Small**
Then the attractiveness of the advertised partial solution is *Medium*

Path attractiveness calculated in this way can take values from the interval [0,1]. The higher the calculated value, the more attractive is advertised path.

The comparison of partial solutions is carried out based on the concept of partial solution badness. Assuming that we are dealing with a minimization problem, we define the badness of a partial solution L_k in the following way:

$$L_k = \frac{L^{(k)} - L_{\min}}{L_{\max} - L_{\min}} \tag{10}$$

where

$L^{(k)}$ is the objective function value of the partial solution discovered by the k-th bee
$L_{\min}$ is the objective function value of the best-discovered partial solution from the beginning of the search process
$L_{\max}$ is the objective function value of the worst discovered partial solution from the beginning of the search process

Table 2
Rules of the 'path-attractiveness' Fuzzy Inference System.

Path length	N. of advertising bees		
	Small	**Medium**	**Big**
Short	*Medium*	*High*	*High*
Average	*Medium*	*Medium*	*Medium*
Long	*Low*	*Low*	*Medium*

Table 3
Rules of the 'loyalty' Fuzzy Inference System.

Path attractiveness	Badness		
	Low	**Medium**	**High**
Low	*Medium*	*Medium*	*Low*
Medium	*High*	*Medium*	*Low*
High	*High*	*Medium*	*Medium*

At this point, each bee artificial decides whether to stay loyal to the previously discovered solution or to abandon it. The approximate reasoning algorithm to determine the degree of loyalty consists of the rules represented in Table 3.

Analogously to the previous table, this table is read as:

If the attractiveness of the discovered partial solution is **Low** and the badness of the same partial solution is **High** **Then** loyalty is *Low*

Before the new forward pass, we change the number of bees n_i joining the path i through the following equation:

$$n_i = \frac{a_i}{\sum_h a_h} \cdot B \tag{11}$$

Recall that B is the number of advertising bees, while a_i is the attractiveness of the advertised partial solution i. Using collective knowledge and sharing information among themselves, bees concentrate on more promising search paths and slowly abandon less promising paths.

4.3. Application of the method to the gate assignment problem

When applying BCO to the FGAP, a bee represents a solution of the problem, which is a feasible assignment of scheduled flights to available gates. Consequently, the decision space is represented as an artificial network (Fig. 5), where the bees fly to find the optimal path. The network is composed of layers or stages; each layer represents a single flight F_i. Towards these layers, there is the set of flights, ordered according to a given schedule. Each node (solution component) represents a possible assignment of a flight F_i to an available gate; therefore, it refers to the variable $Y_{i,j}$ in the problem formulation (Eqs. (1) and (2)).

During a single iteration, in the forward step each bee flies through layers (stages of the BCO), building partial solutions as partial feasible assignments of flights to gates. At each stage, the attractiveness of available gates is evaluated through the FIS 'node-attractiveness'. Then, a bee chooses the next flight-to-gate assignment through a roulette wheel selection according to Eq. (9). In the backward step, the adequacy of all partial solutions in respect of the constraints of the optimization problem is verified, and the value of their associated fitness is calculated by the objective function (6). For each partial solution, the

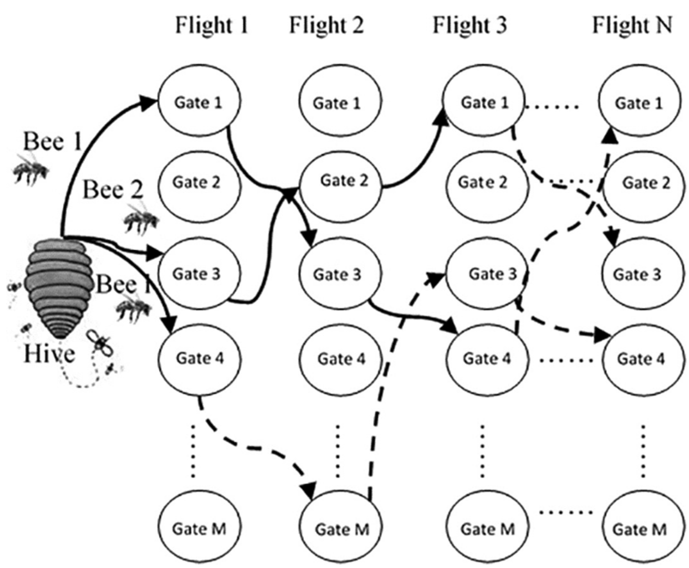

Fig. 5. The artificial network of the flight-to-gate assignment space.

badness is calculated through Eq. (10), while the associated path attractiveness and loyalty are calculated through the FISs 'path-attractiveness' and 'loyalty', respectively. Then, other bees can join a partial solution according to Eq. (11). After this decision-making process, bees' assignments are updated by the selected partial solutions.

At the end of each iteration, a complete path of the artificial network refers to a specific flight-to-gate assignment found by a bee of the colony. All the solutions found are evaluated referring to the associated fitness value, and the best assignment is saved. Then, a new iteration starts searching for new solutions until the maximum number of iterations is reached. In the following pseudocode, we have summarized the application of FBCO to the gate assignment problem.

```
Initialize the number of iterations I and bees B
for each iteration
  for each scheduled flight
    // do forward step
    for each bee in the hive
      evaluate the attractiveness of current gates based on the 'node-attractiveness' FIS
      choose the next gate through roulette selection
      construct partial flight-to-gate assignment
    end for each bee
    // do backward step
    for each constructed partial assignment
      evaluate the 'badness' of the constructed partial assignment
      evaluate the assignment attractiveness based on the 'path-attractiveness' FIS
      evaluate the loyalty based on the 'loyalty' FIS
    end for each partial solution
    evaluate the number of joining bees for each partial assignment
    update bees' partial assignments
  end for each scheduled flight
  store the best solution found so far
end for each iteration
```

5. Application to real cases

In this section, we show the application of the proposed method to two Italian airports, namely the Milan-Malpensa international airport, in the following called Malpensa, and the Turin international airport, in the following called Caselle.

5.1. Malpensa airport

Malpensa airport is of strategic importance for both Italy and Europe: in 2012, Malpensa airport was ranked second in Italy after Rome-Fiumicino airport for overall passenger traffic, with about 18.5 million passengers, and in the first place for freight traffic, with 414,317 tons. It is equipped with two terminals, one for international flights and one for domestic flights, and an area for freight, 65 gates throughout.

We have considered the scheduling of 178 flights in May 2012. The proposed method has been applied considering the structure of the airport and, in particular, an additional constraint related to the assignment of a flight to international or domestic gates based on its origin/destination. The compatibility between gate and airplane has been determined according to Malpensa Airport Regulations. Finally, we have found empirically that the coefficient would be "neutral", in the sense that it would not give preference to any objective for a value of 4.5e+05.

By using the Bee Colony Optimization process, we have obtained the results depicted in Figs. 6 and 7. From Fig. 6, we can observe that the variable δ_1 has a relevant role in the decision-making process: starting from $\delta_1 = 0.4$, the more the δ_1 value increases, the more the method gives importance to the minimization of TWD, which decreases up to 22% for $\delta_1 = 1$. On the other hand, Fig. 7 shows that the more the $\delta_1$value increases, the more the number of flights assigned to RG increases. For $\delta_1 \geq 0.6$ this value even overcomes the Malpensa schedule. Thus, a decision should be made for $\delta_1$in the range [0.4,0.6], where the objective functions have a significant variation, still maintaining their optimality.

In Fig. 8 we reported also the Pareto front for the proposed method. It is worth noting that solutions found by BCO are nearly always better than the values related to current scheduling of Malpensa. In fact, the total walking distance is always lower than the actual values in Malpensa (dotted line in Fig. 6), and the number of flights assigned to remote gates is less than the Malpensa schedule up to $\delta_1 = 0.6$.

5.2. Caselle airport

The Caselle Airport is the tenth airport in Italy for number of passengers; in particular, the overall passenger traffic in 2016 was 620,638 with a percentage increase of 11.50% with respect to 2015. The airport is organized into three levels: levels

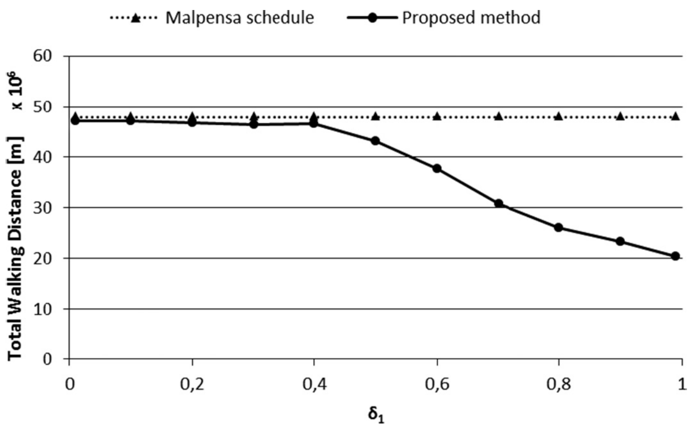

Fig. 6. Comparison of Total Walking Distance for different δ_1 values.

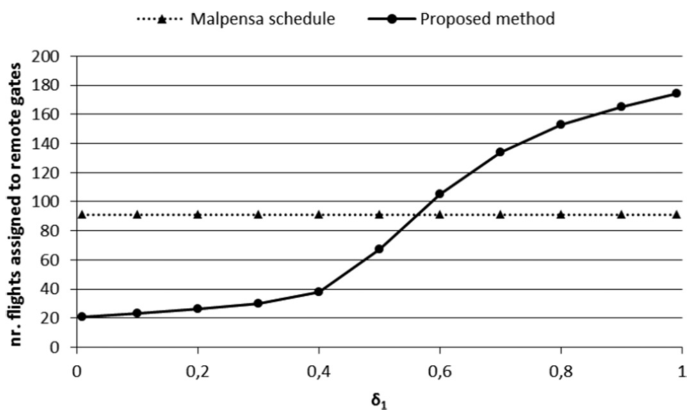

Fig. 7. Number of flights assigned to remote gates for different δ_1 values.

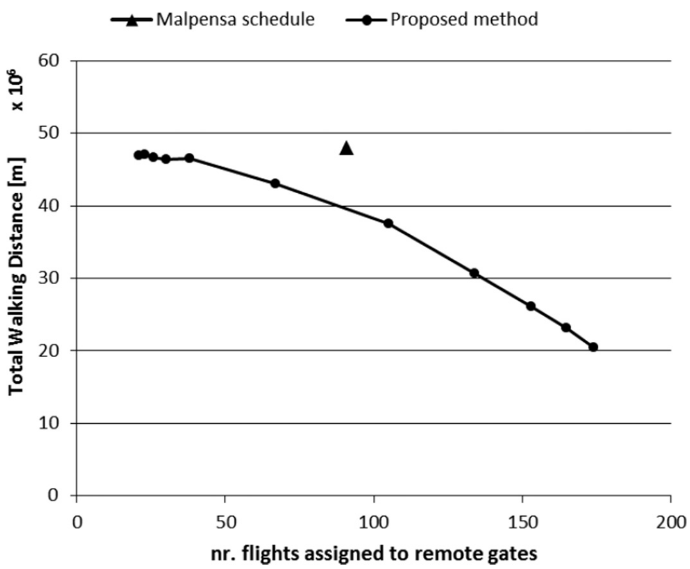

Fig. 8. Pareto front.

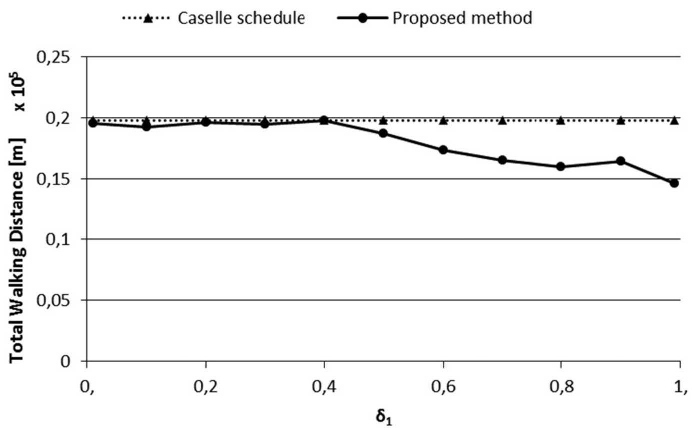

Fig. 9. Comparison of Total Walking Distance for different δ_1 values.

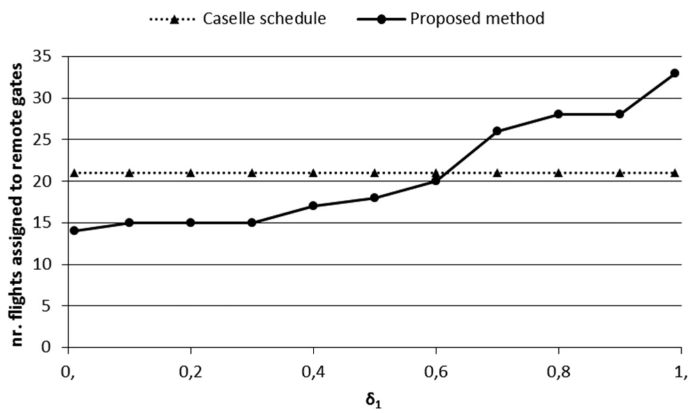

Fig. 10. Number of flights assigned to remote gates for different δ_1 values.

0 and 1 are devoted to arrivals and departures, respectively while at the third level are located a relax area and some restaurants. At level 0, there are 22 gates, divided into remote and fixed gates. Currently, Caselle Airport links ten European hubs – Rome, Amsterdam, Frankfurt, Munich, Paris, Istanbul, London, Brussels, Madrid, and Barcelona.

For this work, we have considered the scheduling of 47 flights in February 2016. Even in this case, we have applied the proposed method considering the structure of the airport and the constraint related to the assignment of a flight to international or domestic gates based on its origin/destination. The compatibility between gate and airplane has been determined based on the Service Charter of the Caselle Airport. We found that, in this case, the most suitable value of the coefficient β was 3.00e+04.

Like in Malpensa case, we have used the Bee Colony Optimization process to obtain the results depicted in Figs. 9 and 10, in terms of RG and TWD.

Even in this case, starting from $\delta_1 = 0.4$, the TWD decreases up to 25% for $\delta_1 = 1$. Fig. 10 shows that, for $\delta_1 \geq 0.6$, this value overcomes the Caselle schedule. Thus, also in the case of Caselle airport, a decision should be made for δ_1 in the range [0.4, 0.6].

In Fig. 11, we have reported the Pareto front for the proposed method. Solutions provided by the BCO are nearly always better than the values related to current scheduling of Caselle airport. In fact, the TWD is always lower than the actual values in Caselle (dotted line in Fig. 6), and the number of flights assigned to remote gates is less than the Caselle schedule up to $\delta_1 = 0.6$.

6. Comparison with the Ant Colony Optimization

The Ant Colony Optimization (ACO) is an efficient algorithm, probably the most well-known, of the Swarm Intelligence methods, a family of nature-inspired metaheuristic optimizations to which also belongs the Bee Colony Optimization. Dorigo (1992) proposed the ACO in his Ph.D. thesis as an algorithm, based on the behavior of ants seeking a path between

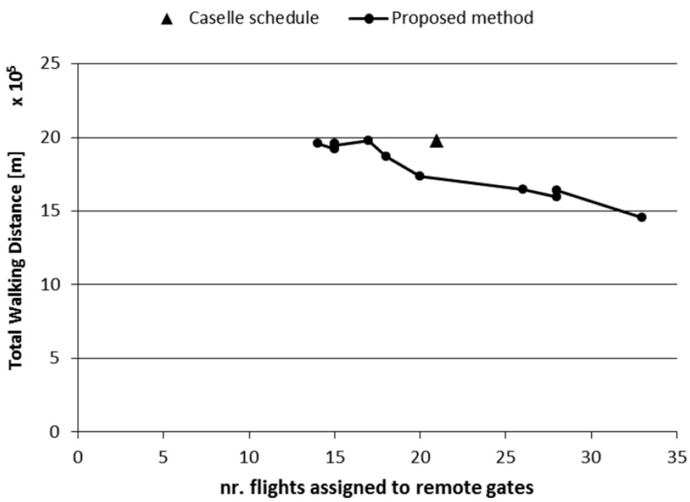

Fig. 11. Pareto front.

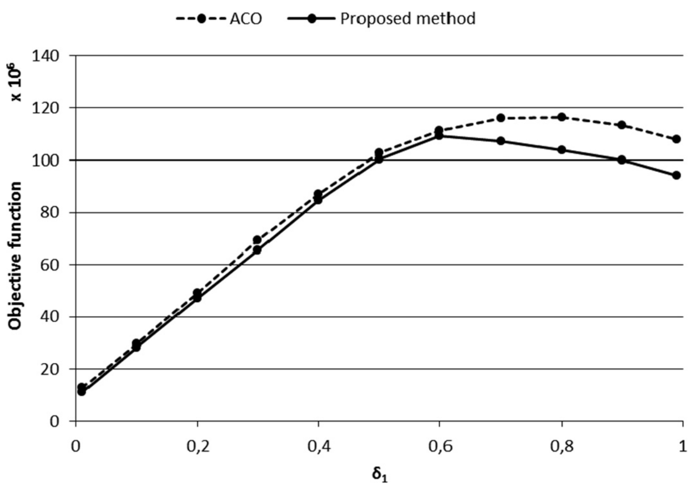

Fig. 12. Comparison of objective function values – Malpensa airport.

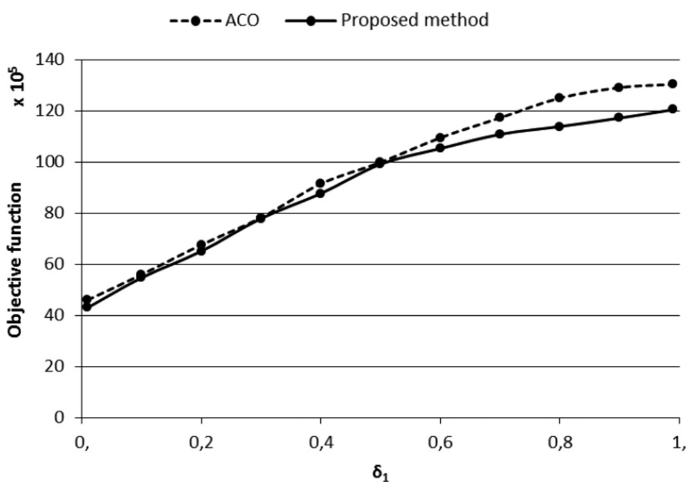

Fig. 13. Comparison of objective function values – Caselle airport.

their colony and a source of food, suitable to search for an optimal path in a graph. Further details can be found in the seminal work by Dorigo and Gambardella (1997).

In the following Figs. 12 and 13, we have reported the comparison of ACO and Fuzzy BCO (FBCO) in terms of objective functions, for Malpensa and Caselle, respectively. It appears that the FBCO outperforms the ACO in both cases: the percent of average improvement of the objective function values is 7.2% in the first case and 4.8% in the second case.

7. Conclusions

In this paper, to solve the flight-gate assignment problem we have considered the minimization of two criteria: the total walking distance and the number of flights assigned to the remote gates, subject to the compatibility constraints. Thus, we needed to carry out a multicriteria method having good capabilities in solving high-order combinatorial problems, like overall combinations in flight assignment to a gate. Consequently, we presented a metaheuristic approach based on the Fuzzy Bee Colony Optimization (FBCO). This method has shown to be efficient when dealing with high-order combinatorial optimization: the results highlight the effectiveness of the proposed method when compared to the actual flight schedule of two Italian airports, a large one (Milano Malpensa) and another medium sized (Torino Caselle), according to the European standards.

The solutions found by the FBCO are usually better than the current schedules of Malpensa and Torino Caselle. Moreover, we compared the proposed method with another efficient and well-known method of Swarm Intelligence, the Ant Colony Optimization (ACO). We found that the FBCO improved average values of the objective functions of ACO by at least 4.8%. Thus, the proposed method seems a good tool to support decision-making in flight scheduling. Further developments cover the adaptation of the method to the dynamic gate assignment problem, considering more constraints related to airline companies' preferences and agreements. Moreover, in future works, we will consider more criteria for a better evaluation of the quality of the assignment.

References

Abbass, H.A., 2001. MBO: Marriage in honey bees optimization – a Haplometrosis polygynous swarming approach. Proceedings of the Congress on Evolutionary Computation, Seoul, South Korea 1, 207–214.
Afshar, A., Bozorg Haddad, O., Mariño, M.A., Adams, B.J., 2007. Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. J. Franklin Inst. 344 (5), 452–462.
Babic, O., Teodorović, D., Tosic, V., 1984. Aircraft stand assignment to minimize walking. J. Transp. Eng. 110, 55–66.
Bard, J., Yu, G., Arguello, M., 2001. Optimizing aircraft routings in response to groundings and delays. IIE Trans. 33, 931–947.
Baron, P., 1969. A simulation analysis of airport terminal operations. Transp. Res. 3, 481–491.
Baker, J.E., 1987. Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms and their Application. L. Erlbaum Associates, Hillsdale, New Jersey, pp. 14–21.
Benayoun, R., deMontgolfier, J., Tergny, J., Larichev, O., 1971. Linear programming with multiple objective functions: step-method (STEM). Math. Program. 1, 366–375.
Bihr, R., 1990. A conceptual solution to the aircraft gate assignment problem using 0–1 linear programming. Comput. Ind. Eng. 19, 280–294.
Bolat, A., 2000. Models and a genetic algorithm for static aircraft gate assignment problem. J. Oper. Res. Soc. 52, 1107–1120.
Camazine, S., Sneyd, J., 1991. A model of collective nectar source by honey bees: self-organization through simple rules. J. Theor. Biol. 149, 547–571.
Cheng, Y., 1997. A knowledge-based airport gate assignment system integrated with mathematical programming. Comput. Ind. Eng. 32, 837–852.
Cheng, C.-H., Ho, S.C., Kwan, C.-L., 2012. The use of meta-heuristics for airport gate assignment. Exp. Syst. Appl. 39, 12430–12437.
Davidović, T., Selmić, M., Teodorović, D., Ramljak, D., 2012. Bee colony optimization for scheduling independent tasks to identical processors. J. Heuristics 18 (4), 549–569.
Davidović, T., Teodorović, D., Selmić, M., 2014. Bee colony optimization part I: the algorithm overview. Yugoslav J. Oper. Res. 25 (1), 33–56 (2∗∗∗015).
Dimitrijević, B., Teodorović, D., Simić, V., Selmić, M., 2011. Bee colony optimization approach to solving the anticovering location problem. J. Comput. Civ. Eng. 26 (6), 759–768.
Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2004. New heuristics for the over constrained airport gate assignment problem. J. Oper. Res. Soc. 55, 760–768.
Dorigo, M., 1992. Optimization, Learning and Natural Algorithms (Ph.D. thesis). Politecnico di Milano, Italy.
Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1 (1), 53–66.
Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling robust flight gate scheduling as a clique partitioning problem. Transp. Sci. 42, 292–301.
Dorndorf, U., Jaehn, F., Pesch, E., 2012. Flight gate scheduling with respect to a reference schedule. Ann. Oper. Res. 194, 177–187.
Drexl, A., Nikulin, Y., 2008. Multicriteria airport gate assignment and Pareto simulated annealing. IIE Trans. 40, 385–397.
Drias, H., Sadeg, S., and Yahi, S. 2005. Cooperative bees swarm for solving the maximum weighted satisfiability problem. In: LNCS: Computational Intelligence and Bioinspired Systems, vol. 3512, pp. 318–325.
Fonseca, R., Paluszewski, M., Winter, P., 2010. Protein structure prediction using bee colony optimization metaheuristic. J. Math. Model. Algor. 9 (2), 181–194.
Geoffrion, A., Dyer, J., Feinberg, A., 1972. An interactive approach for multicriterion optimization with an application to the operation of an academic department. Manage. Sci. 19 (4–Part–1), 357–368.
Gigerenzer, G., Selten, R., 2002. Bounded Rationality. MIT Press, Cambridge.
Gu, Y., Chung, C.A., 1999. Genetic algorithm approach to aircraft gate reassignment problem. J. Transp. Eng. 125 (5), 384–389.
Haghani, A., Chen, M.-C., 1998. Optimizing gate assignments at airport terminals. Transp. Res. A: Policy Pract. 32 (6), 437–454.
Hamzwawi, S., 1986. Management and planning of airport gate capacity: a microcomputer-based gate assignment simulation model. Transp. Plann. Technol. 11, 189–202.
Hassounah, M., Steuart, G., 1993. Demand for aircraft gates. Transp. Res. Rec. 1423, 26–33.
Hu, X., Di Paolo, E., 2009. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. In: Goh, C.K., Ong, Y.S., Tan, K.C. (Eds.), Studies in Computational Intelligence, vol. 171, pp. 71–89.
Karaboğa, D. 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report. Erciyes University, Engineering Faculty Computer Engineering Department Kayseri/Turkiye.
Lam, S., Cao, J., Fan, H., 2002. Development of an intelligent agent for airport gate assignment. J. Airport Transp. 7 (2), 103–114.

Li, W., Xu, X., 2012. Optimized assignment of airport gate configuration based on immune genetic algorithm. In: Measuring Technology and Mechatronics Automation in Electrical Engineering, pp. 347–355.
Lučić, P., Teodorović, D., 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: The Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis. University of Azores, Sao Miguel, Azores Islands, Portugal, pp. 441–445.
Lučić, P., Teodorović, D., 2003. Computing with bees: attacking complex transportation engineering problems. Int. J. Artif. Intell. Tools 12, 375–394.
Mangoubi, R., Mathaisel, D., 1985. Optimizing gate assignments at airport terminals. Transp. Sci. 19, 173–188.
Miller, B.L., Goldberg, D.E., 1995. Genetic algorithms, tournament selection, and the effects of noise. Complex Syst. 9, 193–212.
Nakrani, S., Tovey, C., 2004. On honey bees and dynamic server allocation in internet hosting centers. Adapt. Syst. 12 (3–4), 223–240.
Narasimhan, H., 2009. Parallel artificial bee colony (PABC) algorithm. In: World Congress on Nature & Biologically Inspired Computing. IEEE, pp. 306–311.
Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Zaidi, M., 2006. The bees algorithm – a novel tool for complex optimization problems. Proc. 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006). Elsevier, Cardiff, Wales, UK, pp. 454–459.
Prem Kumar, V., Bierlaire, M., 2014. Multi-objective airport gate assignment problem in planning and operations. J. Adv. Transp. 48 (7), 902–926.
Rezaei, J., 2015. Best-worst multi-criteria decision-making method. Omega 53, 49–57.
Roy, B., 1968. La méthode ELECTRE. Revue d'Informatique et de Recherche Opérationelle (RIRO) 8, 57–75.
Sato, T., Hagiwara, M., 1997. Bee system: finding solution by a concentrated search. In: Proc. IEEE International Conference on Systems, Man, and Cybernetics 'Computational Cybernetics and Simulation', Orlando, FL, USA, pp. 3954–3959.
Şeker, M., Noyan, N., 2012. Stochastic optimization models for the airport gate assignment problem. Transp. Res. E: Logist. Transp. Rev. 48 (2), 438–459.
Srihari, K., Muthukrishnan, R., 1991. An expert system methodology for aircraft-gate-assignment. Comput. Ind. Eng. 21, 101–105.
Stadler, W., 1979. A survey of multicriteria optimization, or the vector maximum problem. J. Optim. Theory Appl. 29, 1–52.
Teodorović D., Dell'Orco M., 2005. Bee colony optimization – a cooperative learning approach to complex transportation problems. In: Advanced OR and AI Methods in Transportation, Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, pp. 51–60.
Teodorović, D., Lučić, P., Markovic, G., Dell'Orco, M., 2006. Bee colony optimization: principles and applications. Neural Netw. Appl. Electr. Eng., 151–156
Wedde, H.F., Farooq, M., Zhang, Y., 2004. BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. In: LNCS: Ant Colony Optimization and Swarm Intelligence, vol. 3172, pp. 83–94.
Wei D.-X., Liu C.-Y., 2009. Fuzzy model and optimization for airport gate assignment problem. In: Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '09), Shanghai, China, pp. 828–832.
Wirasinghe, S., Bandara, S., 1990. Airport gate position estimation for minimum total costs—approximate closed form solution. Transp. Res. 24B, 287–297.
Xu, J., Bailey, G., 2001. The airport gate assignment problem: mathematical model and a tabu search algorithm. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. IEEE, p. 3032.
Yan, S., Chang, C., 1998. A network model for gate assignment. J. Adv. Transp. 32, 176–189.
Yan, S., Huo, C., 2001. Optimization of multiple objective gate assignments. Transp. Res. 35A, 413–432.
Yan, S., Shieh, C.-Y., Chen, M., 2002. A simulation framework for evaluating airport gate assignments. Transp. Res. 36, 885–898.
Yan, S., Tang, C.-H., 2007. A heuristic approach for airport gate assignments for stochastic flight delays. Eur. J. Oper. Res. 180 (2), 547–567.
Yang, C., Chen, J., and Tu, X., 2007. Algorithm of fast marriage in honeybees optimization and convergence analysis. In: Proc. IEEE International Conference on Automation and Logistics, Jinan, China, pp. 1794–1799.
Yang, X.-S., 2005. Engineering optimizations via nature-inspired virtual bee algorithms. In: Mira, J., Alvarez, J.R. (Eds.), Proc. IWINAC International Work-conference on the Interplay between Natural and Artificial Computation, IWINAC 2005. LNCS, vol. 3562. Las Palmas de Gran Canaria, Canary Islands, Spain, June 15–18, pp. 317–323.
Zadeh, L., 1965. Fuzzy sets. Inf. Control 8, 338–353.
Zadeh, L., 1999. From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions. IEEE Trans. Circ. Syst.-I: Fundam. Theory Appl. 45, 105–119.
Zimmermann, H.J., 1996. Fuzzy Sets Theory – And Its Applications. Kluwer Academic Publisher, Assinippi Park, Norwell, MA.