Artificial Bee Colony Algorithm for Solving the Flight Disruption Problem

Tanja Šarčević[1], Ana Paula Rocha[2], Antonio J.M. Castro[3]

[1] Vienna University of Technology, Faculty of Informatics, Austria
[2] LIACC, DEI/FEUP, University of Porto, Portugal
[3] LIACC, University of Porto, Portugal

Abstract. This paper presents the optimization algorithm Artificial Bee Colony (ABC) firstly introduced by in 2005 and proposed for optimizing numerical problems. ABC is the swarm-based meta-heuristic algorithm inspired by intelligent behavior of honey bee colonies. In this paper, ABC has been applied on solving the flight disruption problem, by swapping aircraft and/or cancelling/delaying flights, and its performance has been shown through experimentation. The environment and data for experiments are provided by MASDIMA, Multi-Agent System for DIsruption MAnagement developed by LIACC (Laboratory of Artificial Intelligence and Computer Science).

Keywords: Artificial Bee Colony; swarm intelligence; optimization; flight disruption; air traffic control; disruption management; operations control center; multi-agent system

1 Introduction

Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervis Karaboga in 2005 [1], motivated by the intelligent behavior of honey bees. It is a simple algorithm, and only uses common control parameters such as colony size and maximum cycle number. ABC as an optimization tool provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees over time. The bees' aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation processes. A couple of similar approaches to both numerical and combinatorial problems were introduced based on foraging behavior of honey bees. Chong et al. in 2006 proposed a bee colony optimization algorithm applied on job shop scheduling [2]. Another example is Pham et al. (2005) with Bees Algorithm for both numerical and combinatorial problems [3]. Lučić and Teodorović in 2001 introduced a Bee System for solving difficult combinatorial optimization problems [4], and numerous others. As ABC was successfully applied to many different problems, it is the aim of this paper to discuss its application to flight disruption problem. The problem is arising in Airline Operations Control Centers (AOCC) in dealing with numerous disruptions that distort the flight operations. By disruptions we are talking about events such as weather, aircraft malfunction, late arrival of incoming aircraft, missing crewmember, that implies a delay in one or more flights. When solving the flight disruption problem, we look for recovery actions such as, swapping aircraft, assign reserve aircraft and/or cancelling/delaying flights, trying that operational costs as well as delays are the lowest possible. Each flight-to-aircraft assignment is causing different operational costs due to distances made by each aircraft, delays, fuel prices, and other costs.

The specific task of efficiently solving the flight disruption problem is just one of the numerous tasks that AOCCs are dealing with. Usually, a disruptive event affects not only one, but three dimensions, – aircraft, crew and passenger, and solutions should be find for all of them. Typically, the Airline Operations Control Center deals with disruptions sequentially, which means solving each of mentioned dimensions individually and one after another respectively. Solutions obtained this way give significant imbalance to the importance of the dimensions in the overall solution, giving aircraft part advantage over the other two dimensions, since aircraft is, usually, the first dimension to be solved in the sequence.

As opposed to sequential approach, the system, MASDIMA (MultiAgent System for DIsruption MAnagement) [5], brings a distributed approach with a Multi-Agent System (MAS) paradigm and delivers integrated solution with all dimensions equally considered. Each of the three dimensions is dealt with by specified agents. The events that cause disruption, like aircraft malfunction, weather and other restrictions, crew problems, passenger and baggage delay, are detected in real-time by the system itself and their impact is then assessed in the operational plan of the aircraft. Figure 1 shows the MASDIMA architecture.

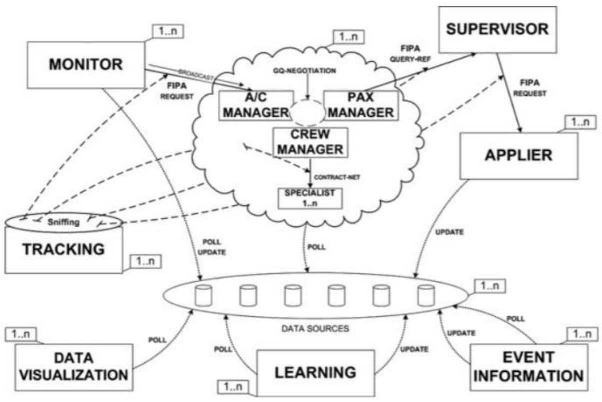

Fig.1.MASDIMA architecture

The work reported in this paper aims to solve the aircraft dimension problem (flight disruption problem), and the proposed approach will be implemented as a Specialist Agent belonging to the Aircraft Manager agent team of MASDIMA. The rest of this paper is as follows. Section 2 introduces swarm-based paradigm, the Artificial Bee Colony algorithm and its application on solving the flight disruption problem. Furthermore, section 3 describes the experiments done and presents a discussion about the algorithm's performance and impact of different parameters. Finally, section 4 brings the most important conclusions from the experiments done and points out some directions of future work.

2 Solving the Flight Disruption Problem

2.1 Swarm-based algorithms

Swarm intelligence is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. The discipline focuses on collective behavior of many individuals which results from their local behavior within their neighborhood and their global communication in range of the whole population. In nature, these systems are commonly used to solve problems such as effective foraging for food, prey evading, or colony relocation. Examples of such natural systems are colonies of ants and other insects, bird flocks, schools of fish, etc.

The properties of a swarm intelligence system can be enumerated in the following points:

- multitude - composed of many individuals
- homogeneity - the individuals are relatively homogeneous, i.e. either identical or belonging to a few typologies

- simplicity - individuals have simple behavioral rules by which they exploit only local information
- self-organization - the overall behavior of the system results from the interactions of individuals with each other and with environment

The main characterizing property of the swarm intelligence system is an absence of necessity for a coordinator with a general knowledge of the swarm state. The swarm system come up with a coordinated behavior without a centralized individual controlling the swarm.

Most popular and studied computational systems that are inspired by collective intelligence are Ant Colony Optimization and Particle Swarm Optimization. But Artificial Bee Colony (ABC) optimization also belongs to the same group of swarm-based, nature inspired algorithms.

2.2 ABC algorithm

The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic algorithm introduced by Dervis Karaboga in 2005[1] for optimizing numerical problems. The algorithm is inspired by intelligent foraging behavior of honey bees. The model for the algorithm was proposed by Tereshko and Loengarov in 2005[6]. The model consists of three essential components:

1. employed bees
2. unemployed honey bees
3. food sources

The aim of the bee colony is to find rich food sources close to the hive. The model also defines two types of behaviors that are necessary for self-organizing and collective intelligence:

- recruitment of foragers to rich food sources
- abandonment of poor food sources

Bees are continuously changing their environment in the process of their search for food sources, and they are capable of continuously adapting to the new environment.

In ABC, artificial bees are in a search for a good solution of a given problem, i.e. a rich food source. To apply ABC, it is necessary to define the problem that can be converted to the problem of finding the parameter vector that minimizes the objective function. Artificial bees randomly discover initial set of food sources (potential solution vectors), and iteratively evaluate them and improve this set by searching for better food sources in their neighborhood and abandoning poor food sources.

The colony of artificial bees contains three groups of bees: employed bees associated with specific food sources, onlooker bees watching the dance of employed bees within the hive to choose a food source, and scout bees randomly searching for food sources. Both onlookers and scouts are also called unemployed bees. Initially, all food source positions are discovered by scout bees. Thereafter, the nectar of food sources is exploited by employed bees and onlooker bees, and this continual exploitation will ultimately cause them to become exhausted. Then, the employed bee which was exploiting the exhausted food source becomes a scout bee in search of further food sources once again. In other words, the employed bee whose food source has been exhausted becomes a scout bee. In ABC, the position of a food source represents a possible solution to the problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of employed bees is equal to the number of food sources (solutions) since each employed bee is associated with one and only one food source.

The general scheme of the ABC algorithm is as follows:

```
Initialization phase
REPEAT
  Employed Bees Phase
  Onlooker Bees Phase
  Scout Bees Phase
  Memorize the best solution achieved so far
UNTIL (Reached maximum number of cycles)
```

1) Employed Bees Phase

Employed bees search for new food sources having more nectar within the neighborhood of the food source in their memory. They find a neighbor food source and evaluate its fitness. After finding the new food source, its fitness is calculated and a greedy selection is applied between old and new food source.

2) Onlooker Bees Phase

Unemployed bees consist of two groups of bees: onlooker bees and scouts bees. Employed bees share their food source information with onlooker bees waiting in the hive, and then onlooker bees probabilistically choose their food sources depending on this information. An onlooker bee chooses a food source depending on the probability value calculated using the fitness values provided by employed bees.

After a food source for an onlooker bee is probabilistically chosen, a neighborhood source is determined in the same way as in the previous phase, and its fitness value is computed. As in the employed bees phase, a greedy selection is applied between the two solutions. Hence, more onlookers are recruited to richer sources and positive feedback behavior appears.

3) Scout Bees Phase

The unemployed bees that choose their food sources randomly are called scouts. Employed bees whose solutions cannot be improved through a predetermined number of trials, specified by the user of the ABC algorithm and called "limit" or "abandonment criteria", become scouts and their solutions are abandoned. Then, the converted scouts start to search for new solutions, randomly. Hence, those sources which are initially poor or have been made poor by exploitation, are abandoned and negative feedback behavior arises to balance the positive feedback.

2.3 Application in solving the flight disruption problem

A solution to a disruptive flight, can be achieved by one of the following actions: swapping aircrafts, using reserve aircraft and/or delaying/cancelling a flight. To adapt the ABC algorithm to provide the best solution to disrupted flights, we represent a possible solution as an assignment of aircrafts to all operational flights, including disrupted ones. Aircrafts are initially, and at any point of the algorithm execution, assigned to a flight if and only if the aircraft is not already assigned to some other flight in the same time in the same schedule (i.e, solution). This restriction of the aircraft choice results in all the solutions being feasible, which implies algorithm calculates costs and finds the best one only among the feasible solutions. On the other hand, cancellation of the flight is also possible. If there is such schedule where for some flight the choice of the aircraft is impossible (e.g. because at that time no aircraft is available), then the flight is cancelled and it is penalized with a high number. Penalization for cancellation amounts to approximately 10% of the total calculated cost of a random solution.

After the parameters are set - number of iterations the algorithm will perform *NUMBER_OF_CYCLES*; size of the bee colony *COLONY_SIZE*; number of food sources *FOOD_SOURCES_N* (half the number of *COLONY_SIZE*); and number of trials *TRIALS* -, ABC randomly generates an initial population of food sources – the initial schedules. The initialization phase is followed by the *NUMBER_OF_CYCLES* iterations where Employed Bees Phase, Onlooker Bees Phase and Scout Bees Phase are being performing their tasks.

The first half of the bee population, employed bees, select distinct food sources and explore their neighborhood in search of better food sources. The neighbor of a food source is defined in the following way:

- Choose a random flight in the solution set
- Choose another available aircraft (from reserve or assigned to another flight)
- Change the aircraft of the chosen flight to the new one

If the new solution-plan (flight plus aircraft) obtained in this way has a better fitness value, the new solution-plan is selected and the old one abandoned. This behavior is the optimization part of the algorithm. The fitness of a solution is calculated based on the costs that this choice of solution-plan produces, shown in following formula:

$$fitness_i = (worstScore - \ objectiveFunction_i) * 100.0\ /bestScore$$

where worst score and best score are the maximum and minimum value of the objective function among the solution population, respectively, and *i* is the index of the solution. Objective function of each possible solution is calculated by following formula:

$$objectiveFunction_i = \sum_{aircrafts} (handlingCost_i + fuelCost_i + maitenance_i + ATC_i + takeOffCost_i + landingCost_i + parkingCost_i)$$

As shown, the objective function sums up all the costs related to the choice of the aircraft for the flight, giving them the same weight. As mentioned before, the objective function value can be additionally penalized in case the cancellation of the flight is necessary.

After the better solutions are chosen, onlooker bees start their work. This other half of the bee population selects their food sources based on information about food sources received from employed bees. The information is given by selection probability attribute calculated from the fitness of the solution, according to the following formula:

$$selectionProbability_i = 0.9 * \left(\frac{fitness_i}{maxFitness}\right) + 0.1$$

After all onlookers bees select their food sources, the same technique as in Employed Bees Phase is applied for searching for better sources in the neighborhood.

Every solution contains information of how many times the bee tried to improve it but failed, in a simple counter. If this counter exceeds the predefined parameter TRIALS, the bee who is exploring it becomes a scout bee and finds a new random solution. Due to performance reasons, a solution that exceeds the number of trials is only being abandoned if it is not the best solution found so far. This modification is introduced because of the nature of the problem - the solution space is too big and bees don't manage to improve the already very good solutions which results in abandonment of the best solutions, intuitively undesired behavior.

The pseudo code of the procedure for a flight disruption problem using ABC, that resumes the description given above, is the following:

- Set the parameters *NUMBER_OF_CYCLES*, *COLONY_SIZE* and *TRIALS*
- Set random *COLONY_SIZE/2* schedules
- FOR (each candidate solution-plan)
  - Calculate fitness
  - Calculate selection probability
- END FOR
- Memorize best candidate solution-plan (with best fitness)
- REPEAT
  - FOR (each candidate solution-plan)
    - Find neighbor solution-plan
    - Calculate neighbor's fitness
    - Apply greedy algorithm to choose between current and new candidate solution-plan
    - Calculate better solution's fitness and selection probability
    - Update trial number
  - END FOR
  - REPEAT
    - Use selection probability to choose some of existing solution-plan
    - Find neighbor solution-plan
    - Calculate neighbor's fitness
    - Apply greedy algorithm to choose between current and new solution-plan
    - Calculate better solution's fitness and selection probability

        - Update trial number
    - UNTIL (reached *COLONY_SIZE/2* iterations)
    - Memorize best solution so far
    - FOR (each candidate solution-plan)
        - IF (trial >*TRIALS* and this solution-plan is not the best one*)*
            - Find another random solution and forget the current one
            - Calculate fitness and selection probability
    - END FOR
    - Memorize best solution so far
- UNTIL (*NUMBER_OF_CYCLES* iterations reached)

3 Experiments and discussion

3.1 Settings

The implementation of ABC and experiments are done in the environment of MASDIMA [5], with real data that contains information about flights and aircrafts from a period of one month.

MASDIMA is a disruption management system based on the Multi-Agent System paradigm (MAS). As shown in Figure 1, agents in the system are separated in three main layers by their functionality: supervisor, manager agents and specialist agents. Each dimension has its own manager agent whose main task is to negotiate with other dimension manager agents (Passenger Manager, Crew Manager and Aircraft Manager) and the supervisor over a several rounds to find a good integrated solution. Selection of the best candidate solution is done by choosing among the solutions found the one with the lowest cost and delay. Specialist agents are also related to each dimension - Passenger Specialist, Crew Specialist and Aircraft Specialist. Each one of them has a specific expertise that is run by different resolution algorithms. These algorithms are delivering the best candidate solutions from their own perspective, i.e. using costs regarding only their dimension into.

Our proposed ABC algorithm was implemented as a MASDIMA specialist agent working in the Aircraft Manager Agent team. Our proposed ABC algorithm was tested by changing each one of the predefined parameters (the number of cycles, the size of the colony, the trial limit) to see their impact on the performance.

3.2 Influence of the number of cycles

This parameter says how long we will let algorithm run. The solution is not likely to reach the global optimum, so it makes sense to stop the algorithm in the phase when the best solution is not improving anymore, or starts to improve very slowly.

Table 1. Test results for different number of cycles

Number of cycles	Best solution's cost	Improvement
20	95894891	0.08%
50	95264194	0.31%
100	95139913	0.46%
200	94782343	1.7%
400	94337640	1.82%
600	92263797	3.81%
800	92035075	4.08%
1000	92037320	4.03%

Table 1 shows the results obtained by several runs of the algorithm while changing the total number of cycles, and remaining the other two parameters with the same value. In this experiment, colony size is set to 20 bees and limit of trials on 10% of the total cycle number.

The improvement of the solution is noticeably increasing from 400th until 800th cycle, after which we don't see a significant change in the solution improvement.

3.3 Influence of the size of the colony

Table 2. Results of the experiment over colony size

Colony size	Best solution	Improvement
10	94443398	1.51%
20	95108117	0.95%
44	94153595	1.47%
72	94276091	1.52%

The experiment in this case was run on 400 cycles and trial limit of 15 cycles. In Table 2 can be observed that colony size does not influence the result much. It is important to mention that final best result very much depends on the best result from the first population of solutions.

Another note is that increasing the size of the colony, the runtime of the algorithm increases almost linearly. Therefore, it is better option to leave the number of bees reasonably low for the best performance.

3.4 Influence of the trial limit

Trial limit says how fast solutions will get exhausted. Setting this parameter on a high value limits the solution space the bees are exploring. However, setting it to a low value, bees "give up" too fast on trying to improve the solution and the solution search becomes random.

Table 3. Results of the experiment over trial limit

Trial limit	Best solution	Improvement
5	95614998	0.44%
20	94933019	0.81%
50	94383364	1.49%
125	94707546	1.25%
300	95829453	0.39%

The third experiment is testing the impact of the trial limit on the performance result. The experiment has run in 400 cycles and with a population of 20 bees. The results of this experiment are shown in Table 3. The best performance is noticed on medium values which "give space" for solution to be improved, but still restrict the randomness.

4 Conclusions

ABC algorithm has been modified for application on the flight disruption problem and its performance has been analyzed in one month operational flight data. By the results obtained, we can conclude that the ABC algorithm can be successfully used for solving flight disruption optimization problems. Further modifications are welcome on the implemented solution to improve performance. Above all, smart choice of heuristics can contribute in faster reaching of better solutions. Due to the wide solution space, the reduction of random factors in the implementation might be a good idea. For example, strategy for choosing the initial population of solution instead of choosing randomly distributed one, or applying additional methods for smarter choice of neighborhood food sources can be investigated in the future work.

5 References

1. Karaboga, D, (2005). An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
2. Chong, C. S., Sivakumar, A. I., Malcolm Low, Y. H., Gay, K. L. (2006). A bee colony optimization algorithm to job shop scheduling. In Proceedings of the 38th conference on Winter simulation WSC '06, pages 1954-1961, California.
3. Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M. (2005). The bees algorithm. Technical report, Manufacturing Engineering Centre, Cardiff University, UK.
4. Lucic, P. and Teodorovic, D. (2001). Bee system: Modeling combinatiorial optimization transportation engineering problems by swarm intelligence.
5. Castro, A.J.M., Rocha, A.P., Oliveira, E. (2014). A New Approach for Disruption Management in Airline Operations Control. Springer-Verlag Berlin Heidelberg.
6. Tereshko, V., Loengarov, A. (2005), Collective decision-making in honey bee foraging dynamics, Computing and Information Systems, 9 (3): 1-7, University of the West of Scotland, UK.