Artificial Bee Colony Optimization Algorithm for Solving Flight Schedule Problem

Tanja Šarčević

Faculty of Engineering, University of Porto

Department of Computer Engineering

Porto, Portugal

***Abstract*—This paper presents the optimization algorithm Artificial Bee Colony (ABC) firstly introduced by D. Karaboga in 2005. and proposed for optimizing numerical problems. In this paper, ABC has been applied on solving a flight schedule problem and its performance has been shown. 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 scheduling; disruption management; Multi-Agent System

I. INTRODUCTION

Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervis Karaboga in 2005, motivated by the intelligent behavior of honey bees. It is simple, and uses only 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 with time and 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 process.

ABC is firstly proposed for optimizing numerical problems by Karaboga in 2005 [1]. A couple of similar approaches to both numerical and combinatorial problems were introduced based on foraging behavior of honey bees. One example is Pham et al. (2005) with Bees Algorithm for both numerical and combinatorial problems [3]. In 2005., Yung developed a Virtual Bee Algorithm for solving numerical optimization problems. Lučić and Teodorović in 2001. introduced a Bee System for solving difficult combinatorial optimization problems, Chong et al. in 2006 proposed a bee colony optimization algorithm applied on job shop scheduling [2], 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 scheduling problem. The problem is arising in Airline Operations Control centers, not only in scheduling flights on regular basis, but with dealing with numerous disruptions that distort the flight operations as well. One of the modern systems for disruption management in traffic is MASDIMA (Multi-Agent System for DIsruption MAnagement).

MASDIMA is a disruption management system with multiple tasks and with a new, more intelligent approach to the disruption management. In means of disruptions we are talking about weather, aircraft malfunction, late arrival of incoming aircraft, missing crewmember... Disruptive events affect at least three dimensions - aircraft assigned to the flight, the crew assignment and passengers' satisfaction. Typically, the Airline Operations Control Centers (AOCC) deal with disruptions sequentially, which means solving each of dimensions - aircraft, crew and passenger - individually and one after the other respectively. That means finding the suitable aircraft first for the delayed flight, considering that solution the qualified crew is being found and assigned to the delayed flight. At the very end, passengers are accommodated according to the estimated flight departure delay caused by aircraft and crew choice from the previous steps. This kind of approach brings an obvious imbalance to the importance of all dimension and in overall solution. MASDIMA's approach with Multi-Agent System (MAS) paradigm delivers integrated solution with all dimensions equally taken into account.

MASDIMA contains numerous features helpful and essential for modern disruption management, not only for airlines, but for any means of transportation. The operational plan is monitored in real-time on a 3D virtual display allowing to track flights. The events that cause disruption like aircraft malfunction, weather and other restrictions, crew problems, passenger and baggage delay are detected in real-time by system itself and their impact is then assessed in operational plan of the aircraft. MASDIMA allows human interaction in decision making such that all candidate solutions are represented to human for feedback and approval as well as contains the customer interaction system.

II. Artificial Bee Colony Algorithm

A. *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 such systems are commonly used to solve problems such as effective foraging for food, prey evading, or colony re-location. Examples of such systems are colonies of ants and other insects, bird flocks, schools of fish, etc.

Properties of swarm intelligence system:

- 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 has 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. Artificial Bee Colony (ABC) optimization as well belongs in the same group of swarm-based, nature inspired algorithms.

B. *ABC algorithm*

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

- employed bees
- unemployed honey bees
- food sources

The aim of the bee colony is to find rich food sources close to the hive. Model also defines two main models 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 continuous adapting to the new environment.

In ABC the artificial bees are in a search for a good solution of a given problem, i.e. a rich food source. In order 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 by searching for better food sources in the defined neighborhood and abandoning poor 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 searching for food sources randomly. 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 then evaluate its fitness. After producing 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. 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. In ABC, an onlooker bee chooses a food source depending on the probability values 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 who 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.

C. *Application on the flight scheduling problem*

In order to adapt the ABC algorithm to flight scheduling problem, we represent possible solution as an assignment of aircrafts to all operating flights. 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. This restriction of the aircraft choice results in all the schedules (i.e. solutions) being feasible, which makes algorithm calculate costs and finding the best one only among the feasible solutions. On the other hand, cancellation of the flight is as well possible. If there is such schedule where for some flight the choice of the aircraft is impossible (e.g. because at that time no aircrafts are available according to schedule), then the flight is being cancelled and it is penalized with a high number. Penal for cancellation amounts to approximately 10% of the total calculated cost of a random solution.

After the parameters are set, i.e. number of iterations we want algorithm to 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 generates randomly distributed initial population of food sources - 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 performed.

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 the food source is defined in the following way:

- Choose a random flight in the solution set
- Choose another available aircraft
- Change the aircraft of the chosen flight to the new one

If the new schedule obtained in this way has the better fitness value, the new schedule is being selected and the old one abandoned. This behavior is the optimization part of the algorithm. The fitness of the solution is being calculated based on the costs that choice of schedule produces, shown in following expression:

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

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 expression:

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

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, the 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:

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

After all onlookers 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 an information of how many times the bee tried to improve it but failed in a simple counter. If this counted exceeds the predefined parameter TRIALS, the bee who is exploring it becomes a scout bee and finds a new random solution. Due to the performance reasons, solution that exceeds the number of trials is only being abandoned if it is not the best solution found so far. This modification is applied 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 detailed procedure for a flight scheduling problem is given below:

```
• Set the parameters NUMBER_OF_CYCLES, COLONY_SIZE and TRIALS
• Set random COLONY_SIZE/2 schedules
• FOR (each schedule)
    o Calculate fitness
    o Calculate selection probability
• END FOR
• Memorize best schedule (with best fitness)
• REPEAT
    o FOR (each schedule)
        ▪ Find neighbor schedule
        ▪ Calculate neighbor's fitness
        ▪ Apply greedy algorithm to choose between current and new schedule
        ▪ Calculate better solution's fitness and selection probability
        ▪ Update trial number
    o END FOR
    o REPEAT
        ▪ Use selection probability to choose some of existing solutions
        ▪ Find neighbor schedule
        ▪ Calculate neighbor's fitness
        ▪ Apply greedy algorithm to choose between current and new schedule
        ▪ Calculate better solution's fitness and selection probability
        ▪ Update trial number
    o UNTIL (reached COLONY_SIZE/2 iterations)
    o Memorize best solution so far
    o FOR (each schedule)
        ▪ IF (trial > TRIALS and this solution is not the best one)
            • Find another random solution and forget the current one
            • Calculate fitness and selection probability
    o END FOR
    o Memorize best solution so far
• UNTIL (NUMBER_OF_CYCLES iterations reached)
```

III. Experiments and Discussion

A. *Settings*

The implementation of ABC and experiments are done in environment of MASDIMA, with provided data that contains information about flights and aircrafts from period of one month; scheduling is in practice mainly done within this period of time.

MASDIMA is a disruption management system based on Multi-Agent System paradigm (MAS).

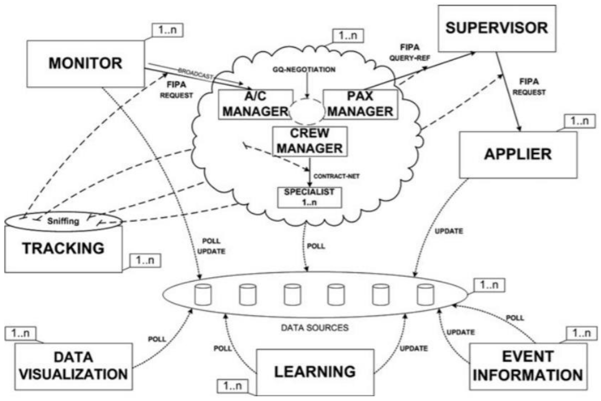

Figure 1 MASDIMA architecture

Agents in the system are separated in two main groups by their functionality: manager agents and specialist agents. Each dimension has its own manager agent whose task is negotiation with other dimensions manager agents and the supervisor over a several rounds to find an integrated solution - Passenger Manager, Crew Manager and Aircraft Manager. Selection of the best candidate solutions is done by choosing those with lowest Total Operational Cost. Specialist agents are also related to each dimension - Passenger Specialist, Crew Specialist and Aircraft Specialist. Each of them has a specific expertise are is run by different resolution algorithms. The algorithms are delivering the best candidate solutions from their own perspective, i.e. using costs regarding only their dimension into account while calculating Total Operational Cost.

The algorithm is tested by changing each of the predefined parameters to see their impact on the performance.

B. *Influence of the number of cycles*

This parameter says how long we will let algorithm to 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 algorithm while changing the total number of cycles, and remaining 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 much change in the solution improvement.

C. *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 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 notice 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.

D. *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 end result. The experiment has run in 400 cycles and with a population of 20 bees. The best performance is notices on medium vales which “give space” for solution to be improved, but still restrict the randomness.

IV. Conclusions

ABC algorithm has been modified for application on the flight scheduling problem and its performance has been analyzed. In conclusion, ABC algorithm can be successfully used for solving scheduling optimization problems. Further modifications are welcome on the implemented solution in order 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.

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 combinatorial optimization transportation engineering problems by swarm intelligence.

[5] Karaboga, D., Basturk, B. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Erciyes University, Engineering Faculty, The Department of Computer Engineering.

[6] Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Springer Science+Business Media B. V.

[7] Antonio J.M. Castro and Eugenio Oliveira, “Disruption Management in Airline Operations Control - An Intelligent Agent-Based Approach”, Web Intelligence and Intelligent Agents, edited by Zeeshan Usmani, ISBN 978-953-7619-85-4, IN-TECH 2010.