

Investigation of Flight Traffic Arrangement With ICA Colonial Competition Algorithm In Air Traffic Management

Mahdi Yousefzadeh Aghdam

Islamic Azad University Mashhad Branch

Seyed Reza Kamel Tabbakh (✉ rezakamel@computer.org)

Islamic Azad University Mashhad Branch

Seyed Javad Mahdavi Chabok

Islamic Azad University Mashhad Branch

Maryam Kheyraadi

Islamic Azad University Neyshabur Branch

Research Article

Keywords: aircraft sequencing problem, ICA Colonial Competition Algorithm, Air Traffic Management

Posted Date: June 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-556131/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Air traffic flow management is one of the most challenging work systems in the world. The issue of aircraft traffic arrangement to prevent interference and flight delays is one of the most important issues in the field of air traffic flow management. In most researches in this field, incoming or outgoing flights are usually dealt with separately and attempts have been made to provide solutions using data mining methods, mathematical problem solving, etc. To solve the problem in this paper, to select the best aircraft ready for operation (landing or takeoff), we use the ICA colonial competition algorithm, which allows selecting aircraft for incoming or outgoing flights, according to various parameters. In designing the system, an attempt has been made to make the symbols more effective in flight, to give proper weight, and to optimize the selection of colonizers according to the lower cost. To evaluate the proposed method, flight data of Mashhad airport were used for testing. The results of the system test indicate better choices for landing or flying aircraft and the acceptable performance of the colonial competition algorithm compared to the latest work done to solve the flights landing and take off sequence problem as an innovative algorithm.

Introduction

The flight landing and take off sequence issue in the air traffic environment of an airport is one of the most important issues in the aviation industry. When the aircraft want to taking off and landing to the airport flight operations area, the substantial result of flights landing and take off sequence planning should be done and the main goal is minimizing delays. On the other hand, many constraints such as the weather, the type of aircraft and the operational acceptance capacity of the airport and navigation constraints can be affective on this issue.

Many efforts have been made to solve the problem of sequencing aircraft in different dimensions. Calculating the balance between the two issues of traffic control and capacity allocation to aircraft at the airport can also be considered as a goal function to reduce costs.

Gregory and Lucas [1] were modeled the Imperialist Competitive Algorithm (ICA), a new approach in solstitial algorithms computing to finds out the best optimized answers for various optimization problems. By mathematically modeling the process of socio-political evolution, it provides the algorithm which could be decide optimization mathematical issues. Practically, this algorithm takes place in the evolutionary optimization algorithms category looking like Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, Annealed Refrigeration and etc.

Like all algorithms in this category, the colonial competition algorithm constitutes the prime set of possible answers. The prime answers are distinguished as "chromosomes" in genetic algorithm, and "particles" swarm algorithm, and "country" in the colonial competition algorithm that it improves these prime answers (countries) with a specific procedure gradually and finally supplied a proper answer to the

optimization issues (desired country). The major infrastructures of this algorithm are the policy of contraction, imperialist competition and revolution.

By simulating the procedure of social, economical and political development of countries and by mathematically designing parts of this procedure, this algorithm supplied operators in regular shape as algorithms which could help to decide the mixed optimization issues. In fact, this algorithm examines the optimization issues solutions in the form of countries and tries to gradually modified these solutions during a repetitive procedures and finally achieve the optimal problem resolvent. In this study, we have tried to investigating this issue to select the best aircraft which ready for operation (landing or taking off) with the ICA colonial competition algorithm, for incoming or outgoing flights according to the capacity of runways. In system design, an attempt has been made to make the more efficient cases in flight suitable for weighting and to optimize the selection of colonizers according to the lower cost. To evaluate the proposed method, flight data of Mashhad airport were used to perform the test. The system test results indicate better choices for landing or taking off aircraft and the acceptable performance of the colonial competition algorithm compared to the latest work done to solve the flights landing and take off sequence problem as an innovative algorithm. In the continuation of the article, first, you will see an overview of the research background in relation to flights landing and take off sequence management, and in Sect. 3, we have explained how the colonial competition algorithm works. In Sect. 4, we describe our proposed method along with the cost function and data used for implementation and testing, and in Sect. 5, the conclusions of our proposed method are given.

Literature Review

Many efforts have been made to solve the problem of sequencing aircraft in different dimensions. In most researches in this field, usually incoming or outgoing flights have been dealt with separately and attempts have been made to provide solutions, by using data mining methods, mathematical problem solving, etc. In the following, we will briefly review them. In 2000 and 2001 by Beasley [8,9], in 2005 and 2008 by Hu and Wang [10,11] and finally Liu [12] in 2010 using metaheuristics called genetic algorithm, ant colony and scattered search ,They have solved this problem.

In 2005 [13] he introduced a precision analysis algorithm based on column generalizability to solve the problem of the sequence of incoming flights. In 2009 [14], a combined linear numerical programming algorithm was used for outbound flights at Dallas International Airport. In 2011 [2], an algorithm called Cellular-Automata-based Optimization (CAO) examined the ASP problem for incoming flights on a runway.

In addition, various studies in 2007 and 2008 were reviewed separately for incoming and outgoing flights using the Simulated Refrigration (SA) and Forbidden Search(TS) algorithms. In 2008 [4] he improved the ATCS algorithm by providing a way to allow scheduling on unprepared tasks. In 2009 [5], by modifying the heuristic algorithm, SA presented a new solution with the initial solution created by the ATCS algorithm. Lin, Lu, Ying [6] presented a similar work in 2011 using a repetitive greedy algorithm. In 2013,

Hancerligullari [7] modeled the ASP problem for inbound and outbound flights using multiple flight lanes, taking into account preparation times, purpose, termination and sequence-dependent separation times.

Colonial Competition Algorithm

One of the best approach in the evolutionary computing field which finds out the optimized answers to multifold optimization issues is the Imperialist Competitive Algorithm (ICA). This procedure provides the method to solve optimization mathematical issues by mathematically designing the socio-political evolution process [1]. The main infrastructures of this procedure are the policy of contraction, imperialist competition and revolution. By simulating the process of social, economical and political development of countries and by mathematically designing parts of this procedure, it could be supplied operators in regular shape as algorithms which can help to decide mixed optimization issues.

Actually, the algorithm examines solutions of the optimization issues in countries and tends to gradually modify these answers during an iterative procedure and finally achieve to the optimal resolvent of the issues. Looking like other solstitial algorithms which has random populations, that is called a "country"; Starts. The other best population elements (equivalent to elites in genetic algorithm) are detached as colonizers. The rest of the population is also considered like a colony. The colonizers, relying on their power, draw these colonies towards them in specific procedure that follows.

The full empire power rely on both its constituent section, the colonial state (as the focal nucleus) and its colonies. Mathematically, this reliance is designed by specify imperial power as sum of the power of the colonial country plus a percentage of the average power of its colonies. Together with the organization of the early empires, colonial rivalry among them begins. Any empire that fails to succeed in colonial competition and increase the power (or however prevent its impression from diminishing) could be removed from the arena of colonial competition.

The survivorship of an empire, therefore, depends on the power to conquer, and dominate, the colonies of rival them. As a result within rivalries, the power of big empires is going to increasing and weak empires could be removed. To increasing their power, empires would have to expand their colonies. [1]

In optimization, the aim is finding an optimized answer in period of problem variables. In our research, a problem variables array is created which must be optimized. The genetic algorithm of array is called a chromosome, and also we call it a country. In a next Nvar optimization issues, a country is length Nvar array. The array is defined as follows.[14]

country = [p1, p2, ..., pNvar] (1)

The variables values in a country are displayed as decimal numbers. From a historical-cultural point of view, the constituent components of a country could be considered socio-political characteristics of that country, like a culture, language, economic structure and other characteristics. Figure 2 shows how the problem optimization variables correspond to socio-political characteristics. [1]

At the beginning of the algorithm, we design the $N_{country}$ number as an initial state. To choose the Nimp from the best parts of the crowd (countries with the least amount of cost tasks) as imperialists. The rest of the N_{col} countries are colonies, each belonging to an empire. To distribute the early colonies among the imperialists, we give each number of colonies according to their power.

Figure 4 symbolically shows how the colonies are distributed among the colonizing countries. It moves x units in the line to the imperialist and is dragged to the New Colony Position. x is a random number with a uniform distribution (or any other suitable distribution). If the distance between the colonizer and colony is usually denoted by d . [1]

$$x \sim U(0, \beta * d) \quad (2)$$

β is a number greater than 1 and close to 2. A suitable option could be $\beta = 2$. The coefficient of 1% causes colonized country to approach it from different directions while moving towards the colonizing country. The outbreak of a revolution brings about sudden changes in the socio-political characteristics. In the algorithm of colonial competition, the random displacement of a colonial country is modeled the revolution to a new random position. From an algorithmic aspect, revolution saves the whole evolutionary motion from getting stuck in the local valleys of optimism, in some cases modify situation of and brings it to a better optimization limit [1].

When the colonies move toward the colonial country, some of them, might reach a better position than the imperialist (reach points in cost tasks that produce lower cost than the value function in the imperial position.) In this study, the country, the colonial state, and the colonizer have swapped places with each other, and the algorithm procedure has continued with the colonial state in a new position, and it is the new imperialist country which begins to apply the policy of simulation to colonies.

How to move colony position and colonizer is explained with Fig. 3. In the course of imperialist rivalries, unable empires, whether they like it or not, gradually fall and all colonies are felt into the hands of stronger empires. Other conditions could be considered for the falling of an empire. In the proposed algorithm, an empire is considered removed when it has lost colonies. Figure 4 provides an overview of the fall of empires in the process of the algorithm cycle [1].

3-1- Colonial competition algorithm Steps

In colonial competition algorithm, for the first step we select a few random points on the function and form the initial empire, for the second step, according to Fig. 5, we move colonies to the imperialist country. In the third step, we apply the revolution operator. If there is a colony that costs lower than the imperialist, we replace it with the imperialist, then compute the total cost, taking into account, the cost of all.

Depending on the result of the total cost obtained, we are selected one (or more) colonies from the weakest empire and give it to the empire most likely to be conquered. And we eliminate weak empires. In the last stage, if there is only one empire left, the work is stopped, otherwise it goes back to step two and

the stages are repeated. In Fig. 5, you can see the steps of the colonial competition algorithm. It is noteworthy that in this chart, the part of the revolution is not shown to be applied to its proper stage. [1]

3-2-The colonial competition usage in problem solving

In general, the colonial competition algorithm is applicable to any kind of optimization problem without any restrictions. This issue has led to the use of this algorithm in solving many problems in the field of electrical engineering, mechanics, industry, management, civil engineering, artificial intelligence, etc. For example, this algorithm has used successfully to resolute the practical issues of following optimization. Figure 6 shows the process of applying the colonial competition algorithm to an optimization problem in general. Neural network learning, which is an optimization issue, will have the same process.

The Proposed Method

At the beginning of the algorithm, we design the N-country number as the primary country, choosing N-imp as imperialist from the best parts of this population (countries with the lowest cost simulation). The rest of the N-col of the countries form colonies, each belonging to empire. To distribute early colonies among the imperialists, we give a number of colonies to each imperialist that are commensurate with its power. To do this, at the charge of all imperialists, we consider the normalizing cost as follows.

$$C_n = \max_i \{C_i\} - C_n \quad (3)$$

Where c_n is the imperial cost n-th and $\max_i \{c_i\}$ is the highest charge among them and C is the normalized cost of those. Any imperialist that costs more (the weaker imperialist) will have a lower cost of normalization. With the cost of normalization, the relative normalized power of them is computed as follows, and based on it, the colonies are distributed among the imperialists.

$$P_n = \left| \frac{C_n}{\sum_{i=1}^n (C_i)} \right| \quad (4)$$

On the other hand, the normalized power of an imperialist is the proportion of the colonies ruled by that. Therefore some colonies of an imperialist could be equal to:

$$N.C_n = \text{round}\{P_n \cdot (N_{col})\} \quad (5)$$

Where $N.C.n$ is the main member of colonies and $N-col$ is the total number of colonies in the population of the original countries. Given the $N.C.$ for each of them, we randomly select the member of early colonial countries and give them to nth imperialist. Having the prime state of all, the colonial competition algorithm begins. The evolutionary procedure is in a loop that continues till cessation condition is met. Because ICA is an evolutionary method, it must first be started by a number of countries (the initial population).

Thus, p creates a random initial population for the problem, and the values of the objective function $f(i)$, the countries with the lowest objective values for the problem, are considered as colonial countries, and by shifting the indices 1 to m of the initial population. They form. Then each of the colonizing countries is assigned an equal number of colonized countries. In addition, due to the nature of the function of the correct component, the other countries are also assigned to the strongest empire.

It should be noted that the assignment of each colony to the colonizing country is random and equal in probability. Naturally, the colonized countries should move culturally and socially towards the colonizing countries using the absorption function. Therefore, in this algorithm, the nearest random neighbor method (two-point enhancer) is used for the absorption function.

Data Used And Their Revenue

In order to evaluate and present the results, it is necessary to collect a database. For this purpose, the real and available information of the airports about the flights has been used. The sample of this database is shown in Fig. 7.

It is also possible to use hypothetical design to create hypothetical traffic and create a specific database, then test on it.

5-1- The objective function definition and calculating the cost

For this problem, it is necessary to obtain the objective function on the efficiency of airport operations and accuracy in servicing flights.

M Number of aircraft in traffic pattern jet landing

N Number of aircraft in apron for flight

R used band number

h_i Delay rater

R_k Band number used

v_i Fuel and headling costs in case of flight priority delay

We basically consider three objective functions:

$$A = \sum_{i=1}^M (h_i + v_i) \quad (6)$$

$$D = \sum_{i=1}^N (h_i + v_i) \quad (7)$$

$$W = R_k \text{ xor } \min(A, D) \quad (8)$$

A is the sum of costs for delays and costs for priority in priority for the incoming flight, and D is the sum of costs for delays and costs for priority in priority for outbound flights, and W is the minimum of delays and costs for each The runway is for inbound and outbound flights. It is clear that inbound and outbound flights can be specified for different runways, as well as the arrangement of aircraft for each runway.

We have considered the following specifications for each aircraft:

Table 1
flight specifications

Type of flight control system	Flight altitude	Aircraft speed	Atmospheric condition and vision	Flight rules	Aircraft type
-------------------------------	-----------------	----------------	----------------------------------	--------------	---------------

In order of input, for each characteristic, a special weighting is done according to the rules of air. We need to be able to manipulate these random initial weights. From the long Country string, we separate the weights of the different layers in the coding order we have chosen.

We place these weights in their proper places in the network and inject the training input data into the network and take its output and compare it with the actual test output and return it .Now we have to train the network. Without manipulating the original code, we replaced the train command used in it with our new function.

We have put the ready-made codes of the colonial competition algorithm (or any other evolutionary algorithm) inside this function. We have considered the length of the country vector string (dimension or the same number of optimization variables) equal to the total number of unknown weights of the network. Having a defined cost function, like any other optimization problem, we look at this optimization problem and apply the code to it. The results of the test for 100 aircraft, taking into account 5 runways and other ICA settings, are described in Fig. 9.

Conclusion

The colonial competition algorithm, while being known and applied to many applications, has not yet been used for many optimization problems. One of the advantages of this colonial competition algorithm is the novelty of the basic idea, at the first optimization algorithm based on a social and political procedure, the ability to making optimize evenly and higher collate to different optimization algorithms, in comparison with different types. He pointed out the problems of optimization and finding the right speed for the optimal answer.

The experimental results of the colonial algorithm on different cost functions show that the introduced algorithm is quite successful in finding the optimal point of these functions. Also, various application problems solved with this algorithm show that the proposed optimization strategy can successfully solve practical and engineering problems with complete success along with other proposed optimization

methods such as genetic algorithm and particle group. Comparison of the results obtained by the proposed algorithm with conventional optimization methods also indicates the relative superiority of this algorithm. In our case, ie air traffic arrangement, considering that the ICA algorithm examines more situations and the regional fragmentation is more due to countries, and the type of flight traffic may occur in many cases and in different regions.

This algorithm is an acceptable choice. It can be adjusted more dynamically, accurately and flexibly with real-world parameters. In terms of typology, the application of the algorithm for simultaneous outgoing and incoming flights, as well as appropriate weighting of aircraft characteristics as the entry points of the algorithm has been proposed. For future work, it is proposed to use a combination of colonial competition algorithm or modified colonial competition algorithms with fuzzy technique for air traffic control laws.

Declarations

Not applicable.

Ethics approval and consent to participate

Not applicable.

Availability of data and materials

www.kaggle.com

Competing interests

The authors declare that they have no competing interests.

Author information

Department of Computer Engineering ,Neyshobur Branch, Islamic Azad University, Neyshabur IRAN

Mahdi.yosefzadeh@gmail.com

Seyed Reza Kamel Tabbakh

Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, IRAN

DrKamel@mshdiau.ac.ir

Seyed Javad Mahdavi Chabo

Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, IRAN

Mahdavi@mshdiau.ac.ir

Maryam kheyrabadi

Department of Computer Engineering ,Neyshabur Branch, Islamic Azad University, Neyshabur IRAN

Maryam.abadi@gmail.com

Funding

Not applicable.

Authors' Contributions

All the corresponding authors contributed equally to the conduct of the present study

Acknowledgements

Not applicable

References

- [1] Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661-4667). Ieee.
- [2] Yu, S. P., Cao, X. B., & Zhang, J. (2011). A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation. *Applied Soft Computing*, 11(4), 3485-3493.
- [3] Atkin, J. A., Burke, E. K., Greenwood, J. S., & Reeson, D. (2007). Hybrid metaheuristics to aid runway scheduling at London Heathrow airport. *Transportation Science*, 41(1), 90-106.
- [4] Atkin, J. A., Burke, E. K., Greenwood, J. S., & Reeson, D. (2008). On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport. *Journal of Scheduling*, 11(5), 323-346.
- [5] Chen, J. F. (2009). Scheduling on unrelated parallel machines with sequence-and machine-dependent setup times and due-date constraints. *The International Journal of Advanced Manufacturing Technology*, 44(11-12), 1204-1212.
- [6] Lin, S. W., Lu, C. C., & Ying, K. C. (2011). Minimization of total tardiness on unrelated parallel machines with sequence-and machine-dependent setup times under due date constraints. *The International Journal of Advanced Manufacturing Technology*, 53(1-4), 353-361.
- [7] Hancerliogullari, G., Rabadi, G., Al-Salem, A. H., & Kharbeche, M. (2013). Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. *Journal of Air Transport Management*, 32, 39-48.

- [8] Hancerliogullari, G., Rabadi, G., Al-Salem, A. H., & Kharbeche, M. (2013). Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. *Journal of Air Transport Management*, 32, 39-48.
- [9] Beasley, J. E., Sonander, J., & Havelock, P. (2001). Scheduling aircraft landings at London Heathrow using a population heuristic. *Journal of the operational Research Society*, 52(5), 483-493.
- [10] Hu, X. B., & Chen, W. H. (2005). Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence*, 18(5), 633-642.
- [11] Hu, X. B., & Di Paolo, E. (2008). Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling. *IEEE Transactions on Intelligent Transportation Systems*, 9(2), 301-310.
- [12] Liu, Y. H. (2011). A genetic local search algorithm with a threshold accepting mechanism for solving the runway dependent aircraft landing problem. *Optimization Letters*, 5(2), 229-245.
- [13] Wen, M. (2005). Algorithms of scheduling aircraft landing problem (Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark).
- [14] Gupta, G., Malik, W., & Jung, Y. (2009, September). A mixed integer linear program for airport departure scheduling. In 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS) (p. 6933).

Figures



Figure 1

The movement of a colonial country towards the colonizer [1]

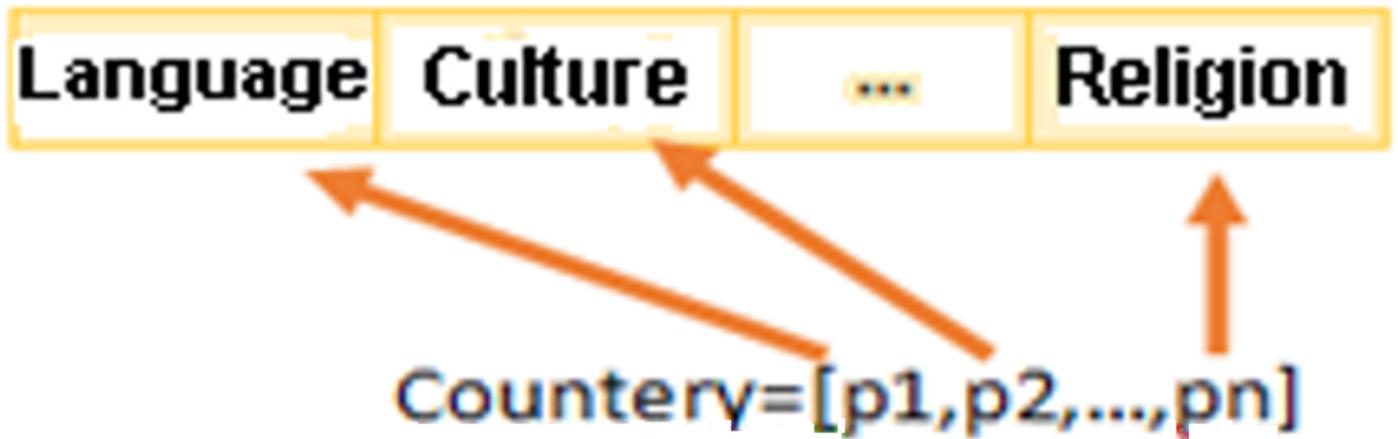


Figure 2

Correlation of problem optimization variables with socio-political characteristics

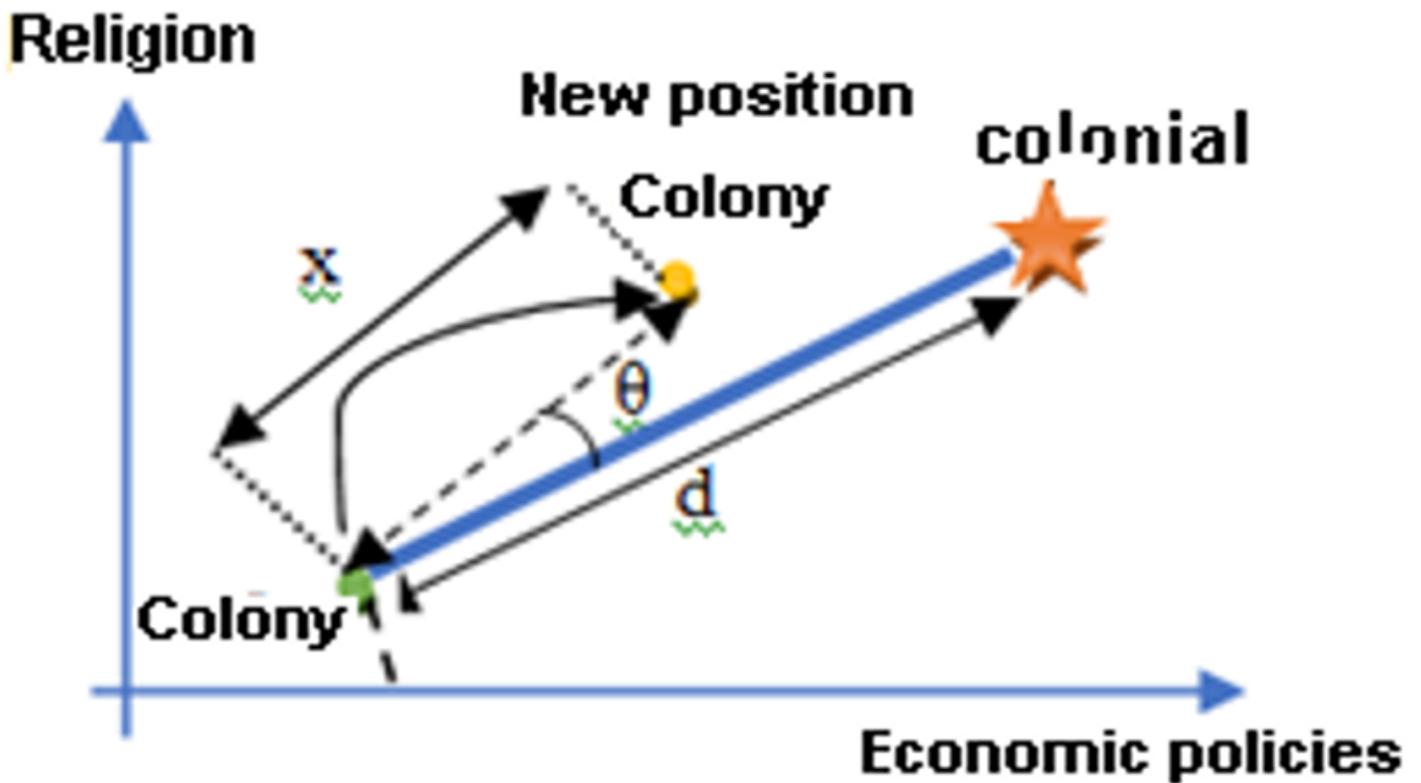


Figure 3

Displacement of the colonial and imperialist position [14]

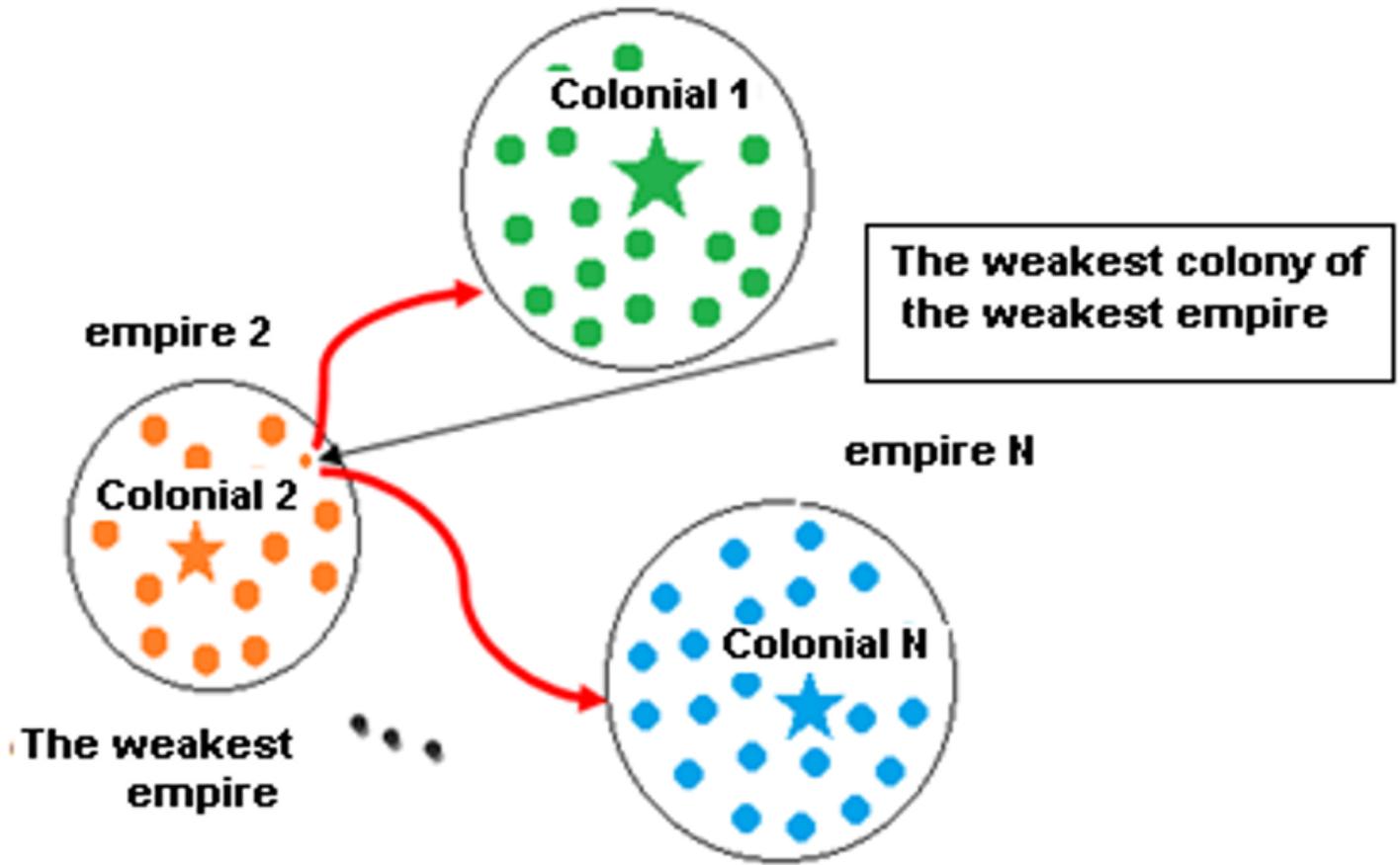


Figure 4

Colonial rivalry between several colonizers

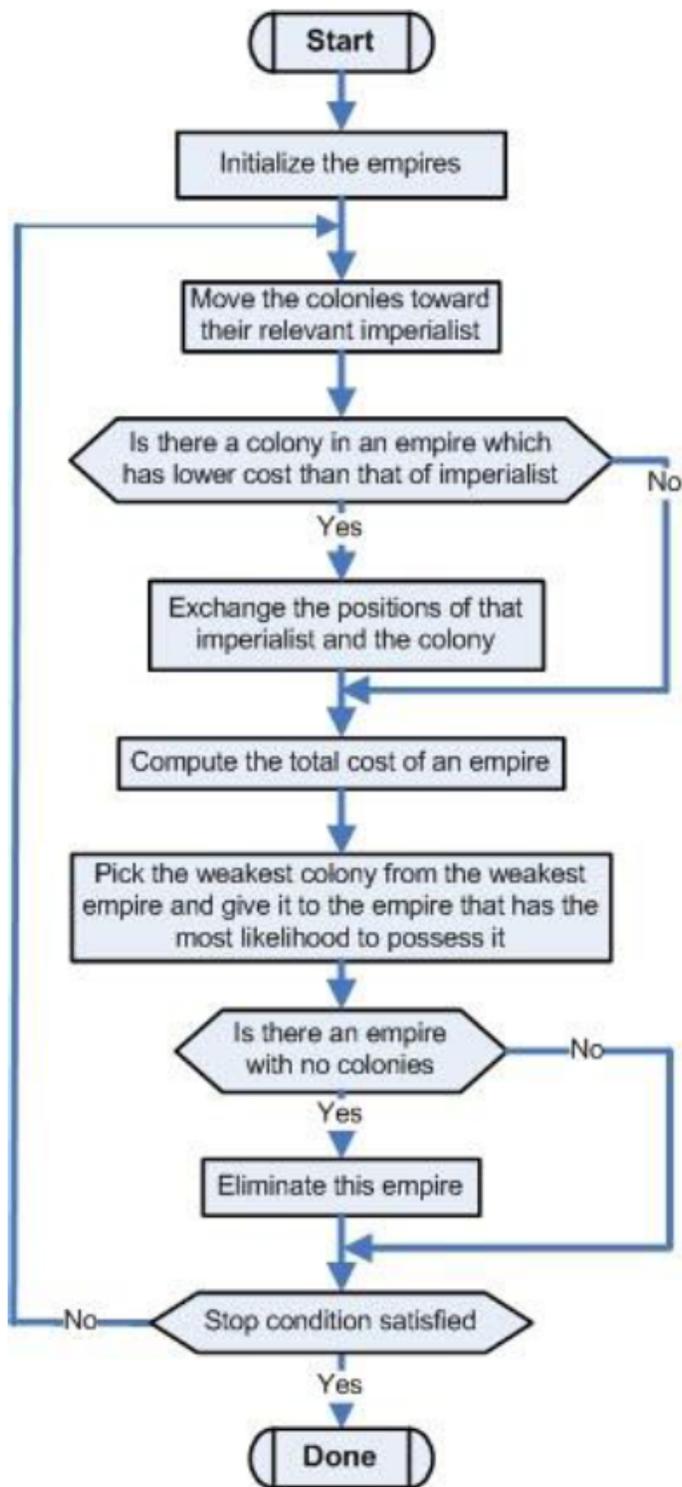


Figure 5

Flowchart of Colonial Competition Algorithm



Figure 6

The process of applying the colonial competition algorithm to an optimization problem in general

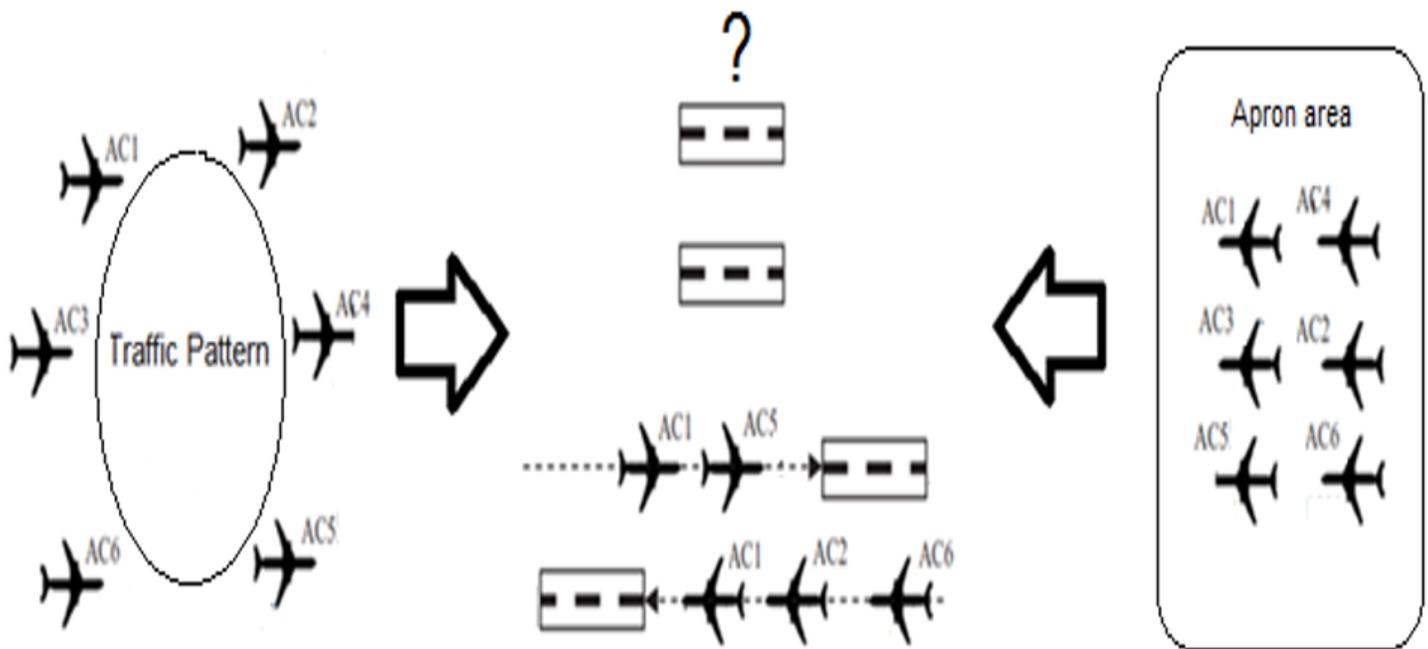


Figure 7

Problem definition range at the airport

Image not available with this version

Figure 8

Figure8

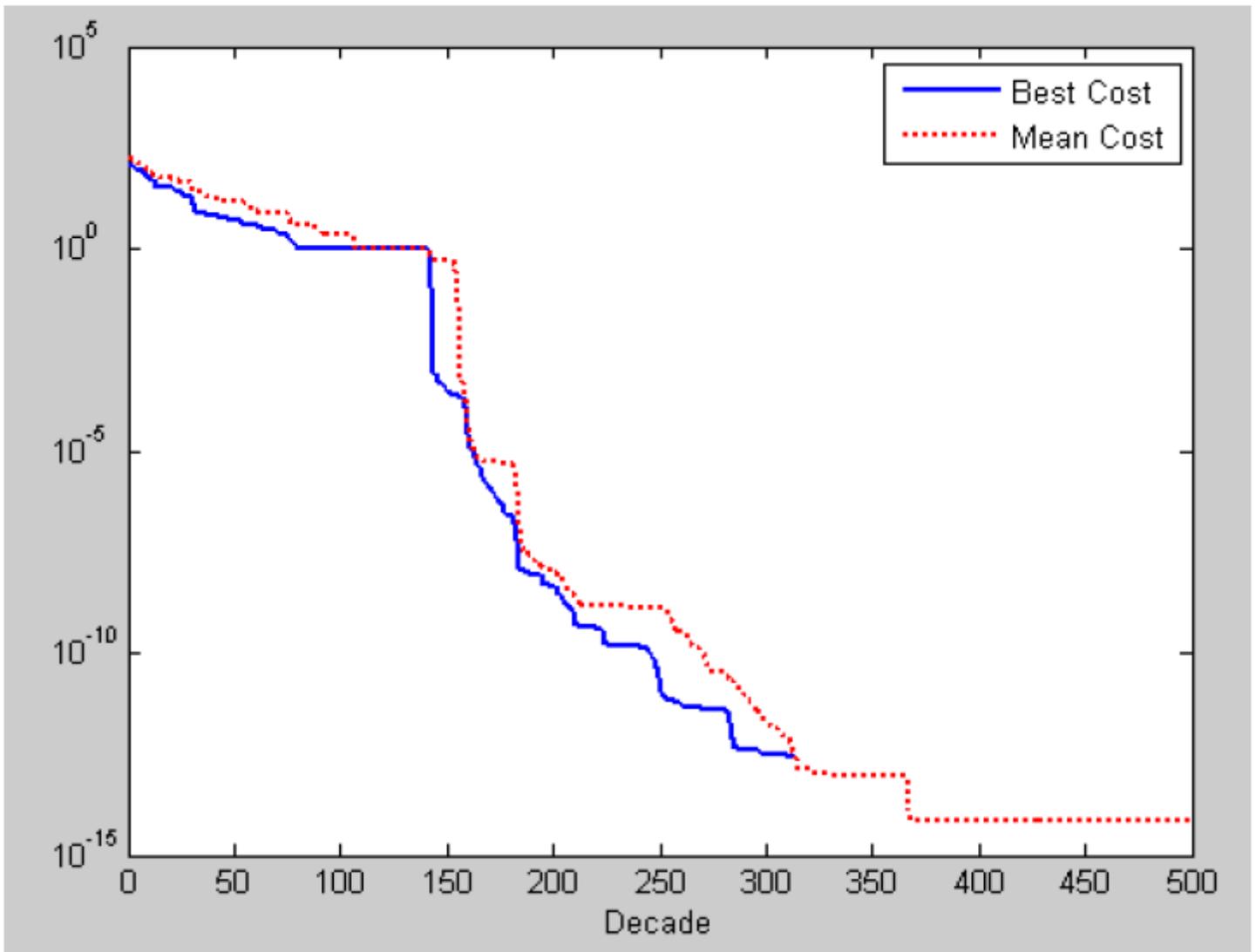


Figure 9

The best cost chart based on the ICA implementation is designed based on the scenario