

# Ant Metaheuristic Algorithm for the Frequency Assignment Problem

KHALED ALRAJHI

King Khaled Military Academy, Riyadh, SAUDI ARABIA

**Abstract:** This study proposes the ant metaheuristic algorithm to solve the frequency assignment problem. In this approach, the descent method constructed as three ants which each of which applying a metaheuristics algorithm. Several novel and existing techniques are used to improve the performance of this algorithm. One of these techniques is applying the concept of a well-known graph colouring algorithm, namely recursive largest first. Furthermore, this study compares this algorithm using two visibility definitions. The first definition is based on the number of feasible frequencies and the second one is based on the degree. Additionally, we compare this algorithm using two trail definitions. The first one is between requests and frequencies. The second is between requests and requests. This study considers real and randomly generated benchmark datasets of the static problem and our algorithm achieved competitive results comparing with other ant colony optimization algorithms in the literature.

**Keywords:** ant colony optimization, graph colouring algorithm, frequency assignment problem.

Received: July 7, 2022. Revised: September 5, 2023. Accepted: October 6, 2023. Published: November 6, 2023.

## 1 Introduction

The frequency assignment problem (FAP) is related to wireless communication networks, which are used in many applications such as mobile phones, TV broadcasting and Wi-Fi. The aim of the FAP is to assign frequencies to wireless communication connections (also known as requests) while satisfying a set of constraints, which are usually related to prevention of a loss of signal quality. Note that the FAP is not a single problem. Rather, there are variants of the FAP that are encountered in practice. The minimum order FAP (MO-FAP) is the first variant of the FAP that was discussed in the literature, and was brought to the attention of researchers by [1]. In the MO-FAP, the aim is to assign frequencies to requests in such a way that no interference occurs, and the number of used frequencies is minimized. As the MO-FAP is NP-complete [2], it is usually solved by meta-heuristics.

Many meta-heuristics have been proposed to solve the MO-FAP including genetic algorithm (GA) [3], evolutionary search (ES) [4], ant colony optimization (ACO) [5], simulated annealing (SA) [6] and tabu search (TS) [6, 7, 8, 9]. It can be seen from literature that there are relatively few papers concerning the application of ACO to solve the FAP. However, existing ACO algorithms in the literature are unable to find a feasible solution in some instances of the MO-FAP. Hence, this study investigates whether ACO can be improved to be an effective solution method for the MO-FAP.

In this study, Ant Metaheuristics Algorithm (AMA) is mainly designed to solve MO-FAP. Several novel and existing techniques are used in this study to improve the performance of AMA. One of these techniques is applying the concept of a well-known graph colouring algorithm, namely Recursive Largest First (RLF) [15]. Furthermore, this study compares AMA using two visibility definitions (see Section 3.3). The first definition is based

on the number of feasible frequencies [10]. The second one is based on the degree [12]. Additionally, we compare AMA using two trail definitions (see Section 4.2.4). The first one is between requests and frequencies [13]. The second trail definition considered in this study is between requests and requests [12].

This paper is organised as follows: the next section gives an overview of the MO-FAP. Section 3 presents the main components of our AMA algorithm for the MO-FAP. Results of this algorithm are given and discussed in Section 4 before this study finishes with conclusions.

## 2 Overview of the MO-FAP

The main concept of the MO-FAP is assigning a frequency to each request while satisfying a set of constraints and minimizing the number of used frequencies. The MO-FAP can be defined formally as follows: given

- a set of requests  $R = \{r_1, r_2, \dots, r_{NR}\}$ , where  $NR$  is the number of requests,
- a set of frequencies  $F = \{f_1, f_2, \dots, f_{NF}\} \subset \mathbb{Z}^+$ , where  $NF$  is the number of frequencies,
- a set of constraints related to the requests and frequencies (described below),

the goal is to assign one frequency to each request so that the given set of constraints are satisfied and the objective function is minimized, where the objective function is minimizing the number of used frequencies. Note that the frequency that is assigned to requests  $r_i$  is denoted as  $f_{r_i}$  throughout of this study. The MO-FAP has four variants of constraints as follows:

1. **Bidirectional Constraints:** this type of constraint forms a link between each pair of requests  $\{r_{2i-1}, r_{2i}\}$ , where  $i = 1, \dots, NR/2$ . In these constraints, the frequencies  $f_{r_{2i-1}}$  and  $f_{r_{2i}}$  that are assigned to  $r_{2i-1}$

and  $r_{2i}$ , respectively, should be distance  $d_{r_i r_j}$  apart. In the datasets considered here,  $d_{r_i r_j}$  is always equal to a constant value (238). These constraints can be written as follows:

$$\begin{cases} |f_{r_i} - f_{r_j}| \\ = d_{r_i r_j} \end{cases} \quad \text{for } i = 1, \dots, NR/2 \quad (1)$$

- Interference Constraints:** this type of constraint forms a link between a pair of requests  $\{r_i, r_j\}$ , where the pair of frequencies  $f_{r_i}$  and  $f_{r_j}$  that is assigned to the pair of requests  $r_i$  and  $r_j$ , respectively, should be more than distance  $d_{r_i r_j}$  apart. These constraints can be written as follows:

$$\begin{cases} |f_{r_i} - f_{r_j}| \\ > d_{r_i r_j} \end{cases} \quad \text{for } 1 \leq i < j \leq NR \quad (2)$$

- Domain Constraints:** the available frequencies for each request  $r_i$  are denoted by the domain  $D_{r_i} \subset F$ , where  $\cup_{r_i \in R} D_{r_i} = F$ . Hence, the frequency which is assigned to  $r_i$  must belong to  $D_{r_i}$ . For the datasets considered in this study, there are 7 available domains.
- Pre-assignment Constraints:** for certain requests, the frequencies have already been pre-assigned to given values i.e.  $f_{r_i} = p_{r_i}$ , where  $p_{r_i}$  is given value.

### 3 Components of the Ant Metaheuristics Algorithm

The components of AMA include solution space and cost function, visibility definitions, trail definitions and descent method.

#### 3.1 Solution Space and Cost Function

The solution space of AMA is defined as the set of all possible feasible assignments, that is, satisfying all of the constraints. The corresponding cost function is defined as the number of unassigned requests.

#### 3.2 Request and Frequency Selection

AMA selects a frequency  $f_j$  greedily by selecting the one which can be assigned feasibly to the most requests. If there is more than one candidate frequency, then one of them is randomly selected. After that, the frequency  $f_j$  is sequentially feasibly assigned to all possible requests until no more can be feasibly assigned. The order of selecting requests from among those that are feasible for  $f_j$  is based on probability  $p_{r_i f_j}$  given by Formula 3.

$$G_r \quad p_{r_i f_j} = \begin{cases} \frac{\tau_{r_i f_j}^\alpha \cdot \eta_{r_i f_j}^\beta}{\sum_{f_k \in G} \tau_{r_i f_k}^\alpha \cdot \eta_{r_i f_k}^\beta} & \text{if } f_j \in G_r \\ 0 & \text{otherwise} \end{cases}$$

where  $G_{r_i}$  is the set of frequencies which can be feasibly assigned by an artificial ant to the request  $r_i$ . The visibility  $\eta_{r_i f_j}$  of a request  $r_i$  to be assigned a frequency  $f_j$  is defined in Section 3.3 and the trail  $\tau_{r_i f_j}$  is defined in Section 3.4. The parameters  $\alpha, \beta \geq 0$  control the relative significance of the pheromone trail  $\tau_{r_i f_j}$  against the visibility  $\eta_{r_i f_j}$ .

After that, a different frequency is selected in the same way and this process is repeated until all requests are feasibly assigned, if possible. This process is inherited from a well-known graph colouring algorithm, namely recursive largest first. In fact, applying recursive largest first aims to improve the performance of selecting frequencies and requests to be assigned. In contrast, ACO for the MO-FAP in the literature (see e.g. [13]) frequently selects a request based on probability and then assign it to a feasible frequency.

#### 3.3 Visibility Definitions

The visibility gives some indication of the desirability of choosing a request based on the experience of previous ants. Hence, the visibility of a request acts as a greedy heuristic. In this study, two types of visibility definition are applied and compared. These two visibilities are defined as follows:

- Visibility  $\eta_{r_i f_j}$  of a request  $r_i$  to be assigned a frequency  $f_j$  is based on the number of feasible frequencies for  $r_i$  ( $NFF_{r_i}$ ), which is given by Formula 4.

$$\eta_{r_i f_j} = \frac{1}{NFF_{r_i}} \quad (4)$$

This definition prioritises those requests that have fewer feasible frequencies. This type of visibility definition was previously used in ACO for the graph colouring problem (GCP) [10].

- Visibility  $\eta_{r_i f_j}$  of a request  $r_i$  to be assigned a frequency  $f_j$  is based on the degree of  $r_i$  ( $DEG_{r_i}$ ), which is defined as the numbers of unassigned requests that cannot be assigned feasibly to  $f_j$  and have a common interference constraint with  $r_i$ . This visibility is given by Formula 5.

$$\eta_{r_i f_j} = DEG_{r_i} + 1 \quad (5)$$

This visibility looks ahead and prioritises requests that have more constraints in common with other requests

that cannot be assigned to the frequencies being considered currently. This visibility definition was previously used in ACO for the GCP [12].

Example 1 clarifies the probability of selecting a request based on the two different visibility definitions.

### 3.4 Trail Definitions

The purpose of the trail within AMA is to provide information about previous construction solutions to influence future constructions. In this study, two different trails are defined, where the initial values of these trails are set to one. Moreover, evaporation and updating of these trails are discussed. The definitions of these trails are given as follows:

i) Trail between requests and frequencies ( $T_A RF$ ): the key component of a solution is to decide to which frequency each request is assigned. Therefore, the most obvious trail definition is between each request and each frequency [14]. The value of the trail indicates the quality of previous solutions when a request is assigned to a frequency.

ii) Trail between requests and requests ( $T_A RR$ ): previous work on the graph colouring problem (GCP) in [12] found that a trail between nodes and nodes was more successful than a trail between nodes and colours. This is because the important aspect of a graph colouring solution is not in which colour each node is placed, as the colours are interchangeable. The important aspect is which nodes are placed together in the same colour class. When considering the static FAP, clearly the actual frequency to which each request is assigned is important. However, given the static FAP has the same underlying model as the GCP, we decided to investigate whether a trail based on which requests are assigned to the same frequencies could be advantageous.

This trail measures the success of previous solutions when requests are assigned to the same frequency using *Average*  $T_A RR$ , which is the average trail between the prospective request and all requests already assigned to the candidate frequency  $f_j$ , which is defined by Formula 6.

$$Average T_A RR\{r_i\} = \sum_{r_j \in H, i \neq j} \frac{T_A RR\{r_i, r_j\}}{|H|} \quad (6)$$

where  $H$  is the set of requests already assigned to frequencies  $f_j$ .

#### 3.4.1 Trail Evaporation

Both types of trail are evaporated after each generation by multiplying the trail by the evaporation parameter, which will be determined experimentally. The trail evaporation can be defined by Formula 7.

$$\tau_{r_i f_j} \leftarrow \rho \cdot \tau_{r_i f_j} \quad (7)$$

where the evaporation parameter  $\rho$  is in the range  $[0, 1)$ .

#### 3.4.2 Trail Updates

The trails are updated using two reward functions, namely  $Cost_1$  and  $Cost_2$ , which are defined as follows:

- $Cost_1$ : counts the number of used frequencies in the current solution. This is appropriate when a solution is feasible.
- $Cost_2$ : counts the number of unassigned requests in the current solution. This is appropriate when a solution is infeasible.

The values of  $T_A RF$  could have been updated using Formula 8.

$$T_A RF\{r_i, f_j\} = T_A RF\{r_i, f_j\} + \frac{10}{Cost_1 + Cost_2 - Best + 1} \quad (8)$$

where  $Best$  is the best minimum number of used frequencies found so far in the search. Note that  $Cost_1 + Cost_2 - Best$  can be equal to 0 when  $Cost_1 = Best$  and  $Cost_2 = 0$ . Thus, we add 1 to the denominator of the last term in Formula 8. A similar trail update function was previously used in ACO for the GCP [12].

Similarly, the values of  $T_A RR$  are updated using Formula 9.

$$T_A RR\{r_i, r_j\} = T_A RR\{r_i, r_j\} + \frac{10}{Cost_1 + Cost_2 - Best + 1} \quad (9)$$

Another problem of trail updates is that only requests that have been assigned to frequencies are updated. Therefore, the trail values on any unassigned requests are not increased, meaning such requests are likely to be selected even later in the following construction processes. As we would prefer to consider them earlier in the construction process, the trail is increased on each unassigned request for all available frequencies. This idea was previously used in ACO for the examination scheduling problem [11].

### 3.5 Descent Method

This method is executed only when no feasible solution can be found by all ants in a generation. In such generations, the descent method is executed only for one ant which constructs the infeasible solution with the minimum number of unassigned requests. First, these requests are assigned to the frequencies which lead to the least number of violations. Then, the descent method aims to reduce the number of violations with a fixed number of frequencies to find a feasible solution, if possible, by going three ants:

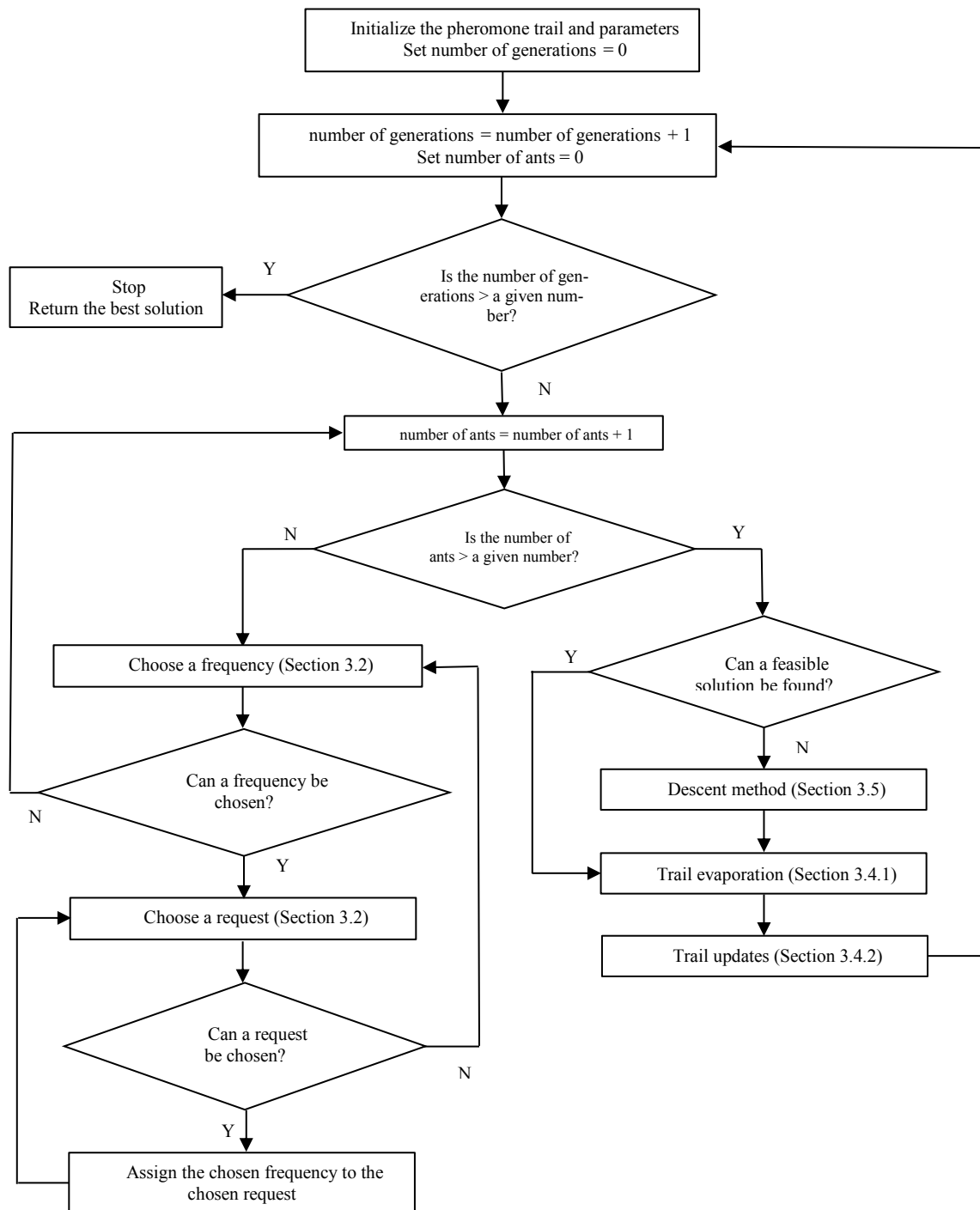
- Ant 1: applying local search [6]
- Ant 2: applying evolutionary search [4]
- Ant 3: applying tabu search [8]

### 3.6 The AMA Algorithm Implementation

AMA consists of a given number of generations, each of which contains a given number of ants, where each ant individually constructs a solution. Each ant starts con-

structuring a solution by selecting a frequency to be assigned to all possible feasible requests. The process is repeated until no frequencies can be selected (see Section 3.2). After all ants in the current generation construct their solutions, if no feasible solution can be found, then the descent method (see Section 3.5) is used to attempt

to achieve a feasible solution. Then, the trail is evaporated and updated (see Section 3.4.1 and 3.4.2). After that, the next generation is executed by the same process. The overall structure of the AMA algorithm is illustrated in Figure 1.



**Fig. 1.** Overall structure of our ACO algorithm for the MO-FAP.

## 4 Experiments and Results

This section presents and compared the performance of AMA in three sections. The first section gives the results of AMA for the static FAP. The second section compares the performance of AMA with existing ACO algorithms in the literature. Finally, the performance of AMA is compared with other algorithms in the literature.

AMA is implemented in FORTRAN 95 and all experiments were conducted on a 3.0 GHz Intel Core I3-2120 Processor (2nd Generation) with 8GB RAM and a 1TB Hard Drive.

### 4.1 Results Comparison of the AMA

In this study, the number of generation of AMA is 200, where this number is selected based on experiments. Moreover, the performance of ACO is compared based on several options of the following components:

1. The number of ants,
2. The trail definition,
3. The visibility definition,
4. The parameters  $\alpha$ ,  $\beta$  and  $\rho$

Different values of the number of ants, two options of the trail definition and two options for visibility definition are compared. For the parameters  $\alpha$ ,  $\beta$  and  $\rho$ , three values of each parameter are tested. By considering all these options, there are 756 versions of AMA to be compared. Moreover, each version is tested on 10 instances with 5 runs being performed on each instance. Therefore, considering all the versions of AMA take excessive time. Hence, the comparison is made for each component while fixing the others; i.e. first, different numbers of ants are compared while fixing the remaining components. After selecting the best number of ants, the two different trail definitions are compared. After that, two definitions of the visibility are compared and finally, different values of the parameters ( $\alpha$ ,  $\beta$  and  $\rho$ ) are compared in the same way. Based on experiments, the best values of the parameters and number of ants given in Table 3.

**Table 3.** The best values of the parameters and number of ants based on experiments.

$\alpha$	$\beta$	$\rho$	number of ants
4	3	0.85	30

Moreover, the performance of AMA using  $T_A RF$  is better than using  $T_A RR$ . The performance of AMA using the two types of trail definitions. It is found by the Wilcoxon signed-rank test at the 0.05 significance level that there is a significant difference between the performances of ACO using  $T_A RF$  and  $T_A RR$ . Moreover, the performance of AMA using the first definition of visibility better than the second one.

### 4.2 Results Comparison with Existing ACO Algorithms

The performance of our AMA is compared with existing ACO in the literature. To the best of my knowledge, only one published research [13] applied ACO for the MO-FAP using CELAR and GRAPH datasets. Table 4 shows the results in the form given in [13], i.e. in the form of (y) where y is the number of violations. Note that y is equal to 0 means a feasible solution is found.

**Table 4.** Results of AMA and existing ACO algorithm in the literature.

Instance	ACO [13]	Our AMA
CELAR 01	(0)	(0)
CELAR 02	(0)	(0)
CELAR 03	(0)	(0)
CELAR 04	(8)	(0)
CELAR 11	(2)	(2)
GRAPH 01	(0)	(0)
GRAPH 02	(0)	(0)
GRAPH 08	(0)	(0)
GRAPH 09	(0)	(0)
GRAPH 14	(0)	(0)

Table 4 shows that both of the algorithms struggled to find a feasible solution for CELAR 11. Moreover, ACO in [13] could not achieve a feasible solution for CELAR 04, whereas our AMA could. Overall, our AMA algorithms performing better than ACO in [13].

## 5 Conclusions

In this study, an improved AMA was introduced to solve the MO-FAP by using different techniques. One of the techniques which was applied to improve the performance of AMA is improve the descent method by applying local search [6], evolutionary search [4] and tabu search [8]. This will be applied only when no feasible solution can be found in a generation. In such generations, the descent method is executed for only one ant which constructs the infeasible solution with the minimum number of unassigned requests. Moreover, applying the recursive largest first. Also, AMA was compared using two trail definitions and two visibility definitions. It was found that using the trail between requests and frequencies led to better performance than the other trail definition. Moreover, using the visibility definition based on the number of feasible frequencies resulted in better performance than another visibility definition. Furthermore, several values for the parameters  $\alpha$ ,  $\beta$ ,  $\rho$  were compared and the best values are 4, 3 and 0.85, respectively. Overall, our AMA performed better than ACO in the literature.

## References

- [1]. Metzger, B. (1970), Spectrum Management Tech-nique, Presentation at 38th National ORSA meeting (Detroit, MI).
- [2]. Garey, M. and Johnson, D. (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman W.H. and Company, San Francisco, California.
- [3]. Kapsalis, A., Chardaire, P., Rayward-Smith, V. and Smith, G. (1995), The Radio Link Frequency As-ignment Problem: A Case Study Using Genetic Al-gorithms, Lecture Notes on Computer Science, pp. 117-131.
- [4]. Crisan, C. and Mühlenbein, H. (1998), The Frequen-cy Assignment Problem: A Look at the Perfor-mance of Evolutionary Search, Lecture Notes in Computer Science, Vol. 1363, pp. 263-274.
- [5]. Parsapoor, M. and Bilstrup, U. (2013), Ant Colony Optimization for Channel Assignment Problem in a Clustered Mobile Ad Hoc Network, International Conference on Swarm Intelligence (ICSI), Vol. 1, pp. 314-322.
- [6]. Tiourine, S., Hurkens, C. and Lenstra, J. K. (1999), Local Search Algorithm for the Radio Link Fre-quency Assignment Problem, Telecommunication System, Vol. 13, pp. 293-314.
- [7]. Bouju, A., Boyce, J., Dimitropoulos, C., Vom Scheidt, G. and Taylor, J. (1995), Tabu Search for the Radio Links Frequency Assignment Problem, In Applied Decision Technologies, London, [ADT'95]. UNICOM Conference.
- [8]. Hao, J., Dorne, R. and Galinier, P. (1998), Tabu Search for Frequency Assignment in Mobile Radio Networks, Journal of Heuristics Vol. 4, pp. 47-62.
- [9]. Bouju, A., Boyce, J., Dimitropoulos, C., Vom Scheidt, G. and Taylor, J. (1995), Intelligent Search for the Radio Links Frequency Assignment Prob-lem, Proceedings of the International Conference on Digital Signal Processing.
- [10]. Costa, D. and Hertz, A., 1997. Ants can colour graphs. Journal of the Operational Research Socie-ty, 48(3), pp.295-305.
- [11]. Dowsland, K.A. and Thompson, J.M., 2005. Ant col-ony optimization for the examination scheduling problem. Journal of the Operational Research Soci-ety, pp.426-438.
- [12]. Dowsland, K.A. and Thompson, J.M., 2008. An im-proved ant colony optimisation heuristic for graph colouring. Discrete Applied Mathematics, 156(3), pp.313-324.
- [13]. Maniezzo, V. and Carbonaro, A., 2000. An ANTS heuristic for the frequency assignment problem. Fu-ture Generation Computer Systems, 16(8), pp.927-935.
- [14]. Parsapoor, M. and Bilstrup, U., 2013. Ant colony op-timization for channel assignment problem in a clus-tered mobile ad hoc network. In Advances in Swarm Intelligence, pp. 314-322. Springer Berlin Heidelberg.
- [15]. Leighton, F.T., 1979. A graph coloring algorithm for large scheduling problems. Journal of Research of the National Bureau of Standards, 84(6), pp.489-506.

### **Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

### **Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

### **Conflict of Interest**

The author has no conflict of interest to declare that is relevant to the content of this article.

### **Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0

[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)