Evaluation of the Simulated Annealing and the Discrete Artificial Bee Colony in the Weight Tardiness Problem with Taguchi Experiments Parameterization

André S. Santos[1], Ana M. Madureira[2], Maria R. Varela[1],

[1]University of Minho / School of Engineering - Department of Production and Systems, Guimarães, Portugal
[2]Polytechnic Institute of Porto / School of Engineering - GECAD Research Group, Porto, Portugal
abg@isep.ipp.pt, amd@isep.ipp.pt, leonilde@dps.uminho.pt

Abstract. Meta-Heuristics (MH) are the most used optimization techniques to approach Complex Combinatorial Problems (COPs). Their ability to move beyond the local optimums make them an especially attractive choice to solve complex computational problems, such as most scheduling problems. However, the knowledge of what Meta-Heuristics perform better in certain problems is based on experiments. Classic MH, as the Simulated Annealing (SA) has been deeply studied, but newer MH, as the Discrete Artificial Bee Colony (DABC) still need to be examined in more detail. In this paper DABC has been compared with SA in 30 academic benchmark instances of the weighted tardiness problem (*1//*$\Sigma w_j T_j$). Both MH parameters were fine-tuned with Taguchi Experiments. In the computational study DABC performed better and the subsequent statistical study demonstrated that DABC is more prone to find near-optimum solutions. On the other hand SA appeared to be more efficient.

Keywords: Complex Combinatorial Problems; Simulated Annealing, Discrete Artificial Bee Colony; Taguchi Experiments; Weighted Tardiness Problem.

1 Introduction

Complex Combinatorial Problems (COPs) due to of their exceptional complex nature have been the focus of much research, in search of techniques that could address them efficiently and effectively. Operational Scheduling Problems (OSPs) given the number of constrains and the number of variables makes the problem extremely hard to approach with exact methods. Even if some OSPs can be addressed by simple heuristic rules, MHs have become the most used technique and have achieved the best results in the resolution of these problems [1,2].

Even with the development and study of several Meta-Heuristics, including the Simulated Annealing (SA) and the Discrete Artificial Bee Colony (DABC), there is still uncertainty on what makes certain MH more efficient, without a computational comparative analysis, since it is impossible to determine a priori, which parameters will be more efficient or even which MH will preform better.

This paper is structured as follows: section 2 overviews Operations Scheduling Problems (OSPs) and presents in detail the problem used in the computational study. Section 3 overviews Meta-Heuristics, and presents the inspiration and algorithms for the two MH implemented in the computational study: Simulated Annealing (SA) and Discrete Artificial Bee Colony (DABC). Section 4 presents the parameterization procedure for both meta-heuristics in detailed, which is conducted with Taguchi Experiments whose results are showed in this section. In section 5 the computational results and the statistical study are described. Section 6 summarizes the paper results and presents the conclusions

2 Operational Scheduling Problems

Operational Scheduling Problems (OSPs) are decision problems that distribute the operations over the available resources and disperse the operations in time, or sequence them, in each resource [3]. In other words, it is the process that determines which resources will be used in each operation and in which sequence the operations should be processed.

2.1 Single Machine Weighted Tardiness Problem

In the computational study the single machine weighted tardiness problem ($1//\Sigma w_jT_j$) will be used. It is a classical benchmark problem that sequences operations in order to minimize the tardiness, weighted by importance criterion. Given the pure sequence nature of the problem, most models of the problem use binary variables that can indicate the position of the operation in the production sequence, where $x_{jk}=1$ indicates that operation j occupies position k on the sequence, the relative position of the operation, where $x_{jk}=1$ indicates that operation j should be executed after operation I or the moment in which the operation should start to be executed, where $x_{jt}=1$ indicates that operation j should start to be executed in t. One model can be examined bellow [4,5]:

OF:

$$Min \sum_{k=1}^{n}\sum_{j=1}^{n} w_j\, t_k \qquad (1)$$

ST:

$$\sum_{j=1}^{n} x_{jk} = 1, \qquad \forall\, k \qquad (2)$$

$$\sum_{k=1}^{n} x_{jk} = 1, \qquad \forall\, i \qquad (3)$$

$$\sum_{j=1}^{n} p_j \sum_{u=1}^{k} x_{ju} - \sum_{j=1}^{n} d_j x_{jk} \leq t_k, \qquad \forall\, k \qquad (4)$$

If t_k is the tardiness of the operation in the k position of the sequence, w_j the weight of operation j and d_j the due date of operation j, (1) is the objective function, which is the minimization of the weighted tardiness. Equation (2) defines that each operation can only be assigned to one position in the sequence and (2) that each position can only have one operation assigned to it. Equation (4) defines the tardiness of the operation in position k. It is also important to mention that the problem deals with tardiness and, as such, the t_k can be defined as $max(L_k,0)$ or tardiness.

3 Meta-Heuristics

Meta-Heuristics can be described as iterative procedures that conduct a heuristic in the exploration of the solution space [6]. Like all heuristic techniques, meta-heuristics cannot assure the optimum, but they usually produce suitable solutions within the available computational time and outperform other approximation methods.

3.1 Simulated Annealing

Simulated Annealing (SA) is one of the best-known MH, proposed by Kirkpatrick et al. [7] and Cerny [8], and inspired by the thermic treatment of metals, which are heated and cooled down repeatedly. It modifies the basic neighborhood search procedure with the introduction of a mechanism that allows SA to overcome local-optimums, which controls the likelihood of solutions, worse than the current one, beeing accepted. SA selects one solution per-iteration, between the solutions in the vicinity of the current solution, but, unlike neighbourhood search techniques, it also accepts solutions that are worse than the current solution. Worse solutions are accepted as the current solution with a predetermined probability that decreases with the evolution of the search procedure. At the start almost all solutions are accepted as the new solution, but with the advance of the search procedure SA becomes more selective. The most common SA algorithm can be examined in [9].

3.2 Discrete Artificial Bee Colony

Discrete Artificial Bee Colony (DABC) is one of the most recent Meta-Heuristics based on the Artificial Bee Colony (ABC), which was proposed by Karaboga [10] and Pham et al. [11], adapted for discrete problems [12]. It works through the interaction of three phases, until the interruption criterion is met: the phase of the worker bees, the phase of opportunistic bees and the phase of the scout bees. Initially it is determined a food source (x_i) for each working bee. In the phase of working bees, each one will explore a solution in the neighbourhood (v_i) of its food source. In the opportunistic bees phases, the bee's wait for the performance of each food source to determine which food source they will explore. Once a food source is abandoned, the working bee will become a scout bee and starts to search for a new food source. The details about the DABC algorithm can be examined in [12].

4 Computational Study

SA and DABC were studied in 30 academic benchmark instances of the weighted tardiness problem (*1//*$\Sigma w_j T_j$) [1]. *1//*$\Sigma w_j T_j$ corresponds to the minimization of the weighted tardiness for *n* task. All the instances of the problem are available in the *ORLibrary* website [13] that also provides the optimum for each instance.

All computational trials were performed on a MacBook Pro with a 3GHz Intel® Core i7 processor, 16GB of RAM and Windows 10 64-bit. SA would conclude after 10,000 iterations while DABC would conclude after it explored the same number of solutions. All the parameters were define by analogy [14] and tuned with Taguchi Experiments [15] in the first instance of the problem used in the computational study not to over-fit the MH for each instance. Taguchi approach to Design of Experiments reduces the full factorial experiment with Orthogonal Arrays. For example, the full factorial experiment with 4 parameters and 3 levels would require 81 experiments, but with a L9 Orthogonal Array it requires only 9 experiments. Unlike full factorial experiments that seek the parameters that improve the mean outcome, Taguchi Experiments search for the parameters that improve the mean outcome and reduce the variance. Each parameter is evaluated in its Sign to Noise Ratio (S/N), which can be seen in [15]. One example of Taguchi Experiments in MH tuning can be seen in [16].

4.1 Simulated Annealing Parameterization

SA does not need an exhaustive parameterization procedure, since with enough time the MH performs well even with carefully determined parameters [17,18]. However, since the computational study is limited to 10,000 iterations the parameters need to be carefully set to maximize its performance. SA parameters are the Initial Temperature (T_i), the Cooling Ratio (CR), the Epoch Length (EL), the Neighborhood Structure (NS) and the Stoppage Condition (SC), which is set to 10,000 iterations. All the other parameters will be set into three levels and fine-tuned with Taguchi Experiments.

Kirkpatrick et al. [19] states that the Initial Temperature (T_i) should accept all new solutions. Other authors refer Initial Temperature in P_i between [0.7, 0.8] [20]. Several metrics to determine the Initial Temperature are presented in [21,22]. In order to determine this parameter, 20 random new solutions from the initial solutions with a P_i of 0.70, 0.75 and 0.80 which resulted in the rounded initial temperatures of 750, 1,000 and 1,250. Rose et al [22] state that the Geometric Cooling Ratio should result in a slow decrease of temperature, with α between [0.80, 0.99]. One metric to calculate the α is presented in [17]. Since the smallest Initial Temperature considered was of 750 and that the procedure will stop after 10,000 iterations the Cooling Ratio should be 0.97, 0,98 and 0,99. In terms of the Epoch Length, it should be identical to the size of the problem, in this case 50, as such, the Epoch Length was set in 50, 75 or 100. In this paper three simple NS were selected: **Insert** which consists of the solutions that can be obtained by removing one element of the solutions and inserting it into another position, **Swap** which consists of the solutions that can be obtained by interchanging two elements of the solution and **Transpose**, which consists of the solutions that are obtained by interchanging two adjacent elements of the solution. In [23,24] there are studies on the impact of the NS in SA.

The S/N for each parameter level can be examined in figure 1, which resulted in an Initial Temperature of 1,250, a Cooling Ratio (α) of 0.97, an Epoch Length of 100 and Swap as the Neighborhood Structure.

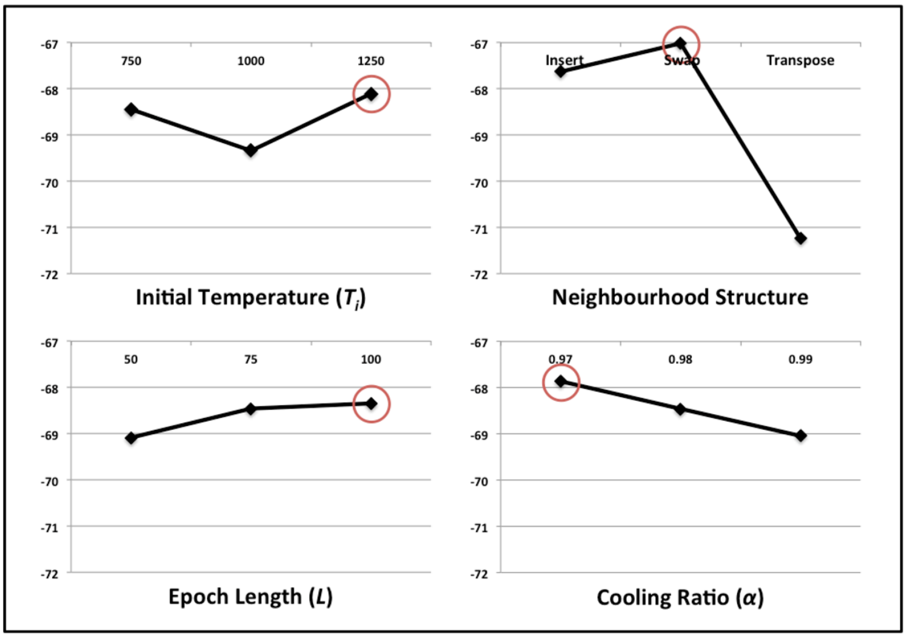

Figure 1. Simulated Annealing Parameters

4.2 Discrete Artificial Bee Colony Parameterization

ABC and DABC are MHs with less parameters [25], but unlike SA, which does not require an exhaustive parameterization procedure; the DABC Meta-Heuristic requires a meticulous parameterization to perform well. In [24] it is showed how the SA scales with the available time, without a rectification of the parameters, while DABC does not. DABC parameters are the Colony Size (L), the Limit (l), the Neighborhood Structure (NS) and the Stoppage Condition. Since SA was limited to 10,000 iterations, the number of iterations of the DABC would be limited to either 10,000/10, 10,000/20 or 10,000/3 iterations, in order to ensure that both MH explored a similar number of solutions. DABC will explore, at least, one solution per-bee per-iteration and, as such, depends as much on the stoppage condition as it does on the colony size. It is important to notice that DABC will explore extra solutions since whenever a food source is abandoned another solution will be explore in the same iteration, however the number of solutions explored by scout bees cannot be estimated. One extra parameter was considered, which was the proportion of worker/scout and opportunistic bees, which is usually fixed at 50%, however the increasing the number of worker/scout bees should increases the diversity of the MH and the increase of the opportunistic bees will increase the intensity of the MH.

Kockanat & Karagoba [26] studied the influence of the Colony Size (*L*) and concluded that the performance of the MH is not overly dependent on the Colony Size, which can be balanced with Stoppage Condition. Other authors refer that the *L* should be related to the size of the problem [27], but also concluded that it did not require detail tuning since its impact on the performance is limited. In the Taguchi Experiments, *L* of 10, 20 and 30 will be tested with the number of iterations previously mentioned to ensure that both MH explored the same number of solutions. For the Limit (*l*) number, which determines the number of iterations without improvement before a food source is abandoned, numerous authors [26-28] concluded it required to be meticulously established for larger *L*. Liu & Liu in [29] present a method to calculate the *l*, based on the *L* and the dimension of the problem. In the Taguchi Experiments the *l* tested will be 500, 550 and 600, determined with the Liu & Liu method [29] in a problem with 50 tasks and *L* of 10, 20 or 30. In the case of the Bees Proportional, which as previously mentioned is not a usual parameter, but can be similar to the set of locked scout-bees. Usually this is fixed at 50%/50%, however 40%/60%, 60%/40% and the usual 50%/50% will also be tested. Finally the Neighborhood Structure (NS) is almost a ubiquitous parameter, which can have an impact on the performance of all MH. For DABC the same three Neighborhood NS used in SA, which are **Insert**, **Swap** and **Transpose**.

The S/N for each parameter level can be examined in figure 2, which resulted in a Colony Size of 10 and consequentially 10,000/10 iterations, a Limit of 600, a Bees Proportion of 100 and Swap as the NS.

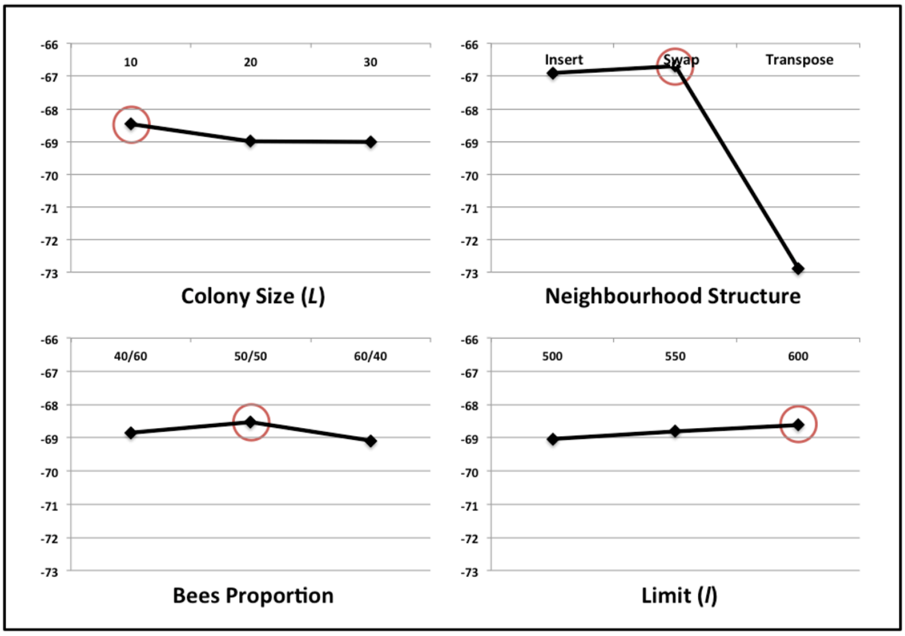

Figure 2. Discrete Artificial Bee Colony Parameters

5 Computational Results

SA and DABC were run 3 times for each of the 30 iterations of the problem with the solution from the best test-run being used in the statistical analyses. Overall both MH performed well, but it appears that the DABC performed better than SA, as can be seen in fig. 3, which shows the best solution. SA found the optimum solution in 5 instances, 16.(6)%, while DABC found the optimum in 11 instances of the 30 instances, 36.(6)%. If DABC performed better than SA, it also took more computational time in almost all instances. SA took less than 0.001s in 24 of the 30 instances and up to 0.031s in the other 6 instances. DABC took more computational time, less than 0.001s in 7 of the instances and up to 0.032s in the other.

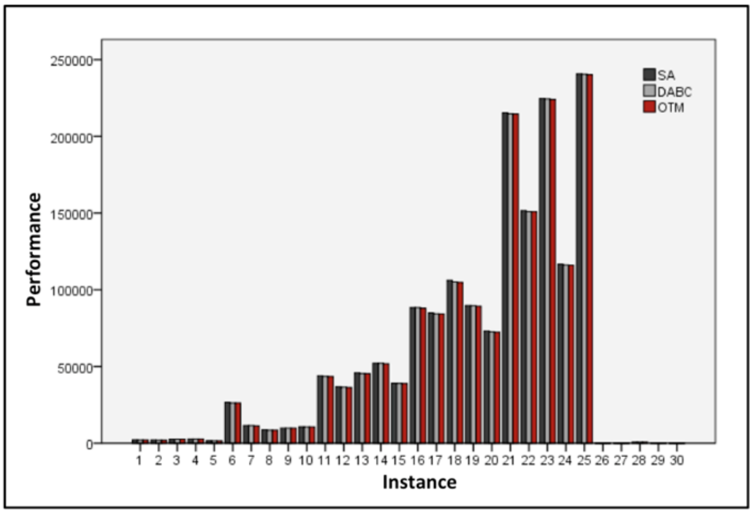

Figure 3. SA and DABC Computational Results

SA achieved better results, in the computational study, than DABC, however to better evaluate this performance variance it is necessary to statistically examine the obtained results. Both MH solutions were normalized around the best-known solution for each instance, available in the ORLibrary, and measured using the mean percent deviation from those solutions, which Silberholz & Golden [30] state is the best metric to compare MH across several instances of one problem. It is preferable to use the mean percent deviation from the optimum, or in this case the best-known solution, over the absolute difference, since it allows MH to be compared across all the different instances. For example, a 10 units difference, in whatever metric is used, cannot represent the same in an instance with an optimum solution of 10 and an optimum of 1000.

Once both MH solutions were normalized around the best-known solution, the superior performance of the DABC becomes evident. DABC obtained 28 near-optimal solutions, or solutions with deviation from the best-known solution of less than 1%, while SA obtained 16, which means that DABC found near-optimums in almost all instances, 93.(3)%, while SA found near-optimums solutions in half of the instances, 53.(3)%. SA also found 5 solutions with substantial difference from the optimum, or solutions with deviation from the best-known solution of over 5%. DABC achieved a mean deviation from the best-know solution of 0.22% while SA mean deviation was of 1.54%. DABC performance also appeared constant, with standard deviation of 0.30 and 1.91 for SA. In fig. 4 the boxplot of the mean deviation from best-known solution, of SA and DABC can be examined.

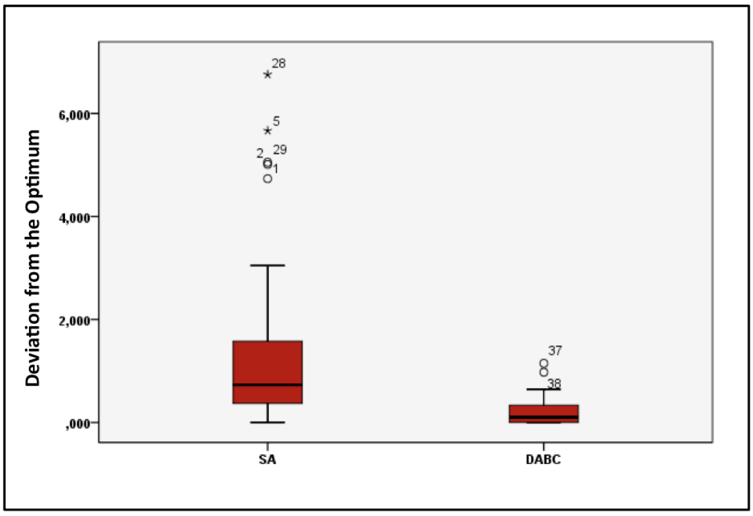

Figure 4. Boxplot of the Deviation from the Best-Known Solution

In the computational study DABC appeared to have achieved the best results, but to determine if the meta-heuristics have the same probability of achieving near optimum solutions (>1% from the best-known solution) a statistical inference test needs to be used, in this case the Chi-Square to determine if the probability to find near-optimum solutions is independent of the MH. In table I the results can be examined.

Table 5. Chi-Square Results

	Value	*df*	*Asym. Sig. (2-Sidded)*
Pearson Chi-Square	12.273	1	0.000
Continuity Correction	10.313	1	0.001
Likelihood Ratio	13.439	1	0.000
N. of Valid Cases	60		

Results from the Chi-Square test demonstrate that the likelihood of near optimum solutions is not identical for both meta-heuristics (*P-Value* 0.001). It is also possible to conclude that DABC is more likely to find a solution closer to the optimum than SA (p-value 0.001/2), which reinforces what was apparent in the computational study.

6 Conclusions

In this paper the performance of SA and DABC were studied in the weighted tardiness problem ($1||\Sigma w_jT_j$). SA is a well known meta-heuristic, but DABC, which is a discreet version of ABC, is more recent and its behavior has not been examined with the required detail. SA and DABC were studied in 30 instances available in the *ORLibrary*. Both MH parameters were tuned with Taguchi Experiments, which simplified the often-extensive parameterization procedure. SA search procedure concluded after 10,000 iterations while DABC after 10,000/10 iterations.

In the computational study DABC appeared to perform better than SA in most instances. SA found the optimum solution in 5 instances, 16.(6)%, while DABC found the optimum in 11 instances of the 30 instances, 36.(6)%. Once the MH solutions were normalized, DABC found near-optimums in 93.(3)% of the instances, while SA found near-optimums solutions in around half of the instances, 53.(3)%. In order to determine if both have the same probability of achieving near-optimums the Chi-Square was used. It demonstrated that DABC is more likely to find near-optimum solutions than SA. On the other hand, DABC took more time than SA to explore the same number of solutions. SA took less than 0.001s in 24 of the 30 instances and up to 0.031s in the other 6 instances. DABC took more computational time, less than 0.001s in 7 of the instances and up to 0.032s in the other.

Acknowledgments. This work is supported by FEDER Funds through the “Programa Operacional Factores de Competitividade - COMPETE” program and by National Funds through FCT “Fundação para a Ciência e a Tecnologia” under the project: PEst-OE/EEI/UI0760/2014, and PEst2015-2020.

References

1. F. Xhafa and A. Abraham ” *Metaheuristics for Scheduling in Industrial and Manufacturing Applications Series: Studies in Computational Intelligence* (128)”, Springer, New York, 2008.
2. H. I. Osman and J. P. Kelly, ”*Meta-Heuristics: An Overview*”, Meta-Heuristics Theory and Applications. Kluwer Academic Publishers, 1996.
3. M. L. Pinedo, “*Scheduling Theory, Algorithms, and Systems* (Forth Edition)”, Springer, New York, 2012.
4. A. B. Keha, K. Khowala and J. W. Fowler ”*Mixed Integer Programming Formulations for Single Machine Scheduling Problems*”, Computers & Industrial Engineering, 2009, 56, 357-367.
5. K. Khowala, A. Keha and J. Fowler ”*A Comparison of Different Formulations for the Non-Preemptive Tardiness Scheduling Problem*”, Proc. International Conference on Scheduling: Theory & Applications, 2005, 643-651.
6. H. I. Osman and J. P. Kelly, “*Meta-Heuristics: An Overview. Meta-Heuristics Theory and Applications*”, Kluwer Academic Publishers, 1996.
7. S. Kirkpatrick, C. D. Gelatt and P. M. Vecchi ”*Optimization by Simulated Annealing*” Science, 1983, 220, 671-680.
8. V. Cerny ”*A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulated Annealing Algorithm*”, Journal of Optimization Theory and Applications, 1985, 45, 41-51.

9. I. Boussaid, J. Lepagnot and P. Siarry ”*A Survey on Optimization Metaheuristics*”, International Journal of Information Sciences, 2013, 237, 82-117
10. D. Karaboga, “*An Idea Based On Honey Bee Swarm For Numerical Optimization*,” Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
11. D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. M. Zaidi, “*The Bees Algorithm. Technical Note*,” Cardiff University, 2005.
12. D. Karaboga and B. Gorkemli, “*A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem*,” Proc. of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 2011, 50-53.
13. Beasley, J. E. (1990). *ORLibrary*: http://www.brunel.ac.uk/~mastjjb/jeb/info.html.
14. E. Montero, M. Riff and B. Neveu, “*A Beginner’s Guide to Tuning Methods*”, Applied Soft Computing, 2014, 17, 39-51.
15. G. Taguchi “*Introduction to Quality Engineering: Designing Quality into Products and Processes*”, Quality Resources, 1986.
16. B. Nadir, M. Zandieh and S. M. T. Fatemi Ghomi, “*Scheduling Job Shop Problems with Sequence-Dependent Setup Times*”, International Journal of Production Research, 2009, 47(21), 5959-5976.
17. M. W. Park and Y. D. Kim, “*A Systematic Procedure for Setting Parameters in Simulated Annealing Algorithm*. Computers & Operations Research”, 1998, 25(3), 207-217.
18. S. Anily and A. Federgruen, “*Simulated Annealing Methods with General Acceptance Probabilities*”, Journal of Applied Probability, 1987, 24, 657-667.
19. S. Kirkpatrick, C. D. Gelatt and P. M. Vecchi, “*Optimization by Simulated Annealing*”, Science, 1983, 220, 671-680.
20. E. G. Talbi, “*Meta-heuristics: From Design to Implementation*”, John Wiley & Sons, 2009.
21. R. W. Eglese, “*Simulated Annealing: A Tool for Operational Research*”, European Journal of Operational Research, 1990, 46, 271-281.
22. J. Rose, W. Klebsch and J. Wolf, “*Temperature Measurement and Equilibrium Dynamics of Simulated Annealing Placement*”, IEEE Transactions on Computer Aided Design, 1990, 9, 253-259.
23. K. M. Chek, J. B. Goldberg and G. Askin, “*A Note on the Effect Neighborhood Structure in Simulated Annealing*”, Computers & Operations Research, 1991, 18, 537-547.
24. A. S. Santos, A. M. Madureira and M. L. Varela, “*Study on the Impact of the NS in the Performance of Meta-Heuristics in the TSP*”, IEEE International Conference on Systems, Man, and Cybernetics, 2016, 1110-1115.
25. G. Yan and C. Li, ”*An Effective Refinement Artificial Bee Colony Optimization Algorithm Based on Chaotic Search and Application for PID Control Tuning*“, Journal of Computational Information Systems, 2011, 7(9), 3309-3316.
26. S. Kockanat and N. Karaboga, ”*Parameter Tuning of Artificial Bee Colony Algorithm for Gaussian Noise Elimination on Digital Images*“, Proc. of the International Symposium on Innovation in Intelligent Systems and Applications (INISTA), 2013, 1-4.
27. B. Akay and D. Karaboga, ”*Parameter Tuning for the Artificial Bee Colony Algorithm*”, Computational Collective Intelligence, Semantic Web, Social Network and Multiagent Systems, Lecture Notes in Computer Science, 2009, 5796, 608-619.
28. M. S. Kiran and M. Gunduz, ”*The Analysis of Peculiar Control Parameters of Artificial Bee Colony Algorithm on the Numerical Optimization Problems*”, Journal of Computer and Communications, 2014, 2, 127- 136.
29. Y. F. Liu and S. Y. Liu, ”*A Hybrid Discrete Artificial Bee Colony Algorithm for Permutation Flowshop Scheduling Problem*”, Applied Soft Computing, 2013, 13, 1459-1463.
30. J. Silberholz and B. Golden, “*Comparison of Metaheuristics*”, in Handbook of Metaheuristics, M. Gendreau and J. Potvin, Eds. Springer, 2010, 625-640