International Journal of Industrial Engineering & Production Research

June 2013, Volume 24, Number 2
pp. 143-150

http://IJIEPR.iust.ac.ir/

ISSN: 2008-4889

Efficient and Robust Parameter Tuning for Heuristic Algorithms

H. Akbaripour & E. Masehian*

Hossein Akbaripour, *Industrial Engineering Department, Tarbiat Modares University, Tehran, Iran, h.akbaripour@modares.ac.ir*
Ellips Masehian, *Industrial Engineering Department, Tarbiat Modares University, Tehran, Iran, masehian@modares.ac.ir*

KEYWORDS

Parameter Tuning,
Design of Experiments,
Signal to Noise (S/N) ratio,
Shannon Entropy,
VIKOR,
Simulated Annealing,
Genetic Algorithms

ABSTRACT

The main advantage of heuristic or metaheuristic algorithms compared to exact optimization methods is their ability in handling large-scale instances within a reasonable time, albeit at the expense of losing a guarantee for achieving the optimal solution. Therefore, metaheuristic techniques are appropriate choices for solving NP-hard problems to near optimality. Since the parameters of heuristic and metaheuristic algorithms have a great influence on their effectiveness and efficiency, parameter tuning and calibration has gained importance. In this paper a new approach for robust parameter tuning of heuristics and metaheuristics is proposed, which is based on a combination of Design of Experiments (DOE), Signal to Noise (S/N) ratio, Shannon entropy, and VIKOR methods, which not only considers the solution quality or the number of fitness function evaluations, but also aims to minimize the running time. In order to evaluate the performance of the suggested approach, a computational analysis has been performed on the Simulated Annealing (SA) and Genetic Algorithms (GA) methods, which have been successfully applied in solving respectively the n-queens and the Uncapacitated Single Allocation Hub Location combinatorial problems. Extensive experimental results showed that by using the presented approach the average number of iterations and the average running time of the SA were respectively improved 12 and 10.2 times compared to the un-tuned SA. Also, the quality of certain solutions was improved in the tuned GA, while the average running time was 2.5 times faster compared to the un-tuned GA.

1. Introduction

One of the most important consequences of progress in modern sciences and technologies has been to understand and model real-life problems realistically and in more details. The natural outcome of this fact is the rapid increase in dimensions and complexity of solvable problems. With the growing complexity of today's large-scale problems, finding optimal solutions by merely exact mathematical methods has become more difficult. Due to efficiency concerns in terms of solution quality, the need for finding near-optimal solutions within acceptable times justifies the use of heuristic and metaheuristic approaches. These methods have been introduced relatively recently in the field of combinatorial optimization. A *heuristic* can be defined as "a generic algorithmic template that can be used for finding high quality solutions for hard combinatorial

* **Corresponding author: Ellips Masehian**
Email: *masehian@modares.ac.ir*
Paper first received Jan. 28, 2013, and in accepted form Feb. 19, 2013.

optimization problems" [1]. Heuristic approaches have proved themselves in many large-scale optimization problems by offering near-optimal solutions while no optimal solutions had been found through exact approaches.

Nonetheless, heuristics, in general, have several parameters that may have a great influence on the efficiency and effectiveness of the search and so need to be 'tuned' before they can yield satisfactory results. Actually, *Parameter Tuning* is a common method for improving efficiency and capability of heuristic algorithms in finding optimal or near-optimal solutions in reasonable times.

Though parameter tuning introduces larger flexibility and robustness in problem solving, it requires careful initialization.

In fact, for an unsolved problem, it is not clear how to determine *a priori* which parameter setting will produce the best result. Moreover, optimal parameter values depend mainly on the nature of the problem, the instance to deal with, and the utmost solving time available to the user. Since a universally optimal parameter values set for a given heuristic does not exist, many researchers have tried to exploit the effectiveness of parameter tuning for specific heuristic algorithms.

Barr *et al*. in [2] studied the important issues of designing and reporting in computational experiments with heuristic methods and noted that the selection of parameter values that drive heuristics is itself a scientific endeavor and deserves due attention. In [3] some standard statistical tests and experimentation design techniques were proposed by Xu *et al*. to improve a specific Tabu Search algorithm. Although the focus of the paper was to improve the performance of a particular algorithm, the implications are general. In [4] and [5] some gradient descent methods were proposed based on minimizing the generalization error, which allowed a larger number of parameters to be considered. However, in practical problems such methods may be affected by the presence of local extrema [6]. Figlali *et al*. show in [7] the lack of parameter design approaches when it comes to parameter-tuning using Ant Colony Optimization (ACO).

In majority of the investigated publications, researchers have a weak motivation for their choice of parameters, if any. In this paper, a new general framework for parameter tuning of metaheuristics is presented. In contrast to other automated parameter tuning methods such as in [8] or [9] for PSO, and the ones mentioned in [10] for ACO, the method proposed here is applicable for any metaheuristic.

In this paper a robust parameter tuning approach is presented which is based on combination of Design of Experiments (DOE), Signal to Noise ratio (S/N), Shannon entropy and VIKOR methods. The proposed approach not only tries to optimize the solution quality or the number of fitness function evaluations, but also considers minimizing the algorithm's runtime and variance of these objectives.

The rest of the paper is organized as follows: In section 2 the proposed parameter tuning method is presented. Section 3 presents the effects of tuning the Simulated Annealing (SA) algorithm for solving the *n*-queens problem, and Section 4 provides the effects of tuning the Genetic Algorithms (GA) method for solving the Uncapacitated Hub Location Problem (UHLP). Finally, conclusions come in Section 5.

2. The Proposed Parameter Tuning Method

There are two different strategies for parameter tuning: Offline parameter initialization (or meta-optimization), and online parameter tuning strategy. In the off-line parameter initialization the values of different parameters are fixed before the execution of the metaheuristic, whereas in the online approach the parameters are controlled and updated dynamically or adaptively during the execution of the metaheuristic [11].

In off-line approach, the metaheuristic designer tunes one parameter at a time, and its optimal value is determined empirically. As a result no interaction between parameters is studied. This sequential optimization strategy (i.e., one-by-one parameter) does not guarantee finding the optimal setting even if an exact optimization setting is performed. To overcome this problem, experimental design is used [12]. In this paper, a robust parameter tuning approach is presented which is based on combination of design of experiments, signal to noise, Shannon entropy and VIKOR methods. It concurrently pursues four goals of solution quality, number of fitness function evaluations, minimizing algorithm's runtime, and variance of these objectives. The proposed parameter tuning approach is composed of three phases: (1) Introducing levels of parameters, (2) Design of experiments, and (3) Modified Entropy-VIKOR. Each phase will be describe in the following.

2-1. Introducing Levels of Parameters

In this part of the proposed method key factors (parameters) of heuristic or metaheuristic algorithms which have a significant effect on the efficiency and effectiveness of the search for solving the particular problem is determined. Each factor may have different levels that are determined by the designer. Indeed, each factor represents the parameter to vary in the experiments and the levels indicate different values of the parameters, which may be quantitative (e.g., mutation probability) or qualitative (e.g., neighborhood generation type).

2-2. Design of Experiments

The reason for using the Design of Experiments (DOE) method is to obtain the largest information possible with the least number of experiments. An experiment is a collection of planned efforts in which purposeful

changes are made to the controlling factors of a process so that the reasons for changes in objective function values are identified [13]. In this study, the DOE is employed in order to discover the effect of each parameter and determine their optimal levels by conducting minimal experiments.

Two of the most well-known DOE approaches are Factorial design and Taguchi design. The main drawback of the factorial design is its high computational cost especially when the number of parameters (factors) and their domain values are large; hence, a very large number of experiments must be realized [14]. However, with respect to cost, time and theories of statistics, high number of experiments is not necessary and cost effective. Therefore, using a Taguchi design is proposed, which requires less number of designs. In order to define a proper Taguchi design, the minimum number of degrees of freedom must be calculated and a proper orthogonal array can be selected in this way. By generating the orthogonal array, levels of parameters are determined in each experiment.

2-3. Modified Entropy-VIKOR

The proposed approach for determining the optimal levels of parameters considers minimizing four objective functions of solution quality or number of fitness function evaluations, overall runtime of algorithm and variance of these objectives simultaneously. This is done by utilizing the Signal to Noise (S/N) ratio for combining the mean and variance of each objective, and then employing the Entropy-VIKOR as a multi-criteria decision making approach to combine the S/N ratios. The approach is described in eight steps as follows:

Step 1. Alternative performance matrix generation:

In the performance matrix of VIKOR each column is assigned to a criterion and each row is assigned to an alternative [15]. In the proposed parameter tuning approach, experiments from Taguchi design and signal to noise ratios are respectively considered to be the alternatives and criteria. The structure of the performance matrix is shown in (1), in which SN_{Cj} and SN_{Tj} are signal to noise ratios for the total cost and the algorithm's runtime in the j-th experiment of Taguchi design, respectively. Since both the cost and runtime objectives should be minimized, their signal to noise ratios are calculated by (2):

$$D = \begin{bmatrix} SN_{C_1} & SN_{T_1} \\ SN_{C_2} & SN_{T_2} \\ \dots & \dots \\ SN_{C_j} & SN_{T_j} \end{bmatrix} \tag{1}$$

$$SN_{ij} = -10 \cdot \log\left(\frac{1}{m}\sum_{k=1}^{m} y_{ijk}^2\right), \quad i = C \text{ or } T, \quad j = 1,2,...,J \tag{2}$$

in which y_{ijk} is the value of i^{th} objective for replicate k of the j-th experiment and m indicates the number of replications in each experiment.

Step 2. Obtaining the weights based on Shannon's entropy:

The procedure of Shannon's entropy for obtaining the weights in a Multi Attribute Decision Making problem can be expressed as follows:

2.a) The performance matrix is normalized using (3):

$$P_{ij} = \frac{SN_{ij}}{\sum_{j=1}^{J} SN_{ij}}, \quad i = C \text{ or } T, j = 1,2,...,J. \tag{3}$$

2.b) Entropy h_i is computed using (4) where h_0 is the entropy constant equal to $1/\ln(J)$:

$$h_i = -h_0 \sum_{j=1}^{J} P_{ij} \cdot \ln(P_{ij}), \quad i = C \text{ or } T. \tag{4}$$

2.c) The degree of diversification is calculated as $d_i = 1 - h_i$, $i = C$ or T.

2.d) The degree of importance of attribute i is obtained as:

$$W_i = \frac{d_i}{\sum_{s=C}^{T} d_s}, \quad i = C \text{ or } T \tag{5}$$

Step 3. VIKOR performance matrix normalization:

For this purpose the transformation formula (6) is used:

$$NSN_{ij} = \frac{SN_{ij}}{\sqrt{\sum_{J=1}^{J} SN_{ij}^2}}, \quad i = C \text{ or } T, j = 1,2,...,J \tag{6}$$

in which NSN_{ij} represents the normalized value of the i-th objective under the j-th experiment.

Thus, the matrix V is obtained as follows:

$$V = \begin{bmatrix} NSN_{C_1} & NSN_{T_1} \\ NSN_{C_2} & NSN_{T_2} \\ \dots & \dots \\ NSN_{C_J} & NSN_{T_J} \end{bmatrix} \tag{7}$$

Step 4. Determination of maximum and minimum values of all criteria:
Since the signal to noise ratios for objectives are of maximization type, maximum values (V^+) and minimum values (V^-) of all criteria are obtained as follows:

$$V_i^+ = \max_j \left\{ NSN_{ij} \right\},\ V_i^- = \min_j \left\{ NSN_{ij} \right\} \tag{8}$$

Step 5. Utility and regret measures computation:
The Utility and Regret measures S_i and R_i are computed using (9) and (10), respectively:

$$S_j = \sum_{i=C}^{T} W_i \frac{(V_i^+ - V_{ij})}{(V_i^+ - V_i^-)} \tag{9}$$

$$R_j = \max \left\{ W_i \frac{(V_i^+ - V_{ij})}{(V_i^+ - V_i^-)} \right\} \tag{10}$$

Step 6. VIKOR index calculation
The VIKOR indices Q_j are computed using (11) such that v is the weight for the strategy of maximum group utility, and $1 - v$ is the weight of the individual regret. v is usually set to 0.5.

$$Q_j = v\left[\frac{S_j - S^-}{S^+ - S^-}\right] + (1-v)\left[\frac{R_j - R^-}{R^+ - R^-}\right] \tag{11}$$

$$S^- = \min S_j,\ S^+ = \max S_j$$

$$R^- = \min R_j,\ R^+ = \max R_j$$

Step 7. Ranking the alternatives:
The alternatives are ranked by Q_j. The less is the value of Q_j, the better is the decision of alternatives.

Step 8. Determining optimal levels of parameter:
By aggregation of signal to noise ratios for solution quality, number of fitness function evaluations, and runtime objectives into the *VIKOR* index, it is possible to determine the optimal levels of parameters through calculation of average *VIKOR* indices of experiments for different levels of parameters and drawing factor plots.

3. Tuning the SA for the *n*-Queens Problem

The *n*-queens problem is a classical combinatorial optimization problem in artificial intelligence. The objective of the problem is to place n non-attacking queens on an $n \times n$ chessboard by considering the chess rules. Although the problem itself has an uncomplicated structure, it has been broadly utilized to develop new intelligent problem solving approaches.
There are three variants of the *n*-queens problem: (1) finding all solutions of a given $n \times n$ chessboard, (2) generating one or more, but not all solutions, and (3) finding only one valid solution. A valid solutions to the 8-queens is shown in Figure 1.

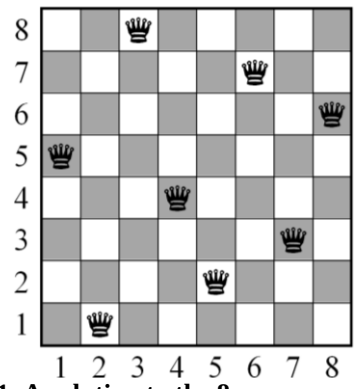

Fig. 1. A solution to the 8-queens problem

Due to the NP-hardness of the *n*-queens problem, metaheuristic techniques are appropriate choices for solving it. For example, Martinjak and Golub [16] represented three metaheuristic algorithms Simulated Annealing, Tabu Search and genetic for finding only one valid solution in the *n*-queens problem. The results show that SA can act better in dealing with *n*-queens problem.
In this paper three vital factors of SA [16] (neighborhood type, cooling rate and initial temperature) is chosen for parameter tuning. Each factor has three different levels, which are shown in Table 1. By considering two degrees of freedom for each of the factors and a degree of freedom for the total mean, we should have at least (3×2)+1=7 run experiments. So, L_9 is chosen as the orthogonal array, and it is shown in Table 2.
By running the Simulated Annealing (SA) algorithm proposed in [16] on a PC with an Intel™ 2.27 GHz processor with 4.00 GB RAM on Matlab™, The VIKOR indices (Q_i) are calculated for each experiment as Table 2.

Tab. 1. Introducing levels of factors in SA

Factors	Index of levels	Levels
Neighborhood type	1	random swap
	2	effective swap [17]
	3	combination of levels 1 and 2
Cooling rate (α)	1	0.90
	2	0.95
	3	0.99
Initial temperature (T_0)	1	n
	2	$5n$
	3	$10n$

Tab. 2. Taguchi Orthogonal array (L_9)

Neighborhood Type	α	T_0	Q_j (VIKOR index)
1	1	1	0.93660
1	2	2	0.81000
1	3	3	1.00000
2	1	2	0.11070
2	2	3	0.00000
2	3	1	0.16420
3	1	3	0.23350
3	2	1	0.27200
3	3	2	0.48330

As mentioned before, for choosing the best combination of parameters it should be considered that which Q_j is closer to 0. Also, Q_j is used to draw the charts of Figure 2. As it is demonstrated in Figure 2, the best levels for the three factors are levels 2, 2 and 3. This means that by selecting the Effective Swap approach [17] for generating neighbourhoods, α=0.95 for cooling rate, and ten times the problem size for the initial temperature results in better solutions for various sizes in n-queens problem. It should be noted that the n-queens problem has been tuned for n=200 but the results can be generalized to other sizes. In this paper, the SA algorithm of Martinjak and Golub was coded for solving the n-queens problem and was run 10 times for each size. The number of iterations and runtimes were computed and compared with the SA algorithm tuned by our proposed method, as reported in Table 3. Also, Figures 3 and 4 demonstrate the comparison between average number of iterations and average runtimes in SA and Tuned-SA, respectively. It is obvious from Table 3 and Figures 3 and 4 that the Tuned-SA has outperformed the SA. The average runtimes of Tuned-SA was about 10.2 times faster than the SA algorithm, and the average iteration of Tuned-SA is 12 times less than that of the SA. At the same time, the iteration variance of the Tuned-SA was less than the SA algorithm. Therefore, the Tuned-SA outperforms SA in acheiving all the four objectives. This implies that by employing the proposed parameter tuning approach, in addition to decreasing the number of iterations in the SA, its computational speed is increased as well.

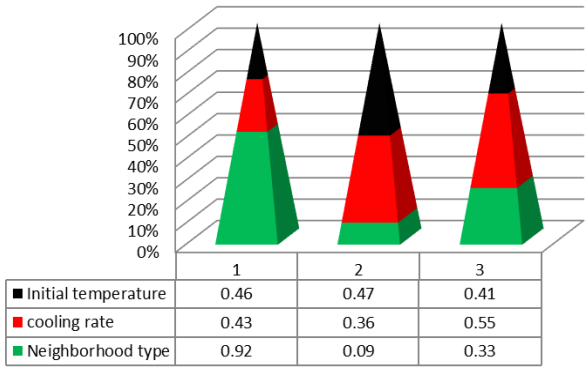

Fig. 2. Demonstration of the best levels for SA [16] using average VIKOR index

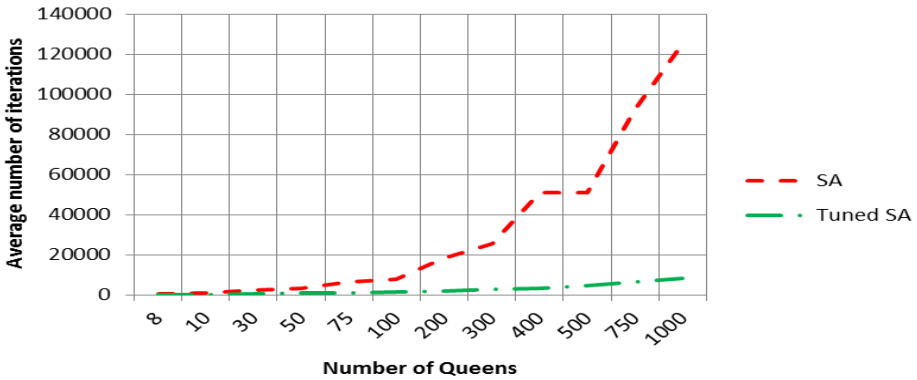

Fig. 3. Comparison of the performances of Tuned-SA and basic SA

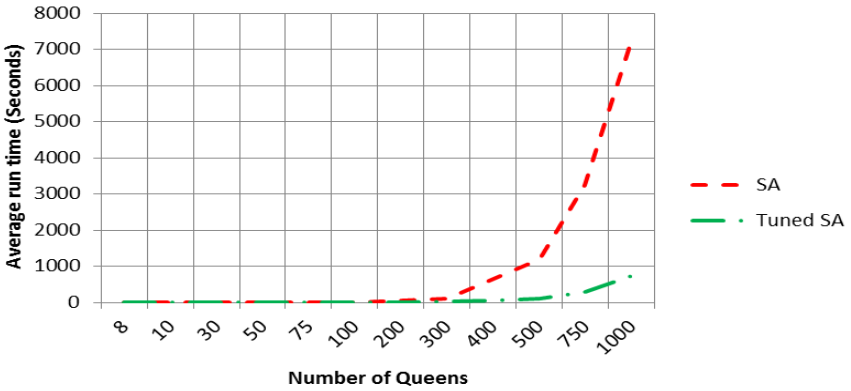

Fig. 4. Demonstration of the Tuned-SA running time versus basic SA performance

Tab. 3. Comparison of SA [16] and Tuned-SA algorithm for *n*-queens problem

n	SA [16]					Tuned-SA (proposed)				
	Number of iterations (in 10 runs)			Average Runtime (seconds)	Convergence Index	Number of iterations (in 10 runs)			Average Runtime (seconds)	Convergence Index
	Min	Max	Average			Min	Max	Average		
8	57	908	481.8	0.11	7.39	1	163	68.1	0.04	1.63
10	391	1508	921.1	0.19	12.61	38	470	129.8	0.14	2.27
30	1520	4756	2544.9	0.90	11.16	429	938	641.9	0.50	1.68
50	1698	7821	3301.7	2.81	7.93	733	1211	902.9	0.82	1.42
75	2433	9895	6275.7	5.29	6.31	988	1398	1198.4	1.71	1.22
100	4668	12109	7858.9	8.90	5.64	1202	1851	1468.6	2.41	1.17
200	8951	27809	18250.4	60.69	4.23	1887	2491	1996.6	9.09	0.98
300	16882	34521	25739.5	110.91	3.78	2512	3811	2828.6	21.39	0.77
400	26851	64791	50922.1	645.21	3.50	3590	4248	3498.1	52.32	0.82
500	41544	82281	51189.7	1203.19	3.41	3800	5088	4710.6	111.12	0.84
750	60974	112011	92147.3	3259.00	3.18	5959	6729	6250.2	301.80	0.75
1000	99841	199118	124957.1	>2 hours	1.92	8052	8421	8299.8	719.94	0.93

4. Tuning the GAs for the Uncapacitated Hub Location Problem

In general, the hub facilities location problem could be considered as a location-allocation problem in which the number of hub facilities, their location, and allocation of non-hub nodes are determined in a way to minimize the network's total cost (sum of fixed and variant costs). Considering different constraints, various versions of the hub location problem have been introduced.

These constraints are: (1) capacity restrictions on the maximum amount of flow a hub can collect, (2) single or multiple allocations of non-hub nodes to the hubs, and (3) the number of selected hubs being settled or left as a decision variable.

The ultimate objective in all the versions of the problem is to find the location of the hubs and the allocation of other nodes to them so that the total cost is minimized. In this paper we deal with the Uncapacitated Single Allocation Hub Location Problem (USAHLP), while no capacity constraint is considered for hub facilities and each non-hub node is allocated to a single hub facility. In the USAHLP the number of hub facilities are considered as a decision variable and with establishing each hub facility a fixed cost is imposed on the system. USAHLP is known to be NP-hard even if the locations of hub facilities are considered fixed. The problem is hard to solve in polynomial time by deterministic techniques due to their exponential time complexity. So heuristic and metaheuristic techniques are used for solving these kinds of problems in reasonable times. Topcuoglu *et al*. in [18] proposed a GA heuristic to find the number and locations of hub facilities and allocation of non-hub nodes, which was capable of attaining satisfactory solutions in short times for the USAHLP.

Their Developed GA does not have a specific approach for tuning the parameters, and a suitable level for the parameters was determined by try and error. In order to further assess the efficiency of our proposed robust parameter tuning approach, the genetic algorithm developed in [18] was chosen as one the best algorithms for the USAHLP in the literature and was coded. Their GA was then tuned using the proposed parameter tuning approach and the results obtained from solving various problem sizes are summarized in Table 4.

It is noted that the comparisons made here are based on benchmark problems taken from the CAB dataset, which is a standard dataset considered as benchmark for evaluating algorithms for different hub location problems.

The CAB data set has been developed based on airline passenger flows between 25 US cities. In the CAB, the four scenarios of the first 10 cities, the first 15 cities, the first 20 cities, and all the 25 cities are considered. For each problem size the discount factor (α) is fixed and set to 0.1, 0.2, 0.4, 0.6, 0.8 and 1. For each discount factor, fixed costs for establishing hub facilities are set to 100, 150, 200 and 250 such that it is equal for all nodes. Thus, the CAB dataset consists of 80 different problems.

Based on computational results the Tuned GA achieved optimal solutions while the GA [18] algorithm achieved optimal solutions for all problems except one ($n = 25$, $f = 100$, $\alpha = 1$). This instance is shown in bold in Table 5. Also, it is observed that the runtime by Tuned-GA is almost 2.5 times less than that of the GA. Figure 5 gives comparisons of the runtimes of GA and Tuned-GA for different sizes of CAB dataset problems.

This indicates that employing proposed parameter tuning approach not only increase the algorithm's capability to obtain better solutions but also decreases the algorithm's runtime significantly.

Table 4. Optimal levels of effective parameters for GA [18]

Parameter	Levels	Average VIKOR index	Optimal level
Crossover operator type	Single-point	0.67	Two-points
	Two-points	0.11	
Mutation operator type	Shift	0.28	Exchange
	Exchange	0.21	
	Cyclic ejection	0.54	
	Shift and exchange combination	0.98	
Crossover operator rate	0.5	0.38	0.6
	0.6	0.05	
	0.7	0.15	
	0.8	0.49	
Mutation operator rate	0.3	0.41	0.4
	0.4	0.32	
	0.45	0.89	
	0.5	0.96	
Population size	100	0.11	150
	150	0.09	
	175	0.23	
	200	0.55	
Number of iterations	100	0.58	150
	150	0.22	
	175	0.31	
	200	0.72	

Table 5. computational results of GA and Tuned-GA for the CAB dataset when *n*=25

α	*f*	Optimal		GA		Tuned GA	
		Cost	Hubs	Cost	Time (sec.)	Cost	Time (sec.)
0.2	100	1029.63	24،17،12،4	1029.63	6.29	1029.63	3.34
	150	1217.34	17،12،4	1217.34	5.42	1217.34	3.06
	200	1367.34	17،12،4	1367.34	5.9	1367.34	2.8
	250	1500.9	12 ،20	1500.9	2.6	1500.9	1.2
0.4	100	1187.51	17،12،4،1	1187.51	6.41	1187.51	2.54
	150	1351.69	18،12،4	1351.69	6.06	1351.69	2.26
	200	1501.62	12 ،20	1501.62	4.65	1501.62	1.26
	250	1601.62	12 ،20	1601.62	4.24	1601.62	1.37
0.6	100	1333.56	12،4،2	1333.56	6.35	1333.56	3.55
	150	1483.56	12،4،2	1483.56	6.47	1483.56	1.36
	200	1601.2	12 ،20	1601.2	4.5	1601.2	1.42
	250	1701.2	12 ،20	1701.2	4.81	1701.2	1.65
0.8	100	1458.83	12،4،2	1458.83	6.32	1458.83	3.6
	150	1594.08	12 ،20	1594.08	6.07	1594.08	1.34
	200	1690.57	5	1690.57	2.64	1690.57	1.05
	250	1740.57	5	1740.57	2.1	1740.57	1.16
1	**100**	**1556.63**	**20،8،4**	**1559.19**	**5.5**	**1556.63**	**3.2**
	150	1640.57	5	1640.57	2.26	1640.57	1.13
	200	1690.57	5	1690.57	2.12	1690.57	1.19
	250	1740.57	5	1740.57	2.13	1740.57	1.12
Average runtime				6.64		1.98	

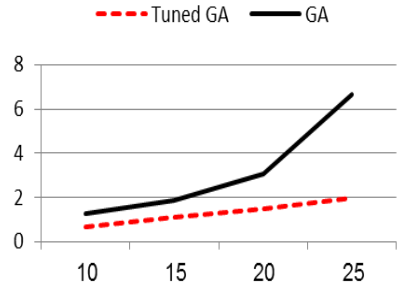

Fig. 5. Comparing average runtimes of GA and Tuned-GA in solving different instances of the CAB dataset

5. Conclusions

In this paper a new approach for robust parameter tuning of heuristics and metaheuristics algorithms is proposed, which is based on a combination of Taguchi design of experiments, signal to noise ratio, and VIKOR methods. The mentioned approach not only considers the solution quality or the number of fitness function evaluations, but also tries to minimize the runtimes of algorithms as a secondary goal. The performance of the developed method is evaluated by solving two combinatorial problems: *n*-queens and uncapacitated hub location problem. Extensive

experimental results showed that by implementing the developed parameter tuning approach for the SA proposed in [16], the average number of iterations and average runtimes of the algorithm were improved 12 and 10.2 times respectively, compared to the un-tuned SA in solving *n*-queens problem.

Also, the implementation of the developed parameter tuning method on the GA proposed in [18] was evaluated by solving a number of benchmark problems taken from the CAB dataset on the Uncapacitated Hub Location Problem. The results demonstrated that the quality of certain solutions was improved in the Tuned-GA, while the average runtimes was 2.5 times faster compared to the un-tuned GA. So, the proposed method can be applied as a power tool for parameter tuning of heuristic algorithms.

References

[1] Birattari, M., "*Tuning Metaheuristics: A Machine Learning Perspective"*, Studies in Computational Intelligence, Volume 197, Springer-Verlag Berlin Heidelberg, 2009.

[2] Barr, R. S., B.L. Golden, J.P., Kelly, M.G.C., Resende Steart, W.R., *Designing and Reporting on Computational Experiments with Heuristic Methods*, Journal of Heuristics, Vol. 1, 1995.

[3] Xu, J., Chiu, S.Y., Glover, F., *Fine-tuning a Tabu Search Algorithm with Statistical Tests*, Int. Trans. Op. Res., Vol. 5, No. 3, 1998, pp. 233-244.

[4] Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., *"Choosing Multiple Parameters for Support Vector Machines*," Journal of Machine Learning Research, Vol. 46, No. 1–3, 2002, pp. 131–159.

[5] R.-E. Fan, P.-H. Chen and C.-J. Lin, *"Working Set Selection using the Second Order Information for Training SVM,"* international Journal of Machine Learning Research, Vol. 6, 2005, pp. 1889–1918.

[6] Imbault, F., Lebart, K., "*A Stochastic Optimization Approach for Parameter Tuning of Support Vector Machines,"* In Proceedings of the 17th International Conference on Pattern Recognition (IPCR), Vol. 4, 2004, pp. 597–600.

[7] Figlali, N., Özkale, C., Engin, O., Figlali, A., *"Investigation of Ant System Parameter Interactions by using Design of Experiments for Job-Shop Scheduling Problems*," Computers & Industrial Engineering, Vol. 56, 2009, pp. 538–559.

[8] Tewolde, G., Hanna, D., Haskell, R., *"Enhancing Performance of PSO with Automatic Parameter Tuning Technique*," 2009, pp. 67–73.

[9] Cooren, Y., Clerc, M., Siarry, P., "*Performance Evaluation of Tribes, an Adaptive Particle Swarm Optimization Algorithm*," Swarm Intelligence, Vol. 3, No. 2, 2009, pp. 149–178.

[10] Wong, K., Komarudin, "*Parameter Tuning for Ant Colony Optimization: A Review*," 2008, pp. 542–545.

[11] Talbi, E.G., *Metaheuristics: from Design to Implementation*, Wiley, 2009.

[12] Box, G. Hunter, J.S., Hunter., W.G., *Statistics for Experimenters: Design*, Innovation, and Discovery. Wiley, 2005.

[13] Montgomery D.C., "*Design and Analysis of Experiments (Sixth Edition)",*John Wiley & Sons, Inc. ISBN 0-471-48735-X, 2005.

[14] Schaffer, J.D., Caruana, R.A., Eshelman, L., Das. R., *A Sudy of Control Parameters Affecting Online Performance of GA for Function Optimization. In J. D. Schaffer*, editor, 3rd International Conference on Genetic Algorithms, Morgan Kaufman, San Mateo, CA, 1989, pp. 51–60.

[15] Hwang, C.L., Yoon., K., *Multiple Attribute Decision Making Method and Applications*, New York :Springer-Verlag, 1981.

[16] Martinjak, I., Golub, M., *"Comparison of Heuristic Algorithms for the N-Queen Problem*," 29th International Conference on Information Technology Interfaces, Cavata, Croatia: IEEE, 2007, pp. 759-764.

[17] Masehian, E., Mohabati-Kalejahi, N., Akbaripour, H., "*Basic and Hybrid Imperialist Competitive Algorithms for Solving the N-Queens Problem*", 4th International Conference on Evolutionary Computation Theory and Applications, 2012, Barcelona, Spain.

[18] Topcuoglu, H., Corut, F., Ermis, M., Yilmaz, G., "*Solving the Uncapacitated Hub Location Problem Using Genetic Algorithms*", Computers & Operations Research, Vol. 32, 2005, pp. 967–984.