A New Multi-Objective Ant Colony Optimisation Algorithm for Solving the Quadratic Assignment Problem

I.D.I.D. Ariyasingha[1*] and T.G.I. Fernando[2]

[1]*Department of Computer Science, Faculty of Natural Sciences, Open University of Sri Lanka, Sri Lanka*
[2]*Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka*

Date Received: 20-12-2018 Date Accepted: 22-04-2019

Abstract

The multi-objective quadratic assignment problem (mQAP) is an NP-hard combinatorial optimisation problem. Real world problems are concerned with multi-objective problems which optimise more objective functions simultaneously. Moreover, QAP models many real-world optimisation problems, such as network design problems, communication problems, layout problems, etc. One of its major applications is the facility location, which is to find an assignment of all facilities to all locations in the way their total is minimised. The multi-objective QAP considers multiple types of flows between two facilities. Over the last few decades several meta-heuristic algorithms have been proposed to solve the multi-objective QAP, such as genetic algorithms, Tabu search, simulated annealing, and ant colony optimisation. This paper presents a new ant colony optimisation algorithm for solving multiple objective optimisation problems, and it is named as the random weight-based ant colony optimisation algorithm (RWACO). The proposed algorithm is applied to the bi-objective quadratic assignment problem and evaluates the performance by comparing with some recently developed multi-objective ant colony optimisation algorithms. The experimental results have shown that the proposed algorithm performs better than the other multi-objective ACO algorithms considered in this study.

Keywords: ACO, multi-objective problem, QAP, travelling salesman problem

1. Introduction

Real ants in a colony find the shortest path between their nest and the food source using the pheromone trails lying on the ground. Ant colony optimisation (ACO) algorithms (Dorigo et al., 1996) inspire this foraging behaviour of real ant colonies for finding solutions for optimisation problems.

This paper proposes a new ant colony optimisation algorithm in the context of multiple objectives for solving the bi-objective quadratic assignment problem, and it is named as random weight-based ant colony optimisation algorithm (RWACO).The performance of the new algorithm is evaluated by comparing it with recently developed multi-objective ant colony optimisation algorithms. The paper is structured as follows. Section 2 presents preliminaries of the study, such as multi-objective optimisation problem, the quadratic assignment problem, the ant colony optimisation algorithm, and multi-objective ant colony optimisation algorithms. The new multi-objective ant colony optimization algorithm, RWACO, is presented in Section 3. Methodology of the study is presented in Section 4 while presenting conclusions in Section 5.

[*] *Correspondence: iresha@ou.ac.lk*
Tel: +94 773917088

2. Preliminaries

2.1 Quadratic assignment problem

Koopmans and Beckmann (1957) introduced the scalar Quadratic Assignment Problem (QAP) to model a plant-location problem. QAP is one of the hardest combinatorial optimisation problems (Stutzle, 1998) and also an NP-hard problem which can be defined as the problem of assigning a set of facilities to a set of locations. The goal is to place facilities on locations such that the sum of the product between distances and flows is minimal. The QAP of size n can be formulated as given in the following equation:

$$\underset{\pi \epsilon \Omega}{\mathit{Minimise}} \quad C(\pi)=\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\, b_{\pi_i \pi_j} \tag{1}$$

where b_{ij} is the distance between the location i and location j;.a_{ij} is the flow between the facility i to facility j; π_i is the location of facility i in the current solution $\pi_i \epsilon \Omega$, Ω is the set of all permutations.

Knowles and Corne (2003) introduced multi-objective quadratic assignment problem (mQAP) by considering several flow matrices and the same distance matrix and it can be defined as follows:

$$\underset{\pi \epsilon \Omega}{\mathit{Minimise}} \quad C(\pi) = \left\{C^{1}(\pi), C^{2}(\pi), \ldots, C^{m}(\pi)\right\} \tag{2}$$

$$C^{k}(\pi)=\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\, f^{k}_{\pi_i \pi_j}, \quad 1 \le k \le m \tag{3}$$

where n is the number of facilities/locations and $C(\pi)$ is the vector of m objective functions. d_{ij} is the distance between location L_i and location L_j. f^{k}_{ij} is the k-th flow between the facilities π_i and π_j .

2.1 Multi-objective optimisation problem

A general multi-objective optimisation problem (MOOP) (Deb, 2001) can be formulated according to the following equation, which includes a set of m objective functions, n decision variables with k restrictions.

$$\left.\begin{array}{l} y=f(x)=\left[f_1(x), f_2(x), \ldots, f_m(x)\right] \\ e(x)=[e_1(x), e_2(x), \ldots, e_k(x) \ge 0, \\ x=(x_1, x_2, \ldots, x_n) \in X \text{ is the decision vector,} \\ y=\left(y_1, y_2, \ldots, y_m\right) \in Y \text{ is the objective vector.} \end{array}\right\} \tag{4}$$

where X denotes the decision space of n decision variables and the objective space with m set of objective functions is denoted by Y.

2.2 Ant colony optimisation

Real ant colonies perform several different patterns, such as foraging, cooperative transport, division of labour, and brood sorting. They indirectly communicate via *stigmergy* which is used to coordinate their activities because ant species are totally blind.

Foraging ant in the colony deposits a chemical called *pheromone* on the ground when travelling and other ants probabilistically prefer to follow the same path. Pheromones deposited on the ground are evaporated over time. Thus, real ants in the colony are able to find the shortest path between their nest and the food source.

Ant colony optimisation (ACO) algorithm is inspired by the behaviour of these real ant colonies (Dorigo et al., 1996; Dorigo and Stutzle, 2004). ACO can be applied to solve combinatorial optimisation problems. Ants in the artificial colony traverse from one node to another on a graph to find a solution for a problem. The probability of choosing a node as the next node depends on the artificial pheromone trail and the heuristic information. After completing a tour by an ant the pheromone trail is evaporated and applies a new pheromone trail on the path. If the completed path by the ant is a good solution, then the pheromone trail on that path will be high, and vice-versa.

2.3 Multi-objective ant colony optimisation algorithms

During the last few years several MOACO algorithms were developed. The following MOACO algorithms were considered for evaluating the performance of the new approach.

Rabanimotlagh (2011) proposed an efficient ant colony optimisation algorithm for multi-objective flow shop scheduling problem named “ACOMOFS.” It minimises two objectives, make span and total flow time simultaneously in flow shop scheduling problem. It uses one ant colony with several heuristic matrices and only one pheromone trail. ACOMOFS compares with the hybrid algorithms for multi-criteria (HAMC) (Yagmahan and Yenisey, 2010) and the multi-objective ant colony system algorithm (MOACSA) (Ho and Chang, 1991) to verify its performance. The experimental results have shown that the ACOMOFS algorithm performs better than the other HAMC and MOACSA algorithms.

Angus (2007) proposed a crowding population-based ant colony optimisation named “CPACO” which is based on a crowding population replacement scheme and the population based ant colony optimisation algorithm (PACO) (Guntsch and Middendorf, 2003). Performance of the CPACO algorithm was compared with the PACO algorithm by applying both algorithms to the travelling salesman problem. The results obtained have shown that the CPACO algorithm achieves better performance and also lower computational complexity than the PACO algorithm.

Baran and Schaerer (2003) introduced a new multi-objective ant colony system (MACS) algorithm for vehicle routing problem with time windows which extended the MACS-VRPTW (Gambardella et al., 1999) algorithm in the multi-objective context. MACS algorithm uses a single colony to obtain a set of solutions considering two objectives simultaneously. Experimental results have shown that the MACS algorithm is better than the MACS-VRPTW algorithm in the multi-objective context.

3. The proposed ant colony optimisation algorithm (RWACO Algorithm)

This paper proposes a new ant colony optimisation algorithm, random weight-based ant colony optimisation (RWACO) algorithm, for solving quadratic assignment problem with multiple objective functions. It uses only one colony, and is based on the ant colony system (ACS) algorithm (Dorigo and Gambardella, 1997). Furthermore, it uses a single pheromone trail and several heuristic information. This new ACO algorithm (RWACO) was experimented in our previous research (Ariyasingha and Fernando, 2015, 2017), and has been applied for solving the travelling salesman problem and the job shop scheduling problem respectively. The RWACO algorithm obtained best results over the other ACO algorithms used in the experimentation. In order to adapt the RWACO algorithm into QAP, modifications have to be done according to the following equations.

3.1 State transition rule

First, generate a random number $q \in [0,1]$, before applying the state transition rule. If the generated value $q \leq q_0$, each ant k assigns facility j to location i according to the following exploitation step.

$$j = \arg\max_{j \in N_i^k} \{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}\} \qquad (5)$$

If $q > q_0$, exploration is performed to assign facility j as follows:

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{u \in N_i^k} [\tau_{iu}]^{\alpha} [\eta_{iu}]^{\beta}} \quad if\, j \in N_i^k \qquad (6)$$

where q_0 is a parameter. Each coupling *(i,j)* has the pheromone information τ_{ij} and the heuristic information η_{ij}.α and β are parameters. N_i^k is the available facilities which are not assigned to locations. The heuristic information of the coupling *(i,j)* is calculated as follows:

$$\eta_{ij} = \frac{1}{\sum_{h=1}^{H} w_h d_{ij}^{h}} \qquad (7)$$

where H is the number of objectives and d_{ij}^{h} is the cost associated to the coupling *(i,j)* of each objective h. Randomly generated weighting coefficient for each objective h is denoted by w_h. Sum of all these weighting coefficients *(w)* set equal to one as follows:

$$\sum_{h=1}^{H} w_h = 1 \qquad (8)$$

3.2 Local pheromone updating

At each construction step, the local pheromone updating rule is performed once the next facility j is selected according to the following equation. First, pheromone trail on the selected coupling *(i,j)* is decreased.

$$\tau_{ij} = (1-\rho') \, \tau_{ij} + \rho' \, \tau_0 \qquad (9)$$

where $\rho' \in [0,1]$=parameter; τ_0 =the initial pheromone information.

3.3 Global pheromone updating

The global updating rule is applied after all ants in the colony have constructed their permutation in each iteration. First, pheromone values of all coupling are evaporated as follows.

$$\tau_{ij} = (1-\rho)\tau_{ij} \qquad (10)$$

Then, pheromone values of all the non-dominated solutions are updated as given in the following equation:

$$\tau_{ij} = \tau_{ij} + \frac{\rho}{\sum_{h=1}^{H} w_h F_h} \qquad (11)$$

where $\rho \in [0,1]$ is a parameter and H is the number of objectives and equal weighting coefficients (w_h) are used for each objective h. F_h represents the objective function value of the non-dominated solution.

4. Performance assessment methodology

Garcia-Martinez et al. (2007) analysed the performance of MOACO algorithms for the bi-objective travelling salesman problem and they concluded that the MACS algorithm performs best among the other algorithms.

Further, our previous study (Ariyasingha and Fernando, 2015) has shown that the MACS and the ACOMOFS algorithms obtain better performance while the CPACO algorithm achieves good solutions in multi-objective travelling salesman problem.

Thus, in order to evaluate the performance of the newly proposed MOACO algorithm, three MOACO algorithms, the MACS, ACOMOFS and CPACO, were considered. These algorithms are described in Section 2.4. All of these algorithms were applied to the bi-objective quadratic assignment problem and six benchmark instances of size 10: KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-3uni, KC10-2fl-1rl, KC10-2fl-2rl, KC10-2fl-3rl (Knowles and Corne, 2003) were considered for the experimentation. Most effective parameters on the performance of MOACO algorithms were identified and the best values of the parameters were applied for the MOACO algorithms. The parameters considered in this study are given in Table 1. Also, each algorithm considered 20 ants, 100 iterations and 10 runs. All algorithms were implemented in the same computer using ‘CodeBlocks 13.12’ under ‘Ubuntu 14.04’ environment running on an Intel Core i3 CPU at 2.40 GHz, with 4 GB memory.

Four performance indicators in the literature were considered to evaluate the performance of the MOACO algorithms in this study. They are, overall non-dominated vector generation (ONVG), overall true non-dominated vector generation (OTNVG), overall true non-dominated vector generation ratio (OTNVGR) (Gottlieb and Raidl, 2006), and coverage performance indicator (C performance indicator) (Deb, 2001). Some of these performance indicators require the pseudo-optimal Pareto front for calculations which is an approximation to the true Pareto-optimal front and it can be obtained by the procedure presented by Pinto and Baran (2005).

Table 1: Parameter values for each MOACO algorithm.

MOACO Algorithm	τ_0	α	β	ρ	ρ'	q_0
ACOMOFS, RWACO	0.000125	0.1	0.5	0.2	0.05	0.98
CPACO, MACS	5.5498E-18	1	2	0.2	0.05	0.98

5. Analysis of results

Four performance indicators presented in Section 4 and also the visual representation methods were considered to evaluate the results of the MOACO algorithms.

5.1 Visual representation

Pseudo-optimal Pareto fronts of each MOACO algorithm were obtained using the non-dominated solutions returned by the 10 runs of each MOACO algorithm. Scatter-plot matrix method have been considered to visually represent these pseudo- optimal Pareto fronts. Therefore, the bi-objective QAP instances: KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-3uni, KC10-2fl-1rl, KC10-2fl-2rl, KC10-2fl-3rl are presented in Figures. 1-6, respectively, using the scatter-plot matrix method. According to the following figures, it has been observed that the RWACO algorithm obtains better non-dominated solutions in KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-3uni and KC10-2fl-3rl QAP instances. Also, the MACS and ACOMOFS algorithms return poor non-dominated solutions in KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-2rl and KC10-2fl-3rl QAP instances. When all QAP instances have been considered, it is shown that the CPACO algorithm returns better solutions same as the RWACO algorithm for most of the instances. However, the RWACO algorithm outperforms all the other MOACO algorithms in KC10-2fl-2uni QAP instance.

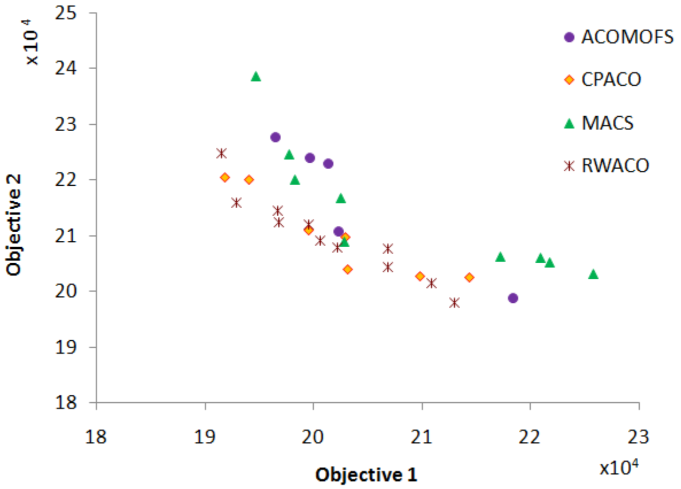

Figure 1. Non-dominated solutions of each MOACO algorithm for KC10-2fl-1uni QAP instance.

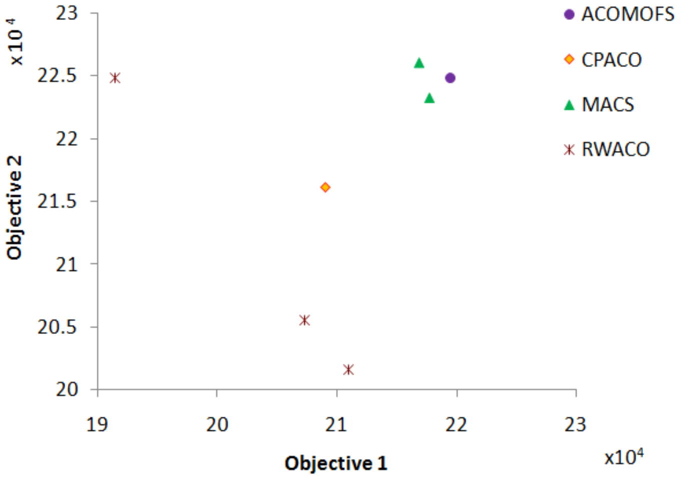

Figure 2. Non-dominated solutions of each MOACO algorithm for KC10-2fl-2uni QAP instance.

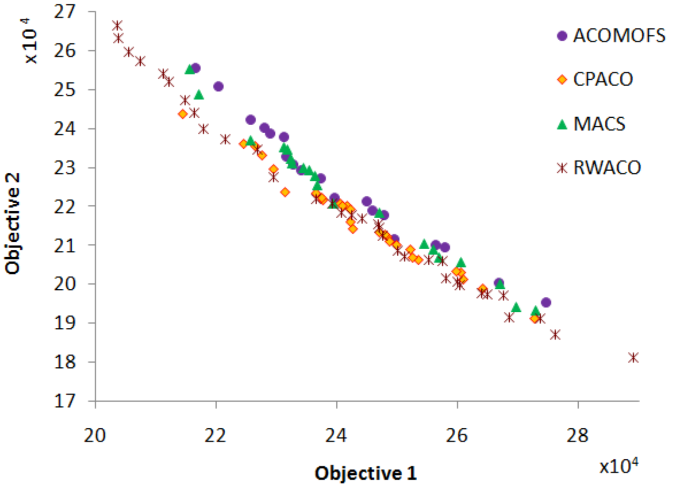

Figure 3. Non-dominated solutions of each MOACO algorithm for KC10-2fl-3uni QAP instance.

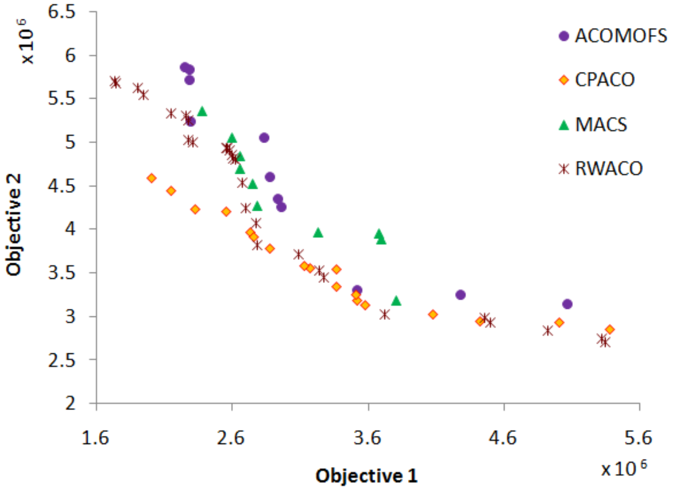

Figure 1. Non-dominated solutions of each MOACO algorithm for KC10-2fl-1rl QAP instance.

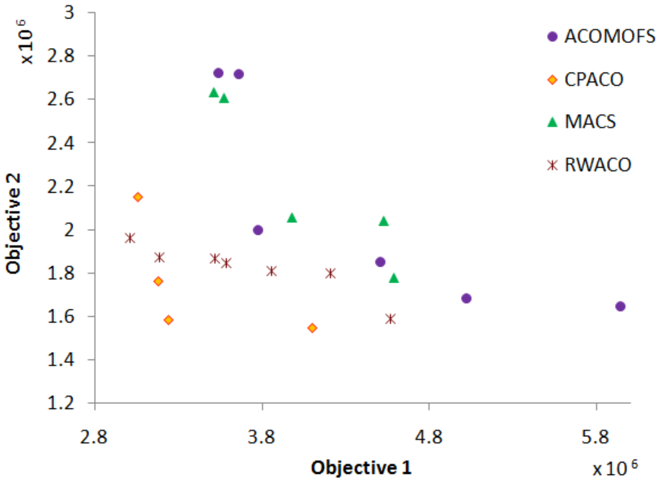

Figure 2. Non-dominated solutions of each MOACO algorithm for KC10-2fl-2rl QAP instance.

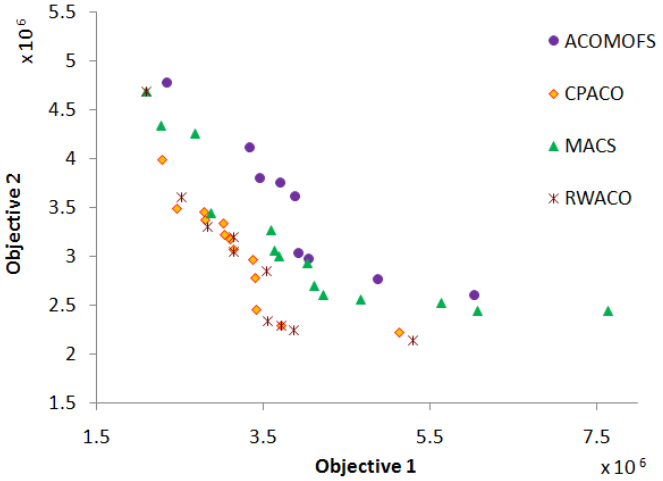

Figure 3. Non-dominated solutions of each MOACO algorithm for KC10-2fl-3rl QAP instance.

5.2 Analysis with OTNVGR

Pseudo-optimal Pareto front y_{apr} for each QAP instance can be obtained by the procedure presented by Pinto and Baran (2005). Table 2 presents the number of non-dominated solutions in $|y_{apr}|$ which was found experimentally for each QAP benchmark instance.

Tables 3-5 present the results with respect to the ONVG, OTNVG and OTNVGR performance indicators (Gottlieb and Raidl, 2006), and also the CPU time in seconds for each MOACO algorithm. The higher values of OTNVGR indicate that the obtained results are better.

When all QAP instances have been considered, the ACOMOFS and MACS algorithms are outperformed by the other two algorithms as they return very poor solutions in OTNVGR. The RWACO algorithm is the best algorithm and outperforms all the other MOACO algorithms for KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-3uni and KC10-2fl-1rl instances as it obtains higher values in OTNVGR. Moreover, the CPACO algorithm outperforms other algorithms for KC10-2fl-2rl and KC10-2fl-3rl instances as it obtains higher values in OTNVGR. However, the RWACO algorithm returns good results in OTNVGR for these two instances.

According to the computational time (CPU time) in seconds for each MOACO algorithm, it is shown that the CPACO algorithm obtains solutions in higher computational time while all the other algorithms obtain solutions in lower computational time.

Table 2: Number of non-dominated solutions of $|y_{apr}|$ for benchmark instances.

QAP Instance	KC10-2fl-1uni	KC10-2fl-2uni	KC10-2fl-3uni	KC10-2fl-1rl	KC10-2fl-2rl	KC10-2fl-3rl
$\lvert y_{apr} \rvert$	12	11	50	33	4	21

Table 3: Comparison of solutions with ONVG, OTNVG and OTNVGR of KC10-2fl-1uni and KC10-2fl-2uni QAP instances.

MOACO algorithm	KC10-2fl-1uni				KC10-2fl-2uni			
	ONVG	OTNVG	OTNVGR (%)	CPU time (second)	ONVG	OTNVG	OTNVGR (%)	CPU time (second)
ACOMOFS	5	0	0.00	192	1	0	0.00	190
CPACO	7	4	33.33	317	1	0	0.00	315
MACS	9	0	0.00	197	2	0	0.00	195
RWACO	11	8	66.67	202	11	11	100	199

Table 4: Comparison of solutions with ONVG, OTNVG, and OTNVGR of KC10-2fl-3uni and KC10-2fl-1rl QAP instances.

MOACO algorithm	KC10-2fl-3uni				KC10-2fl-1rl			
	ONVG	OTNVG	OTNVGR (%)	CPU time (second)	ONVG	OTNVG	OTNVGR (%)	CPU time (second)
ACOMOFS	27	0	0.00	192	11	0	0.00	191
CPACO	32	16	32.00	307	18	14	42.42	304
MACS	44	0	0.00	196	12	0	0.00	198
RWACO	49	34	68.00	204	36	19	57.58	201

Table 5: Comparison of solutions with ONVG, OTNVG, and OTNVGR of KC10-2fl-2rl and KC10-2fl-3rl QAP instances.

MOACO algorithm	KC10-2fl-2rl				KC10-2fl-3rl			
	ONVG	OTNVG	OTNVGR (%)	CPU time (second)	ONVG	OTNVG	OTNVGR (%)	CPU time (second)
ACOMOFS	6	0	0.00	190	9	0	0.00	191
CPACO	4	3	75.00	311	13	12	57.14	307
MACS	5	0	0.00	198	14	2	9.53	197
RWACO	7	1	25.00	200	10	7	33.33	199

5.3 Analysis with C performance indicator

The results of the C performance indicator (Deb, 2001) obtained by all the MOACO algorithms are presented using the set of box-plots as given in Figure 7. From left to right six box-plots are presented in each box for: KC10-2fl-1uni, KC10-2fl-2uni, KC10-2fl-3uni, KC10-2fl-1rl, KC10-2fl-2rl and KC10-2fl-3rl, respectively. Row and column of the figure refer to the two algorithms *A* and *B*, respectively. The C performance indicator measures the proportion of solutions of algorithm *B* which are dominated by solutions of algorithm A. Therefore, $C(A, B)=1$ means that all solutions of *B* are dominated by solutions of A. Also, $C(A, B)=0$ presents that solutions of *B* are not dominated by solutions of A. The middle line of a box presents the median of $C(A, B)$. Also, the bottom scale presents 0 and the top scale presents 1 of each box.

According to the C performance indicator values presented in Figure 7, the RWACO algorithm performs better than the other MOACO algorithms as its C indicator values are almost zero in most of the time. However, the CPACO algorithm performs better than the RWACO algorithm only in the KC10-2fl-2rl instance as the C indicator value of the CPACO algorithm is close to zero.

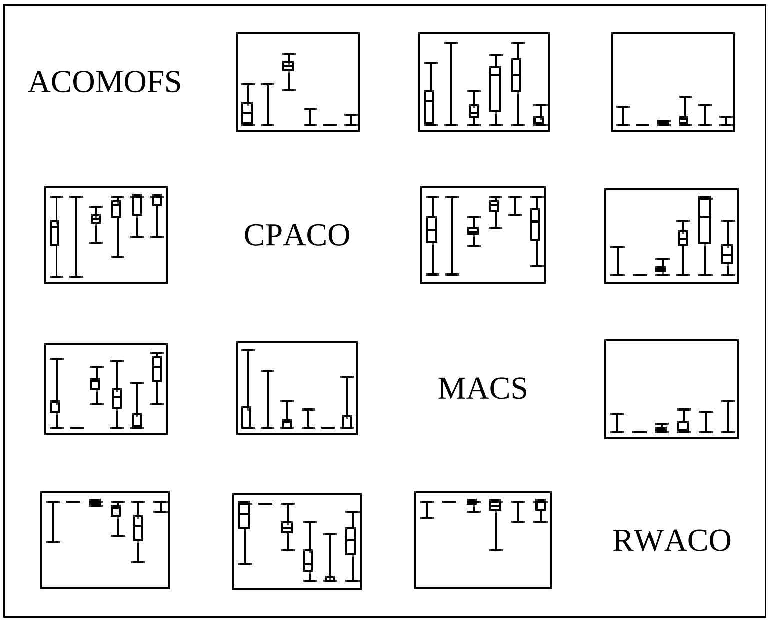

Figure4. Box-plots of the results returns by C performance indicator for MOACO algorithms.

6. Conclusion

In this paper, a new multi-objective ant colony optimisation algorithm named the random weight-based ant colony optimisation (RWACO) has been proposed for solving QAP, and it uses randomly generated weights. The RWACO algorithm has been compared with the recently introduced three MOACO algorithms and applied them for the quadratic assignment problem, in order to evaluate the performance of the proposed algorithm. According to the visual representation, the RWACO algorithm returns better non-dominated solutions for four QAP instances. Also, the RWACO algorithm obtains better solutions for OTNVGR in low computational time. When we consider the results of the C performance indicator, it returns better performance as it obtains C indicator values close to zero in most of the time. Therefore, it can be concluded that RWACO algorithm obtains better performance than the other MOACO algorithms considered.

References

Angus, D., 2007. Crowding population-based ant colony optimization for the multi-objective travelling salesman problem, *Multi-criteria Decision Making (MCDM 2007)*. Honolulu, Hawaii, USA: IEEE Press, 333-340.

Ariyasingha, I.D.I.D., Fernando, T.G.I., 2015. Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem. *Swarm and Evolutionary Computation*, 23:11-26.

Ariyasingha, I.D.I.D., Fernando, T.G.I., 2017. Random weight-based ant colony optimization algorithm for the multi-objective optimisation problems. *International Journal of Swarm Intelligence*, 3:77-100.

Baran, B., Schaerer, M., 2003. A multi objective ant colony system for vehicle routing problem with time windows, *International Conference on Applied Informatics (IASTED)*. Insbruck, Austria, February 10-13.97-102.

Deb, K. 2001. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, New York.

Dorigo, M., Gambardella L.M., 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53-66.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:29-41.

Dorigo, M., Stutzle, T., 2004. *Ant Colony Optimization*, MIT Press, Cambridge, MA.

Gambardella, L., Taillard, E., Agazzi, G., 1999. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. In *D. Corne, M. Dorigo, F. Glover (Eds.)*, New Ideas in Optimization, McGraw-Hill. 73-76.

Garcia-Martinez, C., Cordon, O., Herrera, F., 2007. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria tsp. *European Journal of Operational Research*, 180:116-148.

Gottlieb, J., Raidl, G.R., 2006. *Evolutionary Computation in Combinatorial Optimization*. *6th European Conference, EvoCOP 2006*. Budapest, Hungary, April 10-12. Springer: LNCS, volume 3906.

Guntsch, M., Middendorf, M., 2003. Solving multi-criteria optimization problems with population-based ACO, *Evolutionary Multi-Criterion Optimization (EMO 2003)*. Faro, Portugal: Springer, 8-11 April. Springer LNCS volume 2632, 464-478.

Ho, J.C., Chang, Y.L., 1991. A new heuristic for the n-job, machine flow shop problem. *European Journal of Operational Research*, 52:194-202.

Knowles, J., Corne D., 2003. Instance generators and test suits for the multi objective quadratic assignment problem, *Evolutionary Multi-Criterion Optimization (EMO 2003)*. Faro, Portugal, 8-11 April: Springer LNCS volume 2632, 295-310.

Koopmans, T.C., Beckmann, M., 1957. Assignment problems and the location of economic activities. *Econometrica*, 25:53-76.

Pinto, D., Baran, B., 2005. 'Solving multi-objective multicast routing problem with a new ant colony optimization approach', *IFIP/ACM Latin American Conference on Networking (LANC'05)*. Cali, Colombia. 11-19.

Rabanimotlagh, A., 2011. An efficient ant colony optimization algorithm for multi-objective flow shop scheduling problem. *World Academy of Science, Engineering and Technology*, 75:127-133.

Stutzle, T., 1998. *Local Search Algorithms for Combinatorial Problems-Analysis, Improvements and New Applications*. PhD thesis. Darmstadt University of Technology.

Yagmahan, B., Yenisey, M.M., 2010. A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*, 37:1361-1368.