Hybrid Algorithm for Solving the Quadratic Assignment Problem

Mohammed Essaid Riffi, Fatima Sayoti*

LAROSERI Laboratory, Dept. of Computer Science, Faculty of Sciences, University of Chouaib Doukkali, El Jadida (Morocco)

Received 1 June 2017 | Accepted 27 September 2017 | Published 20 October 2017

Abstract

The Quadratic Assignment Problem (QAP) is a combinatorial optimization problem; it belongs to the class of NP-hard problems. This problem is applied in various fields such as hospital layout, scheduling parallel production lines and analyzing chemical reactions for organic compounds. In this paper we propose an application of Golden Ball algorithm mixed with Simulated Annealing (GBSA) to solve QAP. This algorithm is based on different concepts of football. The simulated annealing search can be blocked in a local optimum due to the unacceptable movements; our proposed strategy guides the simulated annealing search to escape from the local optima and to explore in an efficient way the search space. To validate the proposed approach, numerous simulations were conducted on 64 instances of QAPLIB to compare GBSA with existing algorithms in the literature of QAP. The obtained numerical results show that the GBSA produces optimal solutions in reasonable time; it has the better computational time. This work demonstrates that our proposed adaptation is effective in solving the quadratic assignment problem.

Keywords

Combinatorial Optimization, Golden Ball Algorithm, Simulated Annealing, Quadratic Assignment Problem.

DOI: 10.9781/ijimai.2017.10.003

I. Introduction

The quadratic assignment problem (QAP) is one of the known classical combinatorial optimization problems, in 1976 Sahni and Gonzalez [1] proved that the QAP belongs to the class of NP-hard problems [1]. It was introduced for the first time by Koopmans and Beckmann in 1957 [2]; its purpose is to assign n facilities to n fixed locations with a given flow matrix of facilities and distance matrix of locations in order to minimize the total assignment cost. This problem is applied in various fields such as hospital layout [3], scheduling parallel production lines [4] and analyzing chemical reactions for organic compounds [5].

Many recent hybrid approaches have improved performance in solving QAP such as genetic algorithm hybridized with tabu search method [6], ant colony optimization mixed with local search method [7] and ant colony optimization combined with genetic algorithm and local search method [8]. Recently the hybrid algorithms are much proposed and used by many researchers to find optimal or near optimal solutions for the QAP.

In this paper we propose a new competitive approach when compared with other existing methods in the literature. The golden ball algorithm mixed with simulated annealing (GBSA) is considered here as a hybrid metaheuristic to apply in the quadratic assignment problem.

This work presents an efficient adaptation of GBSA algorithm to the quadratic assignment problem (QAP). This algorithm is based on the concept of soccer; it guides the search by simulated annealing [9] to escape from the local optima. The suggested technique has never been proposed or tested with QAP. In this research we use some small, medium and large test problems for comparing our approach to other recent methods from literature. Our approach is able to explore effectively the search space; it reaches the known optimal solutions in less time.

The rest of this paper is structured as follows: In section I, Introduction. In section II, Quadratic assignment problem formulation. In section III, Methods. In section IV, Results and discussion. In section V, Conclusion.

* Corresponding author.

E-mail address: fatimasayoti@gmail.com

II. Quadratic Assignment Problem

The QAP [1] can be defined as a problem of assigning n facilities to n locations, with given flows between the facilities and given distances between the locations (Fig.1).

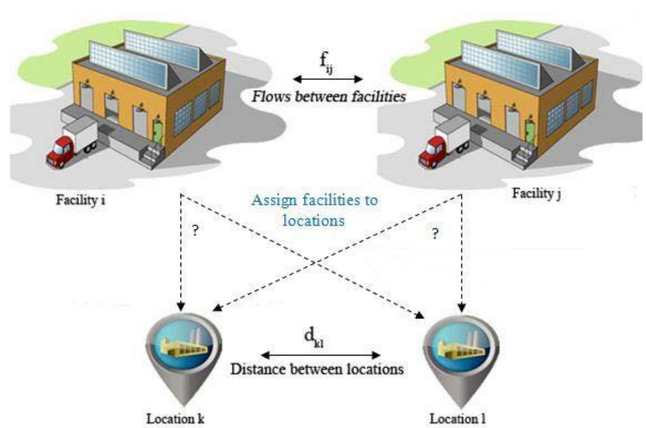

Fig. 1. Quadratic Assignment Problem.

The purpose is to assign the facilities to the locations in such a way that the total cost is minimized. Each facility must be placed just at one location.

We consider two n × n matrices, the flow matrix $F=f_{ij}$ and the distance matrix $D=d_{kl}$. The QAP formulation is given as follows (1):

$$\min_{\pi \epsilon S_n} Cost = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} f_{ij}\, d_{\pi(i)\, \pi(j)} \quad (1)$$

S_n is the set of all permutation of n elements {1, 2, …, n}.

π(i) and π(j) are respectively locations of facilities *i* and *j*, we suppose that *π(i)=k* and *π(j)=l.*

$f_{ij}\, d_{\pi(i)\, \pi(j)}$ is the cost of assigning facility *i* in location *k* and facility *j* in location *l.*

The objective function (Cost) must be minimized.

Several algorithms are usually used to solve the quadratic assignment problem:

- Exact algorithms such as branch and bound algorithm [10] and branch and cut algorithm [11].
- Metaheuristics such as genetic algorithm [12],[13],[14], tabu search method [15],[16],[17], simulated annealing algorithm [9], ant colony optimization [18] and particle swarm optimization [19].

In recent year, metaheuristic algorithms are used in solving the QAP more than the exact algorithms which are unable to solve the hard instances of QAP in a reasonable time. Many researchers compared between different metaheuristic algorithms for solving the QAP [20], [21].

III. Methods

A. Golden Ball Metaheuristic

The GB technique is a metaheuristic proposed by E.Osaba et al. [22],[23]. It uses different principles of soccer to solve combinatorial optimization problems. The quality of this technique is demonstrated applying it to four combinatorial problems [23]: Asymmetric traveling salesman problem (ATSP) [24], Vehicle Routing Problem with Backhauls (VRPB) [25],[26], n-Queen Problem (NQP) [27], One-Dimensional Bin Packing Problem (BPP) [28].This algorithm is a promising metaheuristic to solve combinatorial optimization problems [23].

In Golden Ball algorithm, groups of solutions are considered as soccer teams which are composed of a fixed number of players, the captain of team plays the rule of the best solution of the group. Each team has a coach who determines the type of training to improve the efficiency of its team. There are two types of training: conventional training and custom training. As shown in Fig. 2, the concept of this method is based on four main phases: initialization phase, training phase, competition phase and transfer phase.

In the initialization phase, we set the value of the number of teams (NT) and the number of players per team (PT). We assign randomly to each team a coach.

In the training phase, all teams must train by following a specific type of training. The conventional training is the daily training of a team. When a team becomes unable to improve its capacities, in this case, it must follow a custom training.

In the competition phase, each team must compete with other team chosen randomly. The winning team receives three points, in the case of equality; both teams receive one point. The accumulated points will be used to order the teams in descending order.

In the transfer phase, we detect three cases of transfer:

Season transfer: during the season, all teams must be sorted in the descending order according to the strength value.

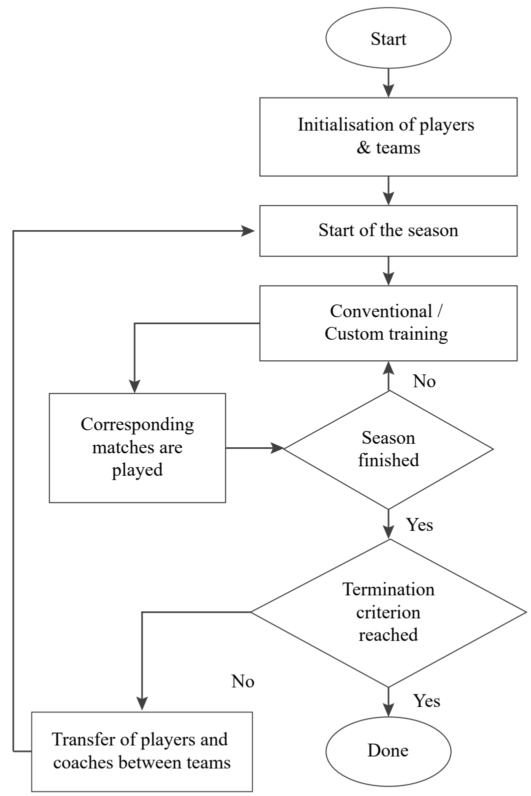

Fig. 2. Flowchart of Golden Ball algorithm.

The strength value is calculated using the following formula (2):

$$\frac{\sum_{i=1}^{PT} qij}{PT} \quad (2)$$

q_{ij} is the quality of player *i* of team *j*

All teams exchange their players in this way: the best player of the first team must be replaced by the worst player of the last team. This worst player will be replaced by this best player.

The best player of the second team must be replaced by the worst player of the penultimate team. This worst player will be replaced by this best player and so forth.

Special transfer: When a player of a given team is unable to improve after a custom training, the team must exchange it with a player of another team chosen randomly.

Cessation of coaches: after having ordered all the teams in descending order according to their accumulated point, the weaker teams must change their conventional training by another randomly selected.

The GB algorithm was tested by E.Osaba et al. with four different combinatorial optimization problems [23]. The same technique was applied on the flow shop scheduling problem [29] and the job shop scheduling problem [30].

B. Simulated Annealing Method

The simulated annealing algorithm [14] is inspired by the physical annealing process which attempt to improve the quality of the solid by using at the beginning a high temperature T_0 at which the solid is in a liquid state. With the slow decrease of the temperature T (cooling phase) the solid regains its solid form (Fig. 3). Metropolis et al. show how to generate a sequence of successive states of the solid. The new state is accepted if the energy produced by this change of state decreases; otherwise, it is accepted with a probability defined by the following equation (3).

$$p = e^{-\Delta E/(c_b \times T)} \quad (3)$$

c_b is Boltzmann constant

ΔE is the energy difference produced by this change of state

T is the temperature of the solid

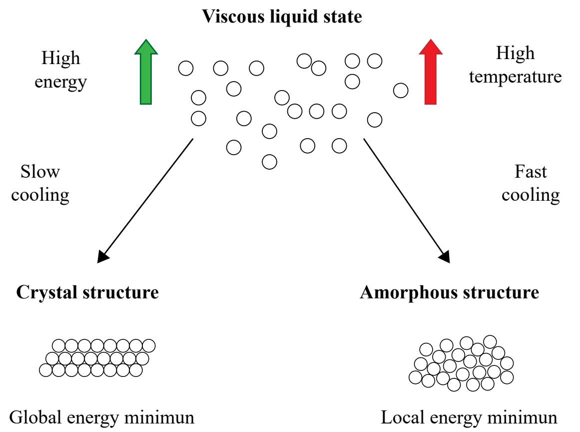

Fig. 3. Evolution of thermodynamic system.

The simulated annealing method [31] is one of the oldest algorithms; it is an iterative metaheuristic very used to solve combinatorial optimization problems in the continuous and discrete case. The strong point of this technique is to escape from the local minima and avoid the cyclic behavior. The performance of simulated annealing algorithm depends on a set of parameters which must be controlled. It means that the correct setting of the parameters produces satisfactory results.

IV. Adaptation of GBSA Algorithm

In the initialization phase we generate randomly the initial population of NT×PT solutions.

Each solution is represented in the following manner (Fig. 4):

Locations →	2	7	4	5	6	3	9	8	1
Facilities →	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Fig. 4. Assign 9 facilities to 9 locations.

In the training phase, we used the following methods as conventional training functions:

2-opt [32], [33]: this iterative method is a local search algorithm, it repeatedly tries to improve the current assignment by exchanging two facilities.

Insertion method [34]: this method inserts a facility chosen randomly between two facilities.

Swapping mechanism [35]: this method swaps two parts selected randomly; the following figure (Fig. 5) explains the concept of this technique.

1	5	3	7	10	6	2	4	8	9
Pos_1		Pos_k					Pos_2		
1	7	10	6	2	4	5	3	8	9

Fig. 5. Illustration of swapping mechanism.

As a custom training function the proposed adaptation used simulated annealing method [14],[31], it is used when the current solution is blocked in the local minima; it helps to accept some movement and escape from the local optimum.

Simulated annealing steps

S1:=Current solution

Generate a new solution based on the current solution. We used the swap of two random locations

S2:=New solution

f(S1):= cost of S1

f(S2):= cost of S2

if (f(S2)<f(S1))

  S1:=S2

Else

  Generate random number r in [0,1)

  Calculate the value of p

$$p = e^{-(f(S2)-f(S1))/(T)}$$

  If(r<p)

  S1:=S2

Decrease the temperature value

Repeat all steps until T= 0.

V. Results and Discussion

The program was run 10 times on different instances of QAPLIB [36]. The GBSA algorithm was implemented in C language and compiled using Microsoft Visual Studio 2008, the program code was executed in computer with Genuine Intel(R) 575 @ 2.00 GHz 2.00 GHz RAM 2,00 Go.

The program uses three parameters: NT (number of groups), PT (number of schedules per group) and T (temperature).

The parameters values in the table below (Table I) produce better results during the algorithm run.

4×5 random solutions are sufficient to obtain good results.

TABLE I. Parameters values

NT	4
PT	5
T	40

At the high temperature, the simulated annealing method becomes unnecessary because proximally 50% of iterations accept decision at the high temperature [37]. In this paper we fixed the high temperature at 40 which is considered a symptom of fever in humans.

Table II represents the following information:

Optimal: Best known Solution

Best: Best permutation

NBest: The number of runs in which the algorithm reaches the best permutation

Worst: The worst permutation

Average: The average cost (= the sum of solutions cost obtained divided by 10)

The relative percentage deviation from the best known solution is calculated as follows (4):

$$\text{RPD} = \frac{\text{Average} - \text{Optimal}}{\text{Optimal}} \times 100\ \% \quad (4)$$

Time: Best time per seconds

TABLE II. Numerical Results of the GBSA Algorithm

Instance	Optimal	Best	Nbest	Worst	Average	%RPD	Time
Bur26a	5426670	5426670	10	5426670	5426670,00	0,00	0
Bur26b	3817852	3817852	10	3817852	3817852,00	0,00	0
Bur26c	5426795	5426795	10	5426795	5426795,00	0,00	1
Bur26d	3821225	3821225	10	3821225	3821225,00	0,00	2
Bur26e	5386879	5386879	10	5386879	5386879,00	0,00	1
Bur26f	3782044	3782044	10	3782044	3782044,00	0,00	1
Bur26g	10117172	10117172	10	10117172	10117172,00	0,00	0
Bur26h	7098658	7098658	10	7098658	7098658,00	0,00	1
Chr12a	9552	9552	10	9552	9552,00	0,00	0
Chr12b	9742	9742	10	9742	9742,00	0,00	0
Chr12c	11156	11156	10	11156	11156,00	0,00	0
Chr15a	9896	9896	10	9896	9896,00	0,00	0
Esc16a	68	68	10	68	68,00	0,00	0
Esc16b	292	292	10	292	292,00	0,00	0
Esc16c	160	160	10	160	160,00	0,00	0
Esc16d	16	16	10	16	16,00	0,00	0
Esc16e	28	28	10	28	28,00	0,00	0
Esc16f	0	0	10	0	0,00	0,00	0
Esc16g	26	26	10	26	26,00	0,00	0
Esc16h	996	996	10	996	996,00	0,00	0
Esc16i	14	14	10	14	14,00	0,00	0
Esc16j	8	8	10	8	8,00	0,00	0
Esc32a	130	136	01	140	139,40	7,23	240
Esc32b	168	168	10	168	168,00	0,00	0
Esc32c	642	642	10	642	642,00	0,00	0
Esc32d	200	200	10	200	200,00	0,00	0
Esc32e	2	2	10	2	2,00	0,00	0
Esc32g	6	6	10	6	6,00	0,00	0
Esc32h	438	438	10	438	438,00	0,00	0
Esc64a	116	116	10	116	116,00	0,00	0
Esc128	64	64	10	64	64,00	0,00	65
Had12	1652	1652	10	1652	1652,00	0,00	0
Had14	2724	2724	10	2724	2724,00	0,00	0
Had16	3720	3720	10	3720	3720,00	0,00	0
Had18	5358	5358	10	5358	5358,00	0,00	1
Had20	6922	6922	10	6922	6922,00	0,00	0
Nug12	578	578	10	578	578,00	0,00	0
Nug14	1014	1014	10	1014	1014,00	0,00	0
Nug15	1150	1150	10	1150	1150,00	0,00	0
Nug16a	1610	1610	10	1610	1610,00	0,00	0
Nug16b	1240	1240	10	1240	1240,00	0,00	0
Nug17	1732	1732	10	1732	1732,00	0,00	0
Nug18	1930	1930	10	1930	1930,00	0,00	0
Nug20	2570	2570	10	2570	2570,00	0,00	0
Rou12	235528	235528	10	235528	235528,00	0,00	0
Rou15	354210	354210	10	354210	354210,00	0,00	0
Rou20	725522	725522	08	725582	725534,00	0,00	1
Scr12	31410	31410	10	31410	31410,00	0,00	0
Scr15	51140	51140	10	51140	51140,00	0,00	0
Scr20	110030	110030	10	110030	110030,00	0,00	3
Sko42	15812	15880	01	16036	15969,00	0,99	240
Sko49	23386	23582	01	23736	23652,40	1,13	240
Tai12a	224416	224416	10	224416	224416,00	0,00	0
Tai15a	388214	388214	10	388214	388214,00	0,00	0
Tai15b	51765268	51765268	10	51765268	51765268,00	0,00	0
Tai17a	491812	491812	10	491812	491812,00	0,00	0
Tai20a	703482	703482	05	713260	706128,90	0,37	32
Tai20b	122455319	122455319	10	122455319	122455319,00	0,00	1
Tai25a	1167256	1181326	01	1193120	1187990,60	1,77	240
Tai30a	1818146	1841180	01	1867650	1858562,80	2,22	240
Tail40a	3139370	3215360	01	3251200	3233951,20	3,01	240
Tail50a	4938796	5084020	01	5143598	5113257,40	3,53	240
Tho30	149936	150578	01	151742	151189,20	0,83	240
Tho40	240516	243362	01	246172	244773,00	1,76	240

The program stops when the optimal solution is reached or when the execution time exceeds 240 seconds. We take two digits after the comma, for the results shown in the two columns: Average and the Relative Percentage Deviation %RPD.

As Table II shows, the proposed algorithm allows to obtain always the optimal solution of 81,25% of the instances tested in a time not exceeding three seconds. The %RPD of 93,75% of the instances does not exceed 2% and this clearly shows that the GBSA algorithm converges well to the optimal solution. According to the values shown in the Table II, when the value of %RPD is equal to 0.00%, this means that the program reaches exactly the optimal solution at least 8 times per 10 tests and in this case the best and the worst solution are often the same.

Abd El-Nasser et al. [38] presented a comparative study between Meta-heuristic algorithms: Genetic Algorithm (GA), Tabu Search (TS), and Simulated annealing (SA) for solving a real-life (QAP) and analyze their performance in terms of both runtime efficiency and solution quality [38].

The Fig. 6 compares the relative percentage deviation of some instances of QALIB for our proposed algorithm GBSA, GA, TS and SA. The result shows that GBSA has more quality than the other algorithms for solving the QAP. We can deduce that our proposed method has really improved SA's effectiveness in solving these instances which we have chosen as an example for our comparative study.

There exist two sets of problems in QAPLIB that represent a challenge for any proposed algorithm. These problems were introduced by Skorin-Kapov [39] and Taillard [40].

We selected 9 instances from Skorin-Kapov and 7 instances from Taillard. For this list of QAPLIB instances, we compared our proposed method with others recent methods such as: Memetic algorithm (BMA) [41], Breakout local search (BLS) [42] and Cooperative parallel tabu search algorithm (CPTS) [43]. The list of instances shown in Table III is a challenge for our algorithm.

We have fixed the maximum execution time of GBSA algorithm at 4 minutes. As the results depict (Table III), the GBSA algorithm needs some improvement to better solve some hard instances of QAP. But in general, the proposed algorithm seems promising to solve the quadratic assignment problem. According to the values of the relative percentage deviation from the best known solution, GBSA algorithm produces results near the global optimum in a reasonable time.

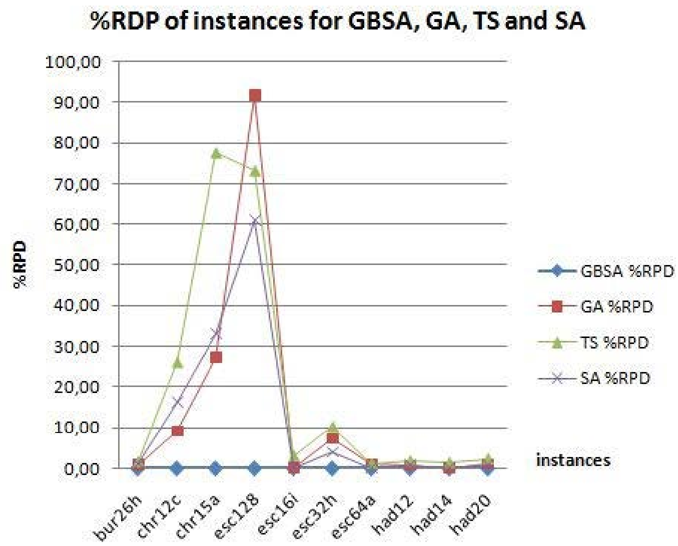

Fig. 6. %RDP of some instances for GBSA, GA, TS and SA algorithms.

VI. Conclusion

The GBSA algorithm is the result of the hybridization of two methods: golden ball metaheuristic and simulated annealing method. This new hybrid algorithm is based on soccer concepts; it incorporates and guides simulated Annealing technique to escape from the local minima and to find the global optimal solution. This method has never been proposed or tested on QAPLIB instances. In this work we proposed an adaptation of our strategy to solve the QAP. The numerical results indicate the efficiency of the proposed GBSA adaptation and its performance compared to algorithms in literature of QAP. As a result, we deduce that our proposed approach has a high convergence speed.

TABLE III. Comparison of GBSA Algorithm with Algorithms in the Literature of the QAP

		GBSA		BMA		BLS		CPTS	
Instance	BKS	%RPD	Time (m)	%RPD	Time (m)	%RPD	Time (m)	%RPD	Time (m)
Sko72	66256	0,543	4.0	0.000	3.5	0.000	4.1	0.000	69.6
Sko81	90998	0,481	4.0	0.000	4.3	0.000	13.9	0.000	121.4
Sko90	115534	0,614	4.0	0.000	15.3	0.000	16.6	0.000	193.7
Sko100a	152002	0,539	4.0	0.000	22.3	0.001	20.8	0.000	304.8
Sko100b	153890	0,679	4.0	0.000	6.5	0.000	10.8	0.000	309.6
Sko100c	147862	0,396	4.0	0.000	12.0	0.000	15.5	0.000	316.1
Sko100d	149576	0,760	4.0	0.006	20.9	0.001	38.9	0.000	309.8
Sko100e	149150	0,528	4.0	0.000	11.9	0.000	42.5	0.000	309.1
Sko100f	149036	0,704	4.0	0.000	23.0	0.000	17.3	0.003	310.3
Tai40a	3139370	3,012	4.0	0.059	8.1	0.022	38.9	0.148	3.5
Tai50a	4938796	3,532	4.0	0.131	42.0	0.157	45.1	0.440	10.3
Tai60a	7205962	2,870	4.0	0.144	67.5	0.251	47.9	0.476	26.4
Tai80a	13499184	2,965	4.0	0.426	65.8	0.517	47.3	0.691	94.8
Tai100a	21052466	2,771	4.0	0.405	44.1	0.430	39.0	0.589	261.2
Tai50b	458821517	0,285	4.0	0.000	1.2	0.000	2.8	0.000	13.8
Tai60b	608215054	0,147	4.0	0.000	5.2	0.000	5.6	0.000	30.4

Moreover, we need to ameliorate this technique even more for some hard QAPLIB instances. Finally, we plan to apply the GBSA algorithm to TSP and compare it with Random-keys Golden Ball algorithm [44]. We plan also to propose a new hybridization such as mixing Golden Ball algorithm with Tabu Search method.

References

[1] S.Sahni, & Gonzalez, T. P-complete approximation problems. Journal of the ACM (JACM), 23(3), 555-565, (1976).

[2] T.C.Koopmans, & M. Beckmann, Assignment problems and the location of economic activities. Econometrica: journal of the Econometric Society, 53-76, (1957).

[3] AN.Elshafei, Hospital layout as a quadratic assignment problem. Operational Research Quarterly, 167-179, (1977).

[4] A. M.Geoffrion, & G. W. Graves, Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach. Operations Research, 24(4), 595-610, (1976).

[5] I. Ugi, J. Bauer, J. Brandt, J. Friedrich, J. Gasteiger, C. Jochum, & W. Schubert, Neue anwendungsgebiete für computer in der chemie. Angewandte Chemie, 91(2), 99-111, (1979).

[6] Z. Drezner, Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem. Computers & Operations Research, 35(3), 717-736, (2008).

[7] Y. Hani, L. Amodeo, F. Yalaoui, & H. Chen, Ant colony optimization for solving an industrial layout problem. European Journal of Operational Research, 183(2), 633-642, (2007).

[8] L. Y. Tseng, & S. C. Liang, A hybrid metaheuristic for the quadratic assignment problem. Computational Optimization and Applications, 34(1), 85-113, (2006).

[9] V.Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 45(1), 41-51, (1985).

[10] R.E.Burkard, E. Cela, P.M.Pardalos, & L.S.Pitsoulis. The quadratic assignment problem. In Handbook of Combinatorial Optimization (pp. 1713-1809). Springer US, (1998).

[11] V. Kaibel, Polyhedral combinatorics of QAPs with less objects than locations. Univ., (1997).

[12] Z. Drezner, Compounded genetic algorithms for the quadratic assignment problem. Operations Research Letters, 33(5), 475-480, (2005).

[13] M. A. El-Baz, A genetic algorithm for facility layout problems of different manufacturing environments. Computers & Industrial Engineering, 47(2), 233-246, (2004).

[14] W. A. N. G. Rong-Long, & K. Okazaki, Solving facility layout problem using an improved genetic algorithm. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 88(2), 606-610, (2005).

[15] Z. Drezner, The extended concentric tabu for the quadratic assignment problem. European Journal of Operational Research, 160(2), 416-422, (2005).

[16] T. James, C. Rego, & F. Glover, A cooperative parallel tabu search algorithm for the quadratic assignment problem. European Journal of Operational Research, 195(3), 810-826, (2009).

[17] A. Misevicius, A tabu search algorithm for the quadratic assignment problem. Computational Optimization and Applications, 30(1), 95-111, (2005).

[18] H. Q. Saremi, B. Abedin, & A. M. Kermani, Website structure improvement: Quadratic assignment problem approach and ant colony meta-heuristic technique. Applied Mathematics and Computation, 195(1), 285-298, (2008).

[19] H. Liu, A. Abraham, & J. Zhang, A particle swarm approach to quadratic assignment problems. In Soft Computing in Industrial Applications (pp. 213-222). Springer Berlin Heidelberg, (2007).

[20] M. Bashiri, & H. Karimi. Effective heuristics and meta-heuristics for the quadratic assignment problem with tuned parameters and analytical comparisons. Journal of Industrial Engineering International, 8(1), 1-9, (2012).

[21] G. Paul.Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem. Operations Research Letters, 38(6), 577-581, (2010).

[22] E. Osaba, F. Diaz, & E. Onieva, A novel meta-heuristic based on soccer concepts to solve routing problems. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation (pp. 1743-1744). ACM, (2013, July).

[23] E. Osaba, F. Diaz, R. Carballedo, E. Onieva, & A. Perallos, Focusing on the Golden Ball Metaheuristic: An Extended Study on a Wider Set of Problems. The Scientific World Journal, 2014, (2014).

[24] A. M. Frieze , G. Galbiati , and F. Maffioli , On the worst-case performance of some algorithms for the asymmetric traveling salesman problem, Networks, 12 (1), 23–39, (1982).

[25] B. Golden, E. Baker, J. Alfaro, and J. Schaffer, The vehicle routing problem with backhauling: two approaches, in Proceedings of the 21st Annual Meeting of SE TIMS, pp. 90–92, South Carolina, SC, USA, (1985).

[26] E.Osaba, & F. Díaz. Design and Implementation of a Combinatorial Optimization Multi-population Meta-heuristic for Solving Vehicle Routing Problems. International Journal of Interactive Multimedia and Artificial Intelligence, 4(2), 89-90, (2016).

[27] I. Rivin, I. Vardi, and P. Zimmermann, The n-queens problem, The American Mathematical Monthly, 101(7), pp. 629– 639, (1994).

[28] S. Martello, and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, NY, USA, (1990).

[29] F. Sayoti, & M. E. Riffi, Golden Ball Algorithm for solving Flow Shop Scheduling Problem. International Journal of Artificial Intelligence and Interactive Multimedia, 4(1), 15-18, (2016).

[30] F. Sayoti, M. E. Riffi, & H. LabaniOptimization of Makespan in Job Shop Scheduling Problem by Golden Ball Algorithm. Indonesian Journal of Electrical Engineering and Computer Science, 4(3), (2016).

[31] A. Misevičius, A modified simulated annealing algorithm for the quadratic assignment problem. Informatica, 14(4), 497-514, (2003).

[32] G. A. Croes, A method for solving traveling-salesman problems. Operations Research, 6(6), 791-812, (1958).

[33] S. Lin, Computer solutions of the traveling salesman problem. Bell System Technical Journal, 44(10), 2245-2269, (1965).

[34] M. Fischetti, Salazar Gonzalez, J. J., & Toth, P.. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Operations Research, 45(3), 378-394, (1997).

[35] C. D. Tarantilis, Solving the vehicle routing problem with adaptive memory programming methodology. Computers & Operations Research, 32(9), 2309-2327, (2005).

[36] R.E.Burkard,S. Karisch, & F.Rendl. QAPLIB-A quadratic assignment problem library. European Journal of Operational Research, 55(1), 115-119, (1991).

[37] W. Ellen E., C.Roger D., & F.Mark A. Parallel simulated annealing using speculative computation. IEEE Transactions on Parallel and Distributed Systems, 1991, 2(4), 483-494.

[38] G.A, El-Nasser A., A.M. Mahmoud, & E.-S. M. El-Horbaty. A comparative study of meta-heuristic algorithms for solving quadratic assignment problem. International Journal of Advanced Computer Science and Applications, 5(1), arXiv:1407.4863, (2014).

[39] J.Skorin-Kapov. Tabu search applied to the quadratic assignment problem. ORSA Journal on computing, 2(1), 33-45,(1990).

[40] E.Taillard. Robust taboo search for the quadratic assignment problem. Parallel computing, 17(4-5), 443-455, (1991).

[41] U.Benlic, & J.K.Hao. Memetic search for the quadratic assignment problem. Expert Systems with Applications, 42(1), 584-595, (2015).

[42] U. Benlic, & J.K. Hao. Breakout local search for the quadratic assignment problem. Applied Mathematics and Computation, 219(9), 4800-4815, (2013).

[43] T.James, C. Rego., & F.Glover. A cooperative parallel tabu search algorithm for the quadratic assignment problem. European Journal of Operational Research, 195(3), 810–826, (2009).

[44] F. Sayoti, & M. E. Riffi, Random-Keys Golden Ball Algorithm for Solving Traveling Salesman Problem. International Review on Modelling and Simulations (IREMOS), 8(1), 84-89, (2015).

Mohammed Essaid Riffi

Ms. Mohammed Essaid Riffi is a professor at the University of Chouaib Doukkali, El Jadida, Morocco, currently he is chief of Computer Science department and former coordinator of Mathematical and Computer Science. Its research axis is the metaheuristics.

Fatima Sayoti

Ms. Fatima Sayoti (15-02-1988) received a master's degree of software quality from University of Chouaib Doukkali, El Jadida, Morocco in 2011. She is currently pursuing his Ph.D. degree (Computer Science) at the University of Chouaib Doukkali, Faculty of Sciences, El Jadida, Morocco. She is also a temporary teacher at Polydisciplinary Faculty, El Jadida (Morocco).