Solving Quadratic Assignment Problem (QAP) Using Invasive Weed Optimization Algorithm

Maryam Omidbakhsh[1] and Mahdi Seifbarghy *[1]

[1] Technical and Engineering Department, Alzahra University, Tehran, Iran

(Received 31 October 2010, Accepted 24 July 2011)

Abstract

A new powerful optimization algorithm inspired from colonizing weeds is utilized to solve the well-known quadratic assignment problem (QAP) which is of application in a large number of practical areas such as plant layout, machinery layout and so on. A set of reference numerical problems from QAPLIB is taken in order to evaluate the efficiency of the algorithm compared with the previous ones which had been applied to solve the addressed problem. The results indicate that the algorithm outperforms the competitive ones for a sizable number of the problems as the problems' dimensions increase.

Keywords: Invasive weed optimization, Meta heuristic algorithms, Quadratic assignment problem, Weed colonization.

Introduction

The quadratic assignment problem (QAP) is known as a special type of assignment problems, which is too complex to be solved using the common exact solution methods (e.g. [1] and [2]). The main difference between QAP and the classic assignment problems is that there is an interaction between each two pairs of facilities, leading to a non-linear objective function. The problem is assigning *n* facilities to *n* locations in such a way as to minimize the transportation costs associated with the flow of materials between the facilities and the distances between the corresponding locations ([3]).

QAP is known as an NP-Hard problem. The solution complexity of QAP is widely accepted among researchers (e.g. [3] and [4]). There is not any known polynomial time algorithm to solve QAP instances with more than a relatively small number of inputs. The exact solution of even small problems ($20 < n < 30$) is considered computationally non-trivial. To obtain optimal solutions for the modest size of QAP instances (i.e. $30<n<40$), a massively parallel computing environment is required [3].

There is a continuous interest in developing different formulations and solution methods among the researches on QAP. Initially, Koopmans and Beckmann [5] proposed the QAP as a mathematical model related to economic activities. Steinberg [6] used the QAP to minimize the number of connections between components in a backboard wiring. Heffley [7] applied the QAP to economics problems. Dickey and Hopkins [8] applied the QAP to the assignment of buildings in a University campus. Francis and White [9] developed a decision framework for assigning a new facility (e.g. police post, supermarket or school) in order to serve a given set of clients. Geoffrion and Graves [10] focused on scheduling problems. Pollatschek et al. [11] invoked the QAP to define the best design for typewriter keyboards and control panels. Elshafei [12] applied it to a hospital planning problem. Krarup and Pruzan [13] applied it to archeology. Hubert [14] applied it to statistical analysis. Bos [15] applied it to a problem related to forest parks. Brusco and Stahl [16] utilized it in a numerical analysis. Parallel computing and networking provide other location analysis problems, which can be formulated as QAPs (e.g. [17,18]. Benjaafar [19] introduced a formulation of the facility layout problem in order to minimize work-in-process. Ciriani et al. [20] explored assigning rooms to persons with undesirable neighborhood constraints could be well formulated as a QAP. A

generalization of the QAP is also used for a service allocation problem by Cordeau et al. [21] with the objective of minimizing the container re-handling operations at a shipyard. Rabak and Sichman [22] and Miranda et al. [23] studied the placement of electronic components as a QAP. For a comprehensive survey of the addressed researches and other applications of the QAP and corresponding special cases, the reader can refer to [24], [25], [26] and [3].

Generally, the methods used in combinatorial optimization problems can be either exact or heuristic. Exact algorithms reach a global optimum within an acceptable time; Some instances are dynamic programming, branch-and-bound, cutting planes or some combinations like branch-and-cut. Heuristic algorithms do not give a guarantee of optimality for the best solution obtained. Meta heuristics are a type of heuristic algorithms with a relatively general structure. The difficulties due to using mathematical optimization methods for solving NP-hard combinatorial problems with a large number of variables and non-linear objective functions have contributed to the development of meta heuristic algorithms. A number of meta heuristic algorithms are based on natural process metaphors such as simulated annealing (SA), genetic algorithms (GA), scatter search (SS) and ant colony optimization (ACO) while a number of them are based on theoretical and experimental considerations such as tabu Search (TS), greedy randomized adaptive Search Procedure (GRASP), variable neighborhood search (VNS) and iterated local search (ILS). There has been applied several meta heuristics algorithms in order to solve the QAP (e.g. [27] and [3]). We now mention to a number of well-known researches in this regard.

Hahn et al. [4] presented a new branch-and-bound algorithm for solving the QAP. The algorithm is based on a dual procedure similar to the Hungarian method for solving the linear assignment problem. It solves the small and large size QAP, however, does not guarantee an optimal solution. Angel and

Zissimopoulos [28] declared that local search was widely used to solve approximately NP-complete combinatorial optimization problems while little was known about the quality of obtained local minima, for a given neighborhood. They proposed an upper bound for the quality of solutions obtained from the deepest local search for the QAP.

Ahuja et al. [1] suggested a genetic algorithm for the QAP and reported its computational behavior. Since the genetic algorithm incorporated many greedy principles in its design, they named it "greedy genetic algorithm". They tested the proposed algorithm on all the benchmark instances of QAPLIB ([29]), a well-known library of QAP instances. Out of the 132 total instances in QAPLIB of varied sizes, the greedy genetic algorithm obtained the best known solution for 103 instances, and for the remaining instances (except one) found solutions within 1% of the best known solutions. Talbi et al. [30] proposed a parallel model for ant colonies to solve the QAP. The cooperation between simulated ants was provided by a pheromone matrix that played the role of a global memory. The exploration of the search space was guided by the evolution of pheromones levels, while exploitation had been boosted by a tabu local search heuristic.

Hasegawa et al. [31] proposed a novel method in order to solve the QAP. Initially, they modify the conventional tabu search into a chaotic version, and then they compare the performance of the novel chaotic search with the conventional tabu search and an exponential tabu search. It is indicated that the exponential tabu search has higher performance than the conventional tabu search, and further that the novel method with a chaotic neural network exhibits the best performance. Misevicius [32] proposed a genetic algorithm hybridized with so-called ruin and recreate a procedure for solving the QAP. The results indicated that the proposed algorithm was of a high performance for the QAP.

In Chawki et al. [33], the network structure of basic solutions to the QAP was revisited. The concept of a relative local star minimum was introduced.

Results characterizing a relative local star minimum were obtained, and an extreme point algorithm was proposed. Misevicius [34] proposed an improved hybrid genetic algorithm (IHGA). It used a robust local improvement procedure as well as an effective restart mechanism that was based on so-called 'shift mutations'. IHGA was applied to the QAP. The results obtained from the experiments on different QAP instances indicated that the proposed algorithm appeared to be superior to other approaches that were among the best algorithms for the QAP.

Erdogan and Tansel [35] declared that exact solution attempts proposed for instances of size larger than 15 had been generally unsuccessful. They presented two new integer formulations based on the flow-based linearization technique which required fewer variables and yielded stronger lower bounds than existing formulations. They also strengthened the formulations with valid inequalities and reported computational experience with a branch-and-cut algorithm.

Drezner [36] introduced the compounded genetic algorithm. They proposed to run a quick genetic algorithm several times as Phase 1, and compile the best solutions in each run to create a starting population for Phase 2. This new approach was implemented on the QAP and caused very good results. In Stützle [37], iterated local search (ILS) as a simple and powerful stochastic local search method was applied to the QAP. Enhanced ILS extensions was also proposed in this research. An experimental evaluation indicated the excellent performance of the enhanced ILS when comparing to the similar state-of-the-art algorithms.

Loiola et al. [3] presented some of the most important QAP formulations and classified them according to their mathematical sources; then, they gave a discussion on the theoretical resources used

to define lower bounds for the exact and heuristic algorithms for QAP. They also gave a detailed discussion of the progresses made in both exact and heuristic solution methods. Drezner [38] solved QAP using various variants of a hybrid genetic algorithm and tested several modifications of the hybrid genetic algorithm. The results indicated the high efficiency of the proposed algorithms.

James et al. [2] introduced a cooperative parallel tabu search algorithm (CPTS) for the QAP. CPTS is a cooperative parallel algorithm in which the processors exchange information throughout the run of the algorithm in contrast to independent concurrent search strategies that aggregate data only at the end of execution. A set of 41 test problems obtained from QAPLIB were used to analyze the quality of the CPTS algorithm. Xia [39] proposed a Lagrangian smoothing algorithm for the QAP where the continuation sub problems were solved by the truncated Frank–Wolfe algorithm. Limited numerical results were also provided in order to measure the efficiency of the proposed algorithm.

In this paper, the invasive weed optimization algorithm (IWO) which has recently proposed by Mehrabian and Lucas 2006 [40] is applied to the QAP. As it will be indicated in Section 4, the addressed algorithm shows a higher performance compared to other similar meta-heuristics, especially when the problem size increases.

The rest of the paper is structured as follows: In Section 2, the formulation of the problem is given. In Section 3, we describe the solution algorithm and customize it to QAP. Section 4 represents the results from utilizing the proposed algorithm on a set of numerical problems. Section 5 gives conclusions and recommends some further research.

Problem formulation

QAP is one of the most difficult problems as a combinatorial optimization problem in which a given set of facilities should be assigned to a set of locations in such a way

that each facility is designated to exactly one location and inversely each location to one facility. The distances between locations, the demand flows between the facilities and the allocation cost of each facility to each location are known. There are different formulation methods for QAP in the literature (e.g. [27] and [3]). The problem's parameters are as follows:

$F = [f_{ij}]$: The flow matrix between facilities i and j; $i = 1, 2, ..., n$ and $j = 1, 2, ..., n$

$D = [d_{kl}]$: The distance matrix between locations k and l; $k, l = 1, 2, ..., n$

$B = [b_{ik}]$: The allocation costs of facility i to location k.

The decision Variable is x_{ik}, a binary variable whose value is 1 if facility i is assigned to location k.

In general, the objective function can be stated as follows:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij}d_{kl}x_{ik}x_{jl} + \sum_{i=1}^{n}\sum_{k=1}^{n} b_{ik}x_{ik} \qquad (1)$$

Since the linear term is easy to solve, we do not consider it; thus, the problem can be stated as follows:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij}d_{kl}x_{ik}x_{jl} \qquad (2)$$

$$s.t: \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad \forall\, 1 \leq j \leq n, \qquad (3)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, 1 \leq i \leq n, \qquad (4)$$

$$x_{ij} \in \{0,1\} \qquad \forall\, 1 \leq i,j \leq n. \qquad (5)$$

The objective function within Eq. (2) represents the total weighted flows between facilities i and j while they are located in locations k and l respectively. Constraint (3) and (4) guarantee each facility being exactly located in one location and each location accommodate exactly one facility. Constraint (5) represents all decision variables must be binary.

Invasive weed optimization algorithm

IWO algorithm is motivated by a common phenomenon in agriculture, which is colonization of invasive weeds. According to the common definition, a weed is any plant growing where it is not wanted. Any tree, vine, shrub, or herb may qualify as a weed, depending on the situation; generally, however, the term is reserved for those plants whose vigorous, invasive habits of growth pose a serious threat to desirable, cultivated plants. Baker and Stebbins [41] mentions that a plant is called weed, if in any specified geographical area, its population grows entirely or predominantly in situations markedly disturbed by man (without, of course, being deliberately cultivated plants). Weeds have shown very robust and adaptive nature, which turns them to undesirable plants in agriculture. The algorithm is simple but has shown to be effective in converging to the optimal solution by employing basic properties, e.g. seeding, growth and competition, in a weed colony [40].

The feasibility, the efficiency and the effectiveness of IWO are tested in details through a set of benchmark multi-dimensional functions, including 'Sphere', 'Griewank' and 'Rastrigin' by Mehrabian and Lucas [40]. The reported results are compared with other recent evolutionary-based algorithms: genetic algorithms, memetic algorithms, particle swarm optimization, and shuffled frog leaping. The results are also compared with different versions of simulated annealing, which are simplex simulated annealing and direct search simulated annealing. The performance of IWO has a reasonable performance for all the test functions. Mallahzadeh et al. [42] utilized IWO algorithm for designing an e-shaped mimo antenna. Karimkashi and Kishk [43] studied IWO Features in Electromagnetics.

To simulate the colonizing behavior of weeds, some basic properties of the process turn out as following ([42]:

1) A finite number of seeds is being spread out over the search area.

2) Every seed grows to a flowering plant and produces seeds depending on its fitness.

3) The produced seeds are being dispersed randomly over the search area and grow to new plants.

4) This process continues until the maximum number of plants is reached; now only the plants with lower fitness can survive and produce seeds, others are being eliminated. The process continues until reaching a maximum number of iterations. Hopefully, the plant with the best fitness is closest to the optimal solution.

The process is addressed in details as follows [40]:

Initialize a population

A population of initial solutions is being dispersed over the d dimensional problem space with random positions. The initial search area is notated by X_{ini} which is bounded by a lower and upper bound. The lower bound value is usually a negative real number while the upper bound is usually a positive real number.

Reproduction

A member of the population of plants is allowed to produce seeds depending on its own and the colony's lowest and highest fitness: the number of seeds each plant produces, increases linearly from a minimum number of seeds, S_{min}, to a maximum number, S_{max}. In other words, a plant will produce seeds based on its fitness, the colony's lowest fitness and highest fitness to make sure the increase is linear. Thus, the above reproduction technique is proposed to give a chance to infeasible individuals to survive and reproduce similar to the mechanism happens in the nature. Moreover, quite often the system can reach the optimal point more easily if it is possible to "cross" an infeasible region (especially in non-convex feasible search space) [44].

Spatial dispersal

The generated seeds are distributed randomly over the d dimensional search space by normally distributed random numbers with mean equal to zero; but varying variance. This means that seeds will be distributed randomly so that they abode near to the parent plant. However, standard deviation of the random function will be reduced from an initial value, $\sigma_{initial}$, to a final value, σ_{final}, in every step (generation).

In simulations, a nonlinear alteration has shown satisfactory performance, which is given as in (6).

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n} (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \tag{6}$$

Where $iter_{max}$ is the maximum number of iterations, σ_{iter} is the standard deviation at the present time step and n is the nonlinear modulation index.

Competitive exclusion

If a plant leaves no offspring then it would go extinct, otherwise it would take over the world. Thus, there is a need of some kind of competition between plants for limiting maximum number of plants in a colony. After passing some iterations, the number of plants in a colony will reach its maximum by fast reproduction, however, it is expected that the desirable plants are reproduced more than the undesirable ones. Reaching the maximum number of plants in the colony, p_{max}, a mechanism for eliminating the plants with poor fitness in the generation activates. When all seeds have found their position in the search area, they are ranked together with their parents' (as a colony of weeds). Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. This mechanism gives a chance to plants with lower fitness to reproduce, and if their offspring has a good fitness in the colony than they can survive.

In this algorithm, number of the initial population N_0, maximum number of iterations it_{max}, maximum number of plants

in the colony p_{max}, nonlinear modulation index n is the key IWO parameters whose effective tuning is very important for the

algorithm convergence. The general structure of the addressed algorithm can be represented as in Fig. 1.
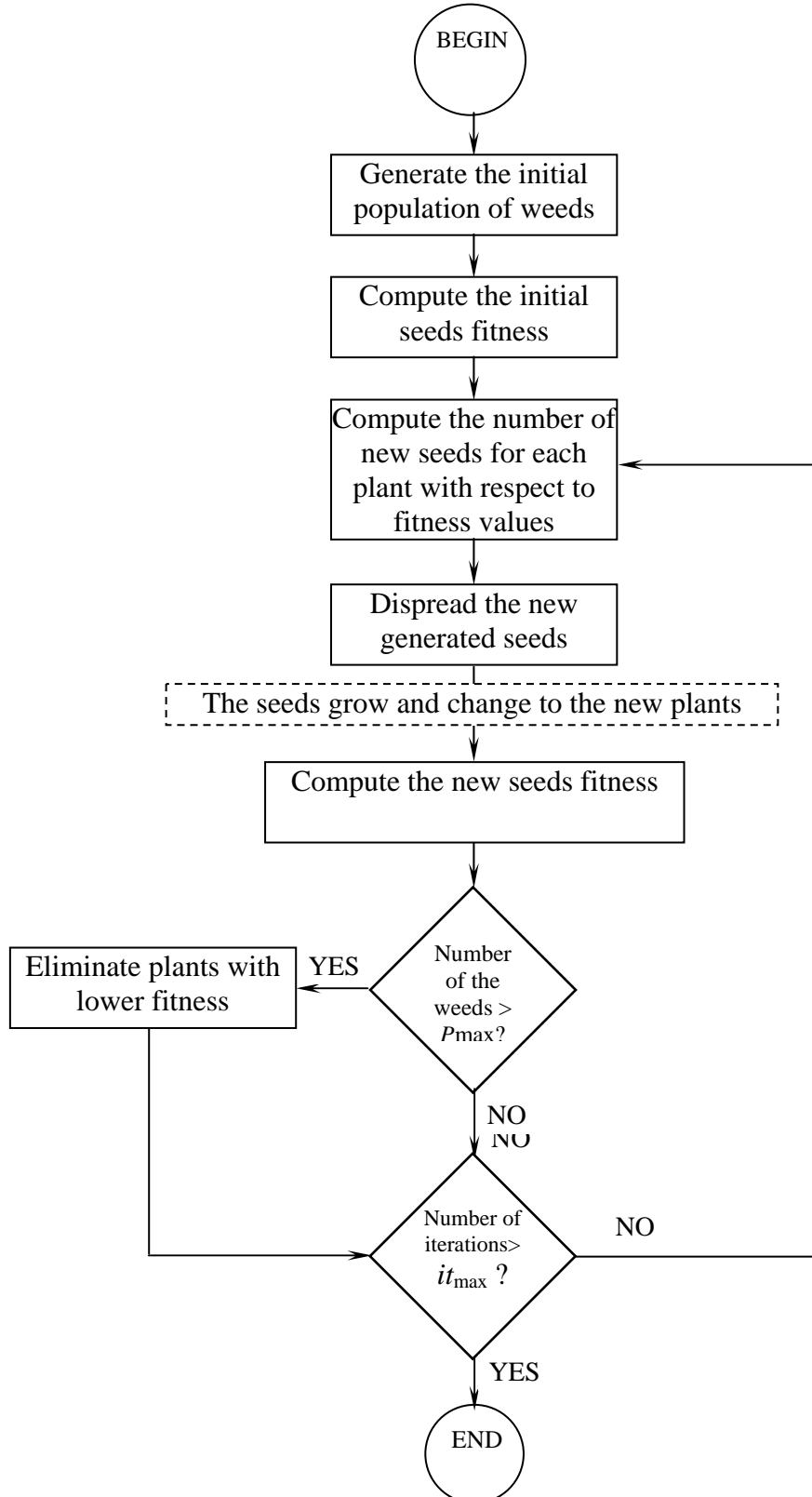

Figure 1: General structure of the IWO algorithm

Customizing IWO algorithm to QAP

IWO can be implemented as the following steps:

Step 1: Define two matrices, the flow matrix between machines (facilities) and the distance matrix between locations: $[f_{ij}]$ and $[d_{kl}]$

Step 2: Generate initial solutions (initial seeds' population) with the dimension of $N_0 \times dim$ in the defined initial search area X_{ini}, by uniformly distributed random numbers with mean equal to zero (each solution is represented by a $1 \times dim$ vector, so the addressed matrix represents N_0 solutions. For example if $X_{ini} \in (Mx = -1, Mn = 1)$ a 1×4 solution may be (-053, 0.72, -0.22, 0.30). If the solution is sorted in ascending order, it will be equal to (1 3 4 2) which means that facility 1 is located in location 1, facility 3 is located in location 2 and, etc.).

Step 3: Compute the fitness value of each seed: calculate the matrix of $[f_{ij} \times d_{kl}]$ and thus the values of the fitness functions for the solutions. After assigning each fitness value to the corresponding seed, it is called a plant.

Step 4: Rank the plants and reproduce new seeds: Plants are ranked due to the corresponding fitness values. They are enabled to generate seeds based on the rank in the colony. The number of new seeds to be generated is calculated for each plant using the linear relationship considering S_{min} and S_{max}.

Step 5: Dispread the new produced seeds: The produced seeds dispread over the search area based on the standard deviation of each step, which is determined using (6).

Step 6: Calculate the fitness values of the new flowering plants.

Step 7: Check the condition I: Is the number of the plants in the current step less than P_{max}?

Yes: Go to step 8

No: Competitive exclusion: the plants with lower fitness values (the higher values of the objective function for the QAP problem) are being eliminated.

Step 8: Check the condition II: Is the iteration number less than it_{max}?

Yes: Repeat the algorithm again.

No: Stop. Calculate the objective function values of the current solutions and select the best value from among them (the minimum value for the objective function of the QAP).

Numerical results

In this section, we aim at evaluating the efficiency of the proposed algorithm using a number of reference pilot numerical problems. The tai*a and tai*b test instances are selected since they are the most famous test problems utilized by other competing algorithms in the literature (James et al. (2009) [18]). As pointed out before, IWO involves some parameters, which should be tuned to provide the QAP best solutions. We have used full factorial design of experiments to tune the algorithm parameters. We have run the algorithm for each problem several times using the different values of the parameters and then have selected the best solution along with the corresponding parameters values.

The parameters' values are as in Table 1.

Tables 2-4 represent the objective function value for the selected test instances along with the best values of the parameters in the considered range. As it can be seen from Table 1-3, the efficiency of the algorithm is nearly increasing as the problem dimension increases. The algorithm gives better results while *dim* is more than 50. The improvement percentage in the objective function versus the problem dimension for the problems with *dim>50* is represented in Fig 2.

Table 1: Initial values of the IWO parameters before tuning

Parameter	Value	Parameter	Value
N_0	15, 20	S_{min}	1
X_{ini}	(-5, 5), (-10, 10)	S_{max}	3, 4
p_{max}	15, 20, 30	$\sigma_{initial}$	20
it_{max}	20, 25	σ_{final}	0.25, 0.5
N	3, 5	dim	Due to the problem

Table 2: IWO best value in comparison to the best known for QAP (chr12 to tai 20) and the best set of parameters

	Instance Name	chr12a	chr15a	tai12a	tai12b	tai15a	tai15b	tai17a	tai20a	tai20b
Initial Parameters	N_0	20	15	20	20	20	20	20	20	20
	dim	12	15	12	12	15	15	17	20	20
	Mx	10	10	5	10	5	5	5	10	5
	Mn	-10	-10	-5	-10	-5	-5	-5	-10	-5
	p_{max}	15	30	20	15	20	15	20	15	20
	it_{max}	20	20	25	20	25	20	25	20	25
	S_{max}	3	3	4	3	4	3	4	3	4
	S_{min}	1	1	1	1	1	1	1	1	1
	$\sigma_{initial}$	20	20	20	20	20	20	20	20	20
	σ_{final}	0.5	0.5	0.25	0.5	0.25	0.25	0.25	0.5	0.25
	n	3	3	3	3	3	3	3	3	3
	IWO Value*	16910	17519	243764	40604857	410378	52098752	533660	750047	133670025
	QAPLIB Best Value	9552	9896	224416	39464925	388214	51765268	491812	703482	122455319

Table 3: IWO best value in comparison to the best known for QAP (tai25 to tai60a) and the best set of parameters

| Instance Name		tai25a	tai25b	tai30b	tai35b	tai40b	tai50a	tai50b	tai60a
Initial Parameters	N_0	20	20	20	15	15	15	20	20
	dim	25	25	30	35	40	50	50	60
	Mx	5	10	10	10	10	10	10	10
	Mn	-5	-10	-10	-10	-10	-10	-10	-10
	p_{max}	20	15	15	30	15	15	20	15
	it_{max}	25	20	20	20	25	20	20	25
	S_{max}	4	3	3	3	3	3	3	3
	S_{min}	1	1	1	1	1	1	1	1
	$\sigma_{initial}$	20	20	20	20	20	20	20	20
	σ_{final}	0.25	0.5	0.5	0.5	0.25	0.5	0.5	0.5
	n	3	3	3	3	3	3	3	3
IWO Value		1294446	367149221	660051474	300436903	647243698	5043300	460928519	7185733
QAPLIB Best Value		1167256	344355646	637117113	283315445	637250948	4938796	458821517	7205962
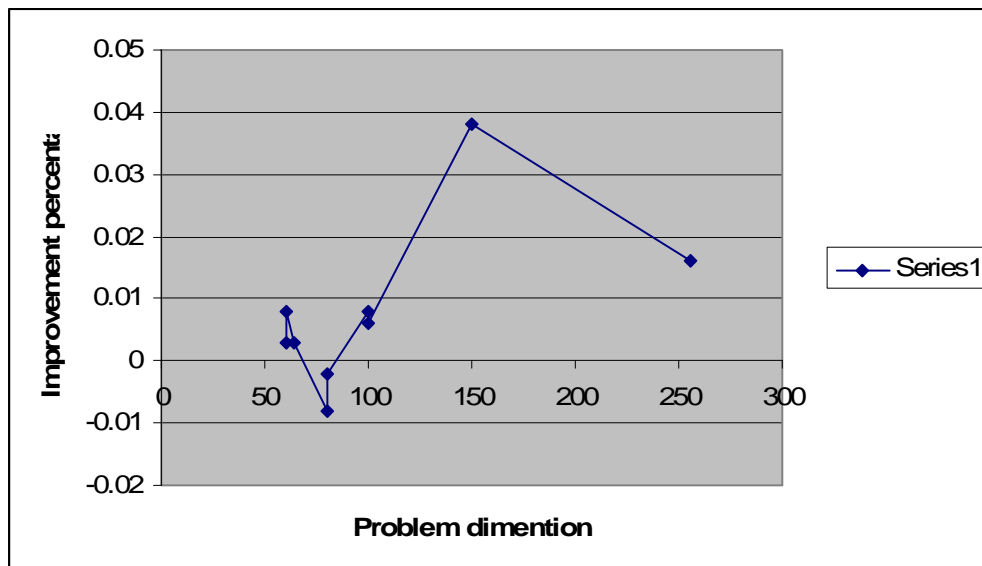
Figure 2: The improvement percentage in the objective function versus the problem dimension

Table 4: IWO best value in comparison to the best known for QAP (tai60b to tai256) and the best set of parameters

Instance Name		tai60b	tai64c	tai80a	tai80b	tai100a	tai100b	tai150b	tai256c
Initial Parameters	N_0	20	20	20	15	20	20	15	20
	dim	60	64	80	80	100	100	150	256
	Mx	5	5	5	5	5	10	5	5
	Mn	-5	-5	-5	-5	-5	-10	-5	-5
	p_{max}	15	30	30	20	20	20	20	20
	it_{max}	20	20	20	20	20	20	20	20
	S_{max}	3	3	4	4	3	3	3	3
	S_{min}	1	1	1	1	1	1	1	1
	$\sigma_{initial}$	20	20	20	20	20	20	20	20
	σ_{final}	0.5	0.5	0.25	0.25	0.5	0.5	0.25	0.5
	n	3	3	3	3	3	3	3	3
IWO Value		603355774	1849510	13619277	819916528	20877903	1179216180	480136527	44043511
QAPLIB Best Value		608215054	1855928	13515450	818415043	21052466	1185996137	498896643	44759294

*Note: The algorithm was run by different value of initial parameters. The best results among these parameters' value are displayed as "IWO Value" row. QAP best values are taken from www.qaplib.com.

The average percentage of improvement in the objective function value is about 0.8 % for the addressed problems. The algorithm was coded in MATLAB 7.6 and run on an Intel Pentium Dual Core of 1.5 GHz CPU computer.

Conclusions and further research

Heuristics or suboptimal algorithms are often used to obtain solutions for QAP instances. IWO is a numerical stochastic search algorithm inspired from natural behavior of weeds colonizing in opportunity spaces for function optimization. Mehrabian and Lucas [40] studied the convergence of IWO for three benchmark problems. The performance of the algorithm was enough good to be applied to QAP. A brief description of the algorithm was given and the algorithm was run on a set of reference numerical problems from QAPLIB (www.qaplib.com). The results indicated that IWO dominated the best solution given by QAPLIB when the problem dimension is more than 50. The average percentage of improvement in the objective function value is about 0.8 % for the addressed problems.

References:

1- Ahuja, R.K., Orlin, J.B. and Tiwari, A. (2000). "A greedy genetic algorithm for the quadratic assignment Problem." *Computers & Operations Research*, Vol. 27, PP. 917-934.

2- James T., Rego C. and Glover F. (2009). "A cooperative parallel tabu search algorithm for the quadratic assignment problem." *European Journal of Operational Research*, Vol. 195, PP. 810-826.

3- Loiola, E.M., Maia de Abreu, N.M., Oswaldo Boaventura-Netto, P., Hahn, P. and Querido, T. (2007). "A survey for the quadratic assignment problem." *European Journal of Operational Research*, Vol. 176, PP. 657–690.

4- Hahn, P., Grant, T. and Hall, N. (1998). "A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method." European Journal of Operational Research, Vol. 108, PP. 629-640.

5- Koopmans T.C. and Beckmann M.J. (1957). "Assignment problems and the location of economics activities." *Journal of Econometrica*, Vol. 25, PP. 53-76.

6- Steinberg L. (1961). "The backboard wiring problem: A placement algorithm," *Journal of SIAM Review*, Vol. 3, PP. 37-50.

7- Heffley D. R. (1980). "Decomposition of the Koopmans–Beckmann problem." *Journal of Regional Science and Urban Economics*, Vol. 10, No. 4, PP. 571-580.

8- Dickey J. W. and Hopkins J. W. (1972). "Campus building arrangement using TOPAZ." *Journal of Transportation Research*, Vol. 6, PP. 59-68.

9- Francis R. L. and White J. A. (1974). *Facility Layout and Location: An Analytical Approach*. Englewood Cliffs, NJ: Prentice-Hall.

[10] Geoffrion A. M. and Graves G. W. (1976). "Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach." *Journal of Operations Research*, Vol. 24, PP. 595-610.

11- Pollatschek M.A., Gershoni N. and Radday Y.T. (1976). "Optimization of the typewriter keyboard by simulation." *Journal of Angewandte Informatik*, Vol. 17, PP. 438–439.

12- Elshafei A. N. (1977). "Hospital layout as a quadratic assignment problem." *Journal of Operational research quarterly*, Vol. 28, PP. 167-179.

13- Krarup J. and Pruzan P.M. (1978). "Computer-aided layout design." *Journal of Mathematical Programming Study*, Vol. 9, PP. 75-94.

14- Hubert L. J. (1987). *Assignment Methods in Combinatorial Data Analysis*. New York and Basel: Marcel Dekker, INC.

15- Bos J. (1993). "Zoning in forest management: a quadratic assignment problem solved by simulated annealing." *Journal of Environment Manage*, Vol. 37, PP. 127-145.

16- Brusco M. J. and Stahl S. (2000). "Using quadratic assignment methods to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices." *Journal of Classification*, Vol. 17, PP. 197-223.

17- Gutjahr W. J., Hitz M. and Mueck M. A. (1997). "Task assignment in Caley interconnection topologies." *Journal of Parallel Computing*, Vol. 23, PP. 1429-1460.

18- Siu F. and Chang R. K. C. (2002). "Effectiveness of optimal node assignments in wavelength division multiplexing networks with fixed regular virtual topogies." *Journal of Computer Network*, Vol. 38, PP. 61-74.

19- Benjaafar S. (2002). "Modeling and analysis of congestion in the design of facility layouts." *J. of Management Science*, Vol. 48, PP. 679-704.

20- Ciriani V., Pisanti N. and Bernasconi A. (2004). "Room Allocation: A polynomial Subcase of the Quadratic Assignment Problem." *Journal of Discrete pplied Mathematics*, Vol. 144, No. 3, PP. 263-269.

21- Cordeau J. F., Gaudioso M., Laporte G. and Moccia L. (2003). "A memetic heuristic for the generalized quadratic assignment problem." *Centre de recherche surles transports*, Montréal, CRT-2003-40.

22- Rabak C. S. and Sichman J. S. (2003). "Using A-Teams to optimize automatic insertion of electronic components." *Journal of Advanced Engineering Informatics*, Vol. 17, PP. 95–106.

23- Miranda G., Luna H. P. L., Mateus G. R. and Ferreira R. P. M. (2005). "A performance guarantees heuristic for electronic components placement problems including thermal effects." *Journal of Computers and Operation Research*, Vol. 32, PP. 2937-2957.

24- Cela F. (1998). *The Quadratic Assignment Problem: Theory and Algorithms*. Dordrecht: Kluwer Academic Publishers.

25- Rendl F., Drezner Z. and Hamacher H. (2004). *The quadratic assignment problem, Facility Location: Applications and Theory*, New York: Springer.

26- Commander C. W. (2005). "A Survey of the Quadratic Assignment Problem with Applications." *Morehead Electronic Journal of Applicable Mathematics*, Volo. 4, PP. 1-15.

27- Drezner Z. (2002). "Heuristic algorithms for the solution of the quadratic assignment problem." *Journal of Applied Mathematics and Decision Sciences*, Vol. 6, PP. 163-173.

28- Angel, E. and Zissimopoulos, V. (1998). "On the quality of local search for the quadratic assignment problem." *Discrete Applied Mathematics*, Vol 82, PP. 15- 25.

29- Burkard, R.E., Karisch, S.E. and Rendl, F. (1997). "QAPLIB – A Quadratic Assignment Problem Library." *Journal of Global Optimization*, Vol. 10, PP. 391–403.

30- Talbi, E.G., Roux, O., Fonlupt, C. and Robillard, D. (2001). "Parallel Ant Colonies for the quadratic assignment problem." *Future Generation Computer Systems*, Vol. 17, PP. 441–449.

31- Hasegawa, M., Ikeguchi, T., Aihara, K. and Itoh, K. (2002). "A novel chaotic search for quadratic assignment problems." *European Journal of Operational Research*, Vol. 139, PP. 543–556.

32- Misevicius, A. (2003). "Genetic algorithm hybridized with ruin and recreate procedure: application to the quadratic assignment problem." *Knowledge-Based Systems*, Vol. 16, PP. 261–268.

33- Chawki, A., Fedjki, S. and Duffuaa, O. (2004). "An extreme point algorithm for a local minimum solution to the quadratic assignment problem." *European Journal of Operational Research*, Vol. 156, PP. 566–578.

34- Misevicius, A. (2004). "An improved hybrid genetic algorithm: new results

for the quadratic assignment problem." *Knowledge-Based Systems*, Vol. 17, PP. 65–73.

35- Erdogan, G. and Tansel, B. (2007). "A branch-and-cut algorithm for quadratic assignment problems based on linearizations." *Computers & Operations Research*, Vol. 34, PP. 1085-1106.

36- Drezner, Z. (2005). "Compounded genetic algorithms for the quadratic assignment problem." *Operations Research Letters*, Vol. 33, PP. 475 – 480.

37- Stützle, T. (2006). "Iterated local search for the quadratic assignment problem." *European Journal of Operational Research*, Vol. 174, No. 3, PP. 1519-1539.

38- Drezner, Z. (2008). "Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem." *Computers & Operations Research*, Vol. 35, PP. 717 – 736.

39- Xia, Y. (2010). "An efficient continuation method for quadratic assignment problems." *Computers & Operations Research*, Vol. 37, PP. 1027–1032.

40- Mehrabian A. R. and Lucas C. (2006). "A novel numerical optimization algorithm inspired from weed colonization." *Journal of Ecological Informatics*, Vol. 1, PP. 355–366.

41- Baker H. G. and Stebbins G. L. (1965). *The Genetics of Colonizing Specie*s. Academic Press, New York

42- Mallahzadeh A. R., Es'haghi S. and Alipour A. (2009). "Design of an E-shaped mimo antenna using IWO algorithm for wireless application." *Journal of Progress In Electromagnetics Research*, Vol. 90, PP. 187-203.

43- Karimkashi Sh. and Kishk A. (2010). "Invasive Weed Optimization and its Features in Electromagnetics." IEEE transactions on antennas and propagation, PP. 1269- 1278.

44- Yuchi M. and Kim J. H. (2005). "Ecology-inspired evolutionary algorithm using feasibility-based grouping for constrained optimization." Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK, PP. 1455-1461.