New Simulated Annealing Algorithm for Quadratic Assignment Problem

Kambiz Shojaee Ghandeshtani
Department of Electrical Engineering
University of Tehran
Tehran, Iran
E-mail: k.shojaee@ece.ut.ac.ir

Nima Mollai
Dept. of Electrical Engineering
Sadjad institute of higher education
Mashhad, Iran
E-mail: nimamollai@yahoo.com

Seyed Mohammad Hosein Seyedkashi
Dept of Mechanical Engineering
Tarbiat Modares University
Tehran, Iran
E-mail: seyedkashi@modares.ac.ir

Mohammad Mohsen Neshati
Department of Electrical Engineering,
Ferdowsi University of Mashhad
Mashhad, Iran
E-mail: mohsen_neshati@stu-mail.um.ac.ir

***Abstract*— In facility layout design, the problem of locating facilities with material flow between them was formulated as a Quadratic Assignment Problem (QAP), so that the total cost to move the required material between the facilities is minimized, where the cost is defined by a quadratic function. In this paper, a new definition in cooling scheduling is proposed for simulated annealing algorithm to solve the QAPs. Also a simple greedy-type algorithm is proposed to improve this method. The algorithm is implemented and tested on 40 benchmarks. In comparison with many other recently developed methods, considerable results are obtained by this approach.**

***Keywords*- QAP; Simulated annealing; Cooling Schedule; Greedy search.**

I. INTRODUCTION

Quadratic Assignment Problem or QAP is one of the most known and complicated problems in combinational optimization problems which was proposed by Koopmans and Beckmann in 1957 [1]. In 1976, Sahni and Gonzales [2] showed that QAP belongs to the class of NP-hard problems. QAP has been considered by many researchers for a long time and is capable to model many daily real problems. Among the applications of QAP, typewriter keyboard design [3], electronic components placement problems [4], campus planning [5], hospital layout [6], numerical analysis [7] and memory layout optimization in signal processors [8] can be mentioned.

In QAP, several facilities (for example *n* factories) are assigned to several locations (for example *n* cities) in such a way that the distance between any of the locations and also the flow between any of the facilities, are constant and predetermined. This assignment should be in a way that the goal function which is affected by the distance between the locations and the circulation number of the goods between the facilities, is minimized. In general, when the goal is to allocate *n* facilities to *n* locations, the number of possible situations is $n!$. That is why this problem is in the NP-complete problem category. It seems that the most achievable deterministic method to solve QAP is the branch-and-bound algorithm [5, 9, 10]. Recent researches illustrate that the accurate solving of QAP takes place with an n up to 36 [11] which in that case takes a long time [12]. So the researchers usually try to use metaheuristic methods to solve QAP. Some of these methods are Neural Networks algorithm [13], Simulated Annealing [14, 15], Threshold Accepting [16], Genetic Algorithms [17, 18], Tabu Search [19-21], Ant Colony Optimization [22, 23], Scatter Search [24].

In this paper, a new version of SA algorithm for QAP solving is presented with redefinition of Time Scheduling program parameters. In Section 2, QAP is fully explained. In Section 3, the base algorithm of simulated annealing is presented. The proposed simulation algorithm for QAP solving is discussed in the next section. In Section 5, the results of simulation is presented and compared with other metaheuristic methods.

II. QAP DESCRIPTION

In the mathematical definition of QAP, there exist n locations with specific coordinates. So an $n \times n$ matrix representing the distance between each pair of the locations will be generated $\left(D = \lfloor d_{ij} \rfloor_{n \times n}\right)$. The other $n \times n$ matrix generated includes the flow of each pair of facilities $\left(F = \lfloor f_{ij} \rfloor_{n \times n}\right)$. Considering the distance between locations and the flow between the facilities, the goal is to find the minimum cost for assigning the facilities. Mathematically, if $S(n)$ is assumed as a set of all possible permutations for a set of {0,1, …, n}, the goal is to find a permutation such as $p \in S(n)$ which is able to minimize a cost function defined as Equation 1.

$$Z(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} f_{p(i)p(j)} \quad (1)$$

 ISBN: 978-1-61208-101-4

In fact, the p permutation shows the sequence of facilities placement in locations. More description is showed with an example in Figure 1.

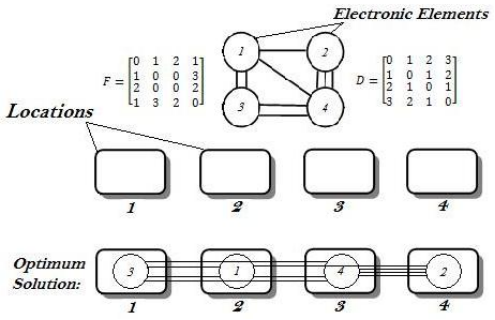

Figure 1. A QAP sample for assignment of 4 facilities to 4 locations

In this figure, it is decided to place 4 electronic elements in 4 specific locations. Matrix D shows the distance between each pair of the locations and Matrix F shows the number of required connections between two parts. As shown in Figure 1, considering the length of consumed wires, the optimum answer for this problem is p=(3,1,4,2). In other words, the optimum result is obtained when the element #3 is placed in location #1, element #1 in location #2, element #4 in location #3 and element #2 in location #4.

III. BASE SIMULATED ANNEALING ALGORITHM

The idea of simulated annealing algorithm was first proposed as the modified Monte Carlo method by Metropolis in 1953 who was working in the publishing industry [25]. He resembled paper to the material which is obtained after cooling of a molten material. SA for combinational optimization applications such as Traveling Salesman Problem was first developed by Kirkpatrick in 1983 inspired by Metropolis algorithm [26]. This algorithm is an adoption of cooling process in which metal is heated to its melting point and then slowly cools. This reduction in temperature is in such a way that the system will approximately be in thermodynamic equilibrium. During the gradual temperature reduction, the system becomes more ordered and approaches to the steady state with minimum energy. The main plan in the determination of temperature and the initial energy state of thermodynamic system is that if the energy changes are negative, the new structure (energy and temperature) will be accepted but if the changes are positive, the acceptance is dependent on Boltzmann distribution function:

$$P\{accept\} = \begin{cases} 1 & ,\Delta f \le 0 \\ e^{-\Delta f / CT} & ,\Delta f > 0 \end{cases} \quad (2)$$

In which Δf is the change value of cost function, T is temperature parameter in simulated annealing process, P is the acceptance probability of the next point and C is a control parameter known as Boltzmann's constant with positive value [27]. The whole process will be repeated while the energy is minimized and the system reaches to the steady state. This algorithm is suitable for mixed discrete problems and complicated nonlinearity problems. In the SA algorithm, cooling schedule parameters control the process in search algorithm. Cooling schedule consists of three factors:

1) Initial temperature (T_0).

2) Convergence criterion or Freezing temperature (T_f).

3) Cooling function.

In this algorithm, when the initial and freezing temperatures are defined properly and the rate of temperature reduction is less than the slope of $T_k=T_0/(1+\log(k))$, then the SA algorithm will be converged to the absolute minimum when the number of tries (k) tends to infinity. But, According to the slope of this curve, temperature reduction makes the solving time very longer, therefore, faster temperature reduction functions are usually used such as $T_{k+1} = \alpha * T_k$ in which $0.8<\alpha<1$ or $T_k=T_{k-1}/(1+\log(k))$. Using these functions, the number of temperature steps from melting point to freezing point has a considerable reduction and hence the probability of passing through an effective temperature range for optimal search also decreases. So by the definition of numerous iterations in the inner search loop including new generation, assessment and decision making in each temperature, it is tried to give enough search time in optimum temperature range for the algorithm. In this algorithm the number of iterations in a specific temperature is called Markov chain length. SA pseudo-code algorithm is as following:

1) Randomly initialize the solution $V = V_{Start}$.

2) Set the initial temperature $T = T_0$.

3) Until stop criterion is reached, do:

Generate a new solution V' from the neighborhood of V.

Let E and E' be the values of the cost function at S and S', respectively.

If (E' < E), accept new solution V = V'.

Else if (exp(-(E'-E)/T)) > a random number Є [0,1], accept new solution V = V'.

4) If freezing condition (convergence criterion) is valid, stop.

5) Reduce the temperature by cooling function.

6) Go to 3).

The way of producing new generation, based on the current generation, makes SA algorithm distinct in continuous or discrete problems. In continuous problems, some definitions such as neighborhood radius are used for producing a new generation in neighborhood of the current generation. In the SA process, the neighborhood radius is reduced according to the temperature in order to increase the convergence speed. This new generation in discrete problems is performed by some operators which implicitly generate the next generation in neighborhood of the current generation. These operators are also called Move Set. In QAP, each possible answer for the problem is corresponding to a permutation of 1 to n. several effective operators in discrete problems such as QAP are as follows:

 ISBN: 978-1-61208-101-4

1. Switching Or Swap Operator:

Randomly selects two locations from permutation and replaces with each other.

2. Translation Operator:

Randomly selects a portion of permutation and replaces in another random location in permutation.

3. k-Opt Operator:

In k-Opt move, the tour is broken into k parts, then the k parts reconnect in the other possible way. Inversion is the case of k-Opt in which $k = 2$.

In fact, these operators are used as local search approaches, in the global search approaches, such as, Tabu search, Simulated Annealing or Genetic algorithm.

IV. PROPOSED ALGORITHM

A. Proper Determination of Initial Temperature

Kirkpatrick [26] defined the initial temperature as essentially all proposed circuit flips are accepted but in quantitative definition of this qualitative significance has sufficed to presentation of a constant value (10) for his problem, so only with the justification that the initial temperature of an algorithm is sufficient to start, the probability of high initial temperature and hence non optimum operation of the algorithm is neglected. Percy [28] has assumed the initial temperature from 100000 to 4000000 according to the dimensions of the problem. Andrew [29] has changed the initial temperature from 0.001 to 100 and discussed the effect of this important parameter in temperature reduction.

In this paper, the initial temperature is defined in such a way that the proportion of, the accepted cases to the whole studied cases (γ), in Markov chain has the value of 0.2-0.5 according to different problems.

B. New Definition for Markov Chain Lenght

Since the Markov chain length is in fact giving enough time for search to the algorithm, it may be reconsidered according to the working temperature in order to optimum use of effective temperature. Constant definition of the number of iterations for search loop in a constant temperature condition (Markov chain length) is the definition of the same need of algorithm try in different temperatures for search. Kirkpatrick [26] has mentioned effective temperature range in search process of the algorithm which declares the effectiveness of search in this range. So the constant definition of inner search chain is not optimum. In this paper, the number of iterations for a specific temperature is proportional to the number of acceptances in an inner loop instead of the number of tries for generation and evaluation. The number of acceptances required for search in the inner loop of the algorithm decreases according to the temperature reduction.

C. Proper Definition of the Freezing Temperature

Definition of the freezing temperature or convergence condition is very important in increasing the speed and accuracy in the search process. If the stop condition of the algorithm is not defined effectively, the algorithm will be stopped sooner which as a result reduces the accuracy or the convergence of the algorithm will be announced by delay which results in the speed reduction.

In different papers the way of determining the convergence conditions for algorithm is explained in different methods and various criteria are discussed. For example Kirkpatrick [26] defined that:

"If the desired number of acceptances is not achieved at three successive temperatures, the system is considered "freeze " and annealing stops."

Percy [28] has considered the approach to the optimum response or to the specific number of tries (500 iterations) as the stop condition of the algorithm.

In this paper, the stop condition is defined when in two sequential searches in Markov chain, there is no change in the best obtained result.

D. Improving the Results by a Simple Greedy Algorithm

In this paper, a kind of a greedy algorithm is used for local search at the end of the simulated annealing algorithm. This algorithm gives the final response of the algorithm. So considering [30], we can say that when Matrices of D and F are symmetric, if permutation of p' is created by replacement of the s_{th} and t_{th} elements in permutation of p, the cost function is calculated by Equation 4.

$$\Delta(p,s,t) = Z(p') - Z(p) = -2 \sum_{\substack{k=1 \\ k \neq s,t}}^{n} \left(f_{p(s)p(k)} - f_{p(t)p(k)}\right)\left(d_{sk} - d_{tk}\right) \qquad (3)$$

Based on this equation, a matrix is defined as $\Delta(p) = [\Delta_{st}]_{n \times n}$ in which Δ_{st} shows the difference in cost function because of displacement of the s_{th} and t_{th} elements in permutation of p. In this simple algorithm, the elements which produce the negative element in the matrix are moved until there will be no negative number in the matrix. For the determination of the negative elements' displacement priority, the lowest negative element is selected greedily.

V. SIMULATION AND COMPARISON

The proposed algorithm is executed for a sample problem presented in QAPLIB site [11] and obtained results are compared with other algorithms. Considering [19], the standard problems discussed in QAPLIB can be classified in 4 categories.

I. Unstructured, randomly generated instances:

They are the problems in which distance and flow matrices are generated randomly with uniform distribution. These problems are usually more complicated than other QAPs. For example *taixxa* is in this category. (Each *x* is an integer)

II. Instances with grid-based distance matrix:

They are the problems in which distance matrix is created inspiring some points in Manhattan Island and flow matrix is randomly created. For example *Nugxx* and *Skoxx* are in this category.

III. Real-life instances:

These problems are derived from real applications of QAP. For instance, hospital layout is discussed in *Kra30x* category and typewriter keyboard design is

discussed in *Bur26x* category. Their flow matrix has usually more zero in comparison with other categories and the input distribution of their flow matrix is not uniform.

IV. Real-life like instances:

Since the size of real-life instances are not so big, E. Taillard discussed Taixxb problems [19], which are like real-life instances with the same distribution to compensate this lack.

Considering the difference in QAP problems, in order to gain a better result, some parameters must be changed. The proposed algorithm starts with the initial temperature proportional to 20% of acceptances to the whole situations, (γ = 0.2) for categories (I) and (II), proportional to 50% of acceptances to the whole situations, (γ = 0.5) for category (III) and proportional to 40% of acceptances to the whole situations, (γ = 0.4) for category (IV).

This ratio has been obtained by the trial and error method in various problems. In the case that at the beginning of the algorithm, searching for the initial temperature performs with proportionate steps starting from the initial value of zero and this temperature increases until the ratio of the number of accepted cases to the total studied cases (γ) in a single markov chain length, reaches the determined ratio, we will reach a temperature equivalent to the melting point.

Finding the above ratio will be very important in the definition of the optimized initial temperature, in order to have a high speed in addition to maintaining the accuracy of the search process.

It has to be said that the performed operation in the existing loop in defining the initial temperature, is exactly the same operation used in the search engine.

This means that it starts with a random variable and after applying the switching operator, the acceptance terms of the algorithm will be checked. In the case of acceptance, the previous generation will be replaced by a new one and the operation will be continued until the markov chain ends and at the end, it is the ratio of accepted, to the total states that represents the desired ratio. If this ratio is sufficient, the loop will stop, but otherwise, a new ratio will be calculated for the increased temperature by the stepped increase of the temperature and repetition of the above stages. This temperature increase will be continued until the ratio of the accepted states to the total states in a single markov chain, reaches the ratio defined at the beginning of the algorithm.

The iterations in the first inner loop will finish when a specific number of acceptances occur related to A_0=2000n/(1-γ). In each temperature reduction the number of acceptances reduces with the equation $A_k = 0.8\sqrt[k]{5}A_0$. Therefore, more tries are performed in effective temperatures in the search process. The repetition rate reduction has been resulted by trial and error.

In the inner loop for producing the new generation, the Switching operator is used. It seems that this operator can find the best possible result.

Simulation and optimization of process is performed in Matlab7.7 by a Core2Due computer with a 2.66 GHZ CPU and 4GB of RAM. The effectiveness of the simulated annealing algorithm combined with the new cooling schedule mentioned in Section IV, has been studied by solving some of the complicated QAPs reported in the literature available in the QAPLIB. The criterion considered for evaluating the performance is The Average Percent Deviation (APD) of the solution quality from the Best-Known Solution (BKS) from the literature. APD is determined as follows:

$$APD = 100*(C - BKS) / BKS,$$

where, C and BKS are, calculated cost by the proposed algorithm and the best-known solution, respectively.

Table I provides a comparison between all the variants of QAP categories. The APD and the average time to completion obtained by the new approaches are compared with the other results given in recent novel researches. The first method chosen for comparison is the iterated fast local search algorithm by using order crossover with random sliding mutation named as IFLS / OXSM [31].The second one is a new iterated fast local search (NIFLS) algorithm by recombination of crossover with sliding mutation (RCSM) scheme that is referred as NIFLS / RCSM proposed in [32]. Finally, the results are compared with [12], for its new diversification TS variants for the QAP named as DivTS.

VI. Conclusion and Results

As the compared results in Table I shows, the new proposed definitions of cooling schedule in SA algorithm indicate the performance improvement in comparison with other algorithms. As it is obvious, the result of proposed method in APD criterion and average time of completion are 0.40 and 78% respectively, which are both better than the results in [31]. Also 0.50 and 87% improvement in APD and running time in comparison with [32] are obtained. Eventually by comparing the results with [12], it's shown that the average time to completion has been improved 63% but APD criterion has been weakened about 0.28.

Acknowledgment

This work is supported by Nano-Age Technology Group in Mashhad, Iran (www.nanoage.ir).

References

[1] T.C. Koopmans and M.J. Beckmann, "Assignment problems and the location of economic activities," Econometrica, vol. 25, pp. 53–76, 1957.

[2] S. Sahni and T. Gonzales, "P-complete approximation problems," Journal of the Association for Computing Machinery, vol. 23, pp. 555–565, 1976.

[3] M.A. Pollatschek, N. Gershoni, and Y.T. Radday, "Optimization of the typewriter keyboard by simulation," Angewandte Informatik, vol. 17, pp. 438-439, 1976.

[4] G. Miranda, H.P.L. Luna, G.R. Mateus, and R.P.M. Ferreira, "A performance guarantee heuristic for electronic components placement problems including thermal effects," Computers and Operations Research, vol. 32, pp. 2937–2957, 2005.

 ISBN: 978-1-61208-101-4

TABLE I: PERFORMANCE OF THE PROPOSED SA FOR THE SELECTED QAP INSTANCES FROM QAPLIB IN COMPARISON WITH BEST KNOWN SOLUTIONS AND OTHER METHODS.

Instances					Proposed SA		IFLS / OXSM		NIFLS / RCSM		DivTS	
No.	Category	*problem Name*	*N*	BKS	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*
1	Unstructured, randomly generated	Lipa30a	30	13178	0.000	12.40	0.000	119.72	0.000	81.57	0.000	72
2		Lipa40a	40	31538	0.000	51.23	0.000	489.91	1.060	346.37	0.000	189.6
3		Lipa50a	50	62093	0.083	188.02	1.020	1556.28	0.740	1061.26	0.000	394.2
4		Rou15	15	354210	0.000	0.74	0.000	2.95	0.000	2.89	0.000	54
5		Rou20	20	725522	0.115	19.38	0.020	11.73	0.000	11.40	0.000	14.4
6		Tai30a	30	1818146	0.680	72.64	1.110	83.06	1.510	82.81	0.000	78.6
7		Tai40a	40	3139370	1.349	233.20	1.850	354.38	1.870	346.93	0.222	309.6
8		Tai50a	50	4941410	1.803	462.58	2.250	1104.03	2.130	1076.12	0.725	613.8
9		Tai60a	60	7208572	1.930	948.61	2.750	2739.83	1.610	2701.20	0.718	1541.4
10		Tai80a	80	13557864	1.487	1242.47	2.340	11332.95	2.160	11584.81	0.753	3164.4
11	Grid-based distance matrix	Nug20	20	2570	0.000	6.21	0.000	16.06	0.000	11.16	0.000	13.8
12		Nug24	24	3488	0.000	8.81	0.000	39.75	0.000	27.29	0.000	24
13		Nug27	27	5234	0.000	16.94	0.000	80.56	0.000	53.49	0.000	34.8
14		Scr12	12	31410	0.000	0.12	0.000	1.11	0.000	11.09	0.000	24
15		Scr15	15	51140	0.000	0.35	0.000	3.09	0.000	3.11	0.000	54
16		Scr20	20	110030	0.001	7.90	0.000	12.69	0.000	12.34	0.000	13.8
17		Sko56	56	34458	0.144	522.73	0.470	2612.69	0.270	1828.30	0.002	789.6
18		Sko72	72	66256	0.188	1304.26	0.730	8663.36	0.540	6169.37	0.006	2278.8
19		Sko81	81	90998	0.070	2004.03	0.430	16959.59	0.510	11729.39	0.016	3381.6
20		Sko100a	100	152002	0.099	4211.41	1.300	308.66	0.320	33616.39	0.027	7753.2
21		Tho30	30	149936	0.073	51.82	0.290	118.94	0.350	78.12	0.000	72
22		Wil50	50	48816	0.068	336.72	0.280	1498.69	0.240	1039.73	0.000	475.2
23	Real-life	Bur26h	26	7098658	0.008	23.37	0.000	57.47	0.000	37.44	0.000	31.2
24		Chr12c	12	11156	0.000	2.38	0.000	1.02	0.000	1.01	0.000	24
25		Chr15a	15	9896	0.827	6.79	0.000	2.97	1.150	2.95	0.000	54
26		Esc16j	16	8	0.000	0.05	0.000	2.91	0.000	2.03	0.000	6.6
27		Esc32h	32	438	0.000	24.19	0.000	85.75	0.000	54.90	0.000	82.2
28		Esc64a	64	116	0.000	1.09	0.000	1521.70	0.000	1059.92	0.000	1041.6
29		Esc128	128	64	0.000	71.80	-	-	0.000	23370.06	0.000	9781.2
30		Had12	12	1652	0.000	1.01	0.000	0.97	0.000	0.96	0.000	24
31		Had14	14	2724	0.000	2.21	0.000	1.97	0.730	2.05	0.000	42
32		Had20	20	6922	0.062	11.12	0.000	10.58	0.000	10.18	0.000	13.8
33		Kra30b	30	91420	0.172	68.94	0.130	101.83	0.190	68.67	0.000	72.6
34		Ste36a	36	9526	0.277	48.30	0.000	204.36	3.000	202.17	0.000	138
35	Real-life like	Tai12b	12	39464925	0.000	1.69	0.000	1.03	0.000	1.03	-	-
36		Tai15b	15	51765268	0.004	3.35	0.000	3.05	0.000	3.07	-	-
37		Tai25b	25	344355646	0.552	19.42	5.590	34.52	5.590	34.26	0.000	27.6
38		Tai30b	30	637117113	1.374	26.74	2.220	80.55	1.400	83.15	0.000	78.6
39		Tai35b	35	283315445	1.103	53.34	3.540	186.42	5.080	190.14	0.000	143.4
40		Tai80b	80	818415043	0.717	1147.28	2.790	10532.89	2.610	10942.90	0.006	3494.4
Average					0.330	330.39	0.746	1562.56	0.827	2698.55	0.065	958.2
Proposed SA Average					-	-	0.338	337.02	0.330	330.39	0.347	347.4

[5] J.W. Dickey and J.W. Hopkins, "Campus building arrangement using Topaz," Transportation Research, vol. 6, pp. 59–68, 1972.

[6] A.N. Elshafei, "Hospital layout as a quadratic assignment problem," Operations Research Quarterly, vol. 28, pp. 167–179, 1977.

[7] M.J. Brusco and S. Stahl, "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, 2000.

[8] B. Wess and T. Zeitlhofer, "On the phase coupling problem between data memory layout generation and address pointer assignment," Lecture Notes in Computer Science, vol. 3199, pp. 152–166, 2004.

[9] E. L. Lawler, "The quadratic assignment problem," Manage. Sci., vol. 9, pp. 586–599, Jul. 1963.

[10] A. A. Assad and W. Xu, "On lower bounds for a class of quadratic 0, 1 programs," Oper. Res. Lett., vol. 4, pp. 175–180, Dec. 1985.

[11] R. E. Bedkard, S. E. Karisch, and F. Rendl, "QAPLIB-A Quadratic assignment problem library", http://www.opt.math.tu-graz.ac.at/qaplib

[12] Tabitha James, César Rego, and Fred Glover, "Multistart tabu search and diversification strategies for the quadratic assignment problem," IEEE Transactions on systems, man, and cybernetics-Part A: systems and humans, vol. 39, pp. 579-596, May 2009.

[13] C. Bousono-Calzon and M. R.W. Manning, "The hopfield neural network applied to the quadratic assignment problem," Neural Comput. Appl., vol. 3, pp. 64–72, Jun. 1995.

[14] D. T. Connolly, "An improved annealing scheme for the QAP," Eur. J. Oper. Res., vol. 46, pp. 93–100, May 1990.

 ISBN: 978-1-61208-101-4

[15] M. R. Wilhelm and T. L. Ward, “Solving quadratic assignment problems by simulated annealing,” IIE Trans., vol. 19, pp. 107–119, 1987.

[16] V. Nissen, “Solving the quadratic assignment problem with clues from nature,” IEEE Trans. Neural Netw., vol. 5, pp. 66–72, Jan. 1994.

[17] J. P. Cohoon and W. D. Paris, “Genetic placement,” IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 6, pp. 956–964, Nov. 1987.

[18] D. M. Tate and A. E. Smith, “A genetic approach to the quadratic assignment problem,” Comput. Oper. Res., vol. 22, pp. 73–83, Jan. 1995.

[19] E. Taillard, “Robust taboo search for the quadratic assignment problem,” Parallel Comput., vol. 17, pp. 443–455, 1991.

[20] J. Skorin-Kapov, “Tabu search applied to the quadratic assignment problem,” ORSA J. Comput., vol. 2, pp. 33–45, 1990.

[21] A. Misevicius, “A tabu search algorithm for the quadratic assignment problem”, Comput. Optim. Appl., vol. 30, pp. 95–111, Jan. 2005.

[22] V. Maniezzo and A. Colorni, “The ant system applied to the quadratic assignment problem,” IEEE Trans. Knowl. Data Eng., vol. 11, pp. 769–778, Sep/Oct. 1999.

[23] L. M. Gambardella, E. D. Taillard, and M. Dorigo, “Ant colonies for the quadratic assignment problem,” J. Oper. Res. Soc., vol. 50, pp. 167–176, Feb. 1999.

[24] V. D. Cung, T. Mautor, P. Michelon, and A. Tavares, “Scatter search for the quadratic assignment problem,” in Proc. IEEE Int. Conf. Evol. Comput, pp. 165–169, 1996.

[25] N. Metropolics, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” J. Chem. Phy, vol. 21, pp. 1087-1092, 1953.

[26] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by Simulated Annealing,” SCIENCE, vol. 220, pp. 671-680, May 1983.

[27] V. Cerny, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm,” J. Opt. Theory Appl, vol. 45, pp. 41-51, 1985.

[28] Percy P. C. Yip and Yoh-Han Pao, “Combinatorial a guided evolutionary simulated annealing approach to the quadratic assignment problem", IEEE transactions on systems, man, and cybernetics, vol. 24, September 1994.

[29] S. Andrew, “Parallel N-ary speculative computation of simulated annealing", IEEE transactions on parallel and distributed systems, vol. 6, pp. 997-1005, October 1995.

[30] R. E. Burkard and F. Rendl, "A thermodynamically motivated simulation procedure for combinatorial optimization problems," European journal operational research, vol. 17, pp. 169-174, 1984.

[31] A.S. Ramkumar, S.G. Ponnambalam, N. Jawahar, and R.K. Suresh “Iterated fast local search algorithm for solving quadratic assignment problems,” Robotics and Computer-Integrated Manufacturing, vol. 24 pp. 392–401, 2008.

[32] A.S. Ramkumar, S.G. Ponnambalam, and N. Jawahar “A new iterated fast local search heuristic for solving QAP formulation in facility layout design,” Robotics and Computer-Integrated Manufacturing, vol. 25 pp. 620– 629, 2009.

 ISBN: 978-1-61208-101-4