An evolutionary search heuristic for solving QAP formulation in facility layout design

Conference Paper · October 2007

DOI: 10.1109/CEC.2007.4424993 · Source: IEEE Xplore

3 authors, including:

Ponnambalam S.G.
Universiti Malaysia Pahang

Jawahar N
Thiagarajar College of Engineering

An Evolutionary Search Heuristic for solving QAP Formulation in Facility Layout Design

A. S. Ramkumar, S. G. Ponnambalam, N. Jawahar

***Abstract*: The quadratic assignment problem (QAP) is one of the most challenging combinatorial optimization problems in existence and is known for its diverse applications. In this paper, we propose an evolutionary search heuristic (ESH) with population size equal to two, for solving QAPs and reported its performance on solution quality. The ideas we incorporate in the ESH is iterated self-improvement with evolutionary based perturbation tool, which includes (i) recombination crossover as perturbation tool and (ii) self improvement in mutation operation followed by a local search. Three schemes of ESH are proposed and the obtained solution qualities by the three schemes are compared. We test our algorithm on the benchmark instances of QAPLIB, a well-known library of QAP instances. The performance of proposed recombination crossover with sliding mutation (RCSM) scheme of ESH is well superior to the other two schemes of ESH.**

I. INTRODUCTION

The quadratic assignment formulation in facility layout design is a well studied combinatorial optimization problem in manufacturing planning. Manufacturing companies spend a significant amount of time and money in designing or redesigning the facilities because the design of a facility layout has a tremendous effect on the operation of the system that it houses. A poor facility layout design will cost more and may result in poor system performance as well as customer satisfaction [1]. For the equal sized facility layout problem, the quadratic assignment problem [2] is used to model the problem. Since the QAP is notoriously difficult for exact solution methods [3] even for smaller size problems, i.e. number facilities less than or equal to 30, the development of a good heuristic procedures based on metaheuristics and evolutionary based search procedures are gaining importance among the researchers.

Evolutionary algorithms have been applied to many fields of combinatorial optimization, and it has been shown in augmenting evolutionary algorithm with problem-specific heuristics can lead to highly effective approaches [4]. In this paper we present a new evolutionary algorithms based iterated fast local search procedure for solving QAPs, a well known NP-hard [5] combinatorial optimization problem.

A. S. Ramkumar is with the Department of Mechanical Engineering, Amrita University, India (email: as_ramkumar@ettimadai.amrita.edu).

S. G. Ponnambalam is with the School of Engineering, Monash University, Malaysia Campus, Malaysia (phone: 03-56360600; fax: 03-56329314; email: sgponnambalam@eng.monash.edu.my).

N. Jawahar is with the Department of Mechanical Engineering, Thiagarajar College of Engineering, India (email: jawahartce@yahoo.co.uk).

II. THE QUADRATIC ASSIGNMENT PROBLEM

The QAP of order 'n' consists in looking for the best allocation of 'n' facilities to 'n' known locations, where the terms facility and location should be considered in their most general sense. It was first formulated in [2] and since then it has been recognized as a model of many different real situations; applications have been described concerning planning of buildings in university campuses, arrangement of departments in hospitals, warehouse management and distribution strategies, minimization of the total wire length in electronic circuits, ordering of correlated data in magnetic tapes and others [6]. Mathematically the QAP formulation in facility layout design is defined by two matrices of dimension n x n:

Let,

$D = (d_{ij})$ = the distance from location i to location j

$F = (f_{ij})$ = the flow of materials from facility i to facility j

Matrices D and F are integer-valued matrices. The cost of transferring materials between two facilities can be expressed as the product of the distance between the locations to which the facilities are assigned, $f_{ij}.d_{\pi(i)\pi(j)}$. To solve the QAP one must thus find a permutation Π of the indices (1,2,…,n) which minimizes the total assignment cost:

$$\min_{\pi} \sum_{ij} f_{ij} d_{\pi(i)\pi(j)}$$

In addition to the facility layout design, the QAP arises in many other applications, such as the allocation of plants to candidate locations, backboard wiring problem, design of control panels and typewriter keyboards, turbine balancing and ordering of interrelated data on a magnetic tape etc. The details can be found in [7, 8, and 9]. Exact algorithms for solving QAP include approaches based on (1) branch and bound [10, 11], (ii) cutting planes [12], and (iii) dynamic programming [13]. Among these, the branch and bound algorithms are the most successful, but they are generally unable to solve problems of size larger than n=30. Branch and bound techniques have evolved greatly over the past 40 years [14], starting with Gilmore [15] who solved a QAP of size n=8, to the solution of the nug30, a QAP of size n=30 in 2000 by [3].

III. HEURISTICS FOR QAPS

Heuristics or suboptimal algorithms are often used to estimate solutions for QAP instances. These procedures can produce good answers within reasonable time constraints. There are following categories of heuristics

1-4244-1340-0/07$25.00 ©2007 IEEE

for the QAP: Construction methods, Limited enumeration methods, Improvement methods, Simulated annealing techniques, and Genetic algorithms. Construction methods create suboptimal permutations by starting with a partial permutation which is initially empty. The permutation is expanded by repetitive assignments based on set selection criterion until the permutation is complete.

The CRAFT (Computerized Relative Allocation of Facilities Technique), used for the layout of facilities was first introduced by [16]. Limited enumeration methods are motivated when one expects that an acceptable suboptimal solution can be found early during a brute force enumeration examination. Imposing either a time limit or an iteration limit could terminate such an enumeration methods and these improvement methods are the most researched class of heuristic [8].

The popular methods are iterated local search and the tabu search. Both methods work by starting with an initial basic feasible solution and then attempting to improve it. The local search iteratively seeks a better solution in the neighborhood of the current solution, terminating when no better solution exists within that neighborhood [17]. The tabu search [18, 19] works similarly to the local search; however it is sometimes more favorable since it was designed to overcome the problem of a heuristic getting trapped at local optima. Simulated annealing methods receive it name from the physical process that it imitates. For more details on these methods, see [20, 21].

Genetic algorithms receive their name from an intuitive explanation of the manner in which they behave. This explanation is based on Darwin's theory of natural selection [8]. Genetic algorithms store a set of solutions and then work to replace these solutions with better ones based on some fitness criterion, usually the objective function value. Genetic algorithms are parallel and are helpful when applied in such an environment. Greedy Randomized Adaptive Search Procedure (GRASP) is a relatively new heuristic used to solve combinatorial optimization problems. At each iteration, a solution is computed by randomized search process and the final solution is taken as the one, which is the best after all GRASP iterations are performed.

The GRASP was first applied to the QAP in 1994 by [22]. They applied the GRASP to 88 instances of the QAP, finding the best known solution in almost every case, and improved solutions for a few instances. In recent times, attempts are being made to solve QAP by heuristic approaches. It includes (i) construction methods [23, 24], (ii) limited enumeration methods [25, 6], (iii) greedy randomized adaptive search procedure [22], (iv) simulated annealing [26], (v) tabu search [27, 28], (vi) genetic algorithms [29] –[32]and (vii) ant systems [33, 34].

IV. QAP INSTANCES

It is known from recent research on solving QAPs, the particular type of a QAP instances has a considerable influence on solution quality. According to [35] four classes of QAP have been defined. Class (i), Unstructured, randomly generated instances in which, the distance and flow matrix entries are generated randomly by uniform distribution. Class (ii), Grid-based distance matrix instances in which, the distance matrix stems from a n_1 X n_2 grid and the distances are defined as the Manhattan distance between grid points. Class (iii), Real-life instances in which, the flow matrices have many zero entries and the remaining entries are clearly not uniformly distributed. Class (iv), Real-life like instances in which, the instances are generated in such a way that the matrix entries resemble the distributions found for real-life problems.

V. PROPOSED METHODOLOGY

In this section, the features of ESH are described. ESH is a hybrid algorithm having population size = 2. The number of iterations used for termination criterion governs the complexity of the algorithm. Initial parents are generated randomly. A permutation of the machine size 'n' is mapped into chromosome with the alleles assuming different and non-repeating integer value is the [1, n] interval. The objective function of the permutation π is F (π) to minimize the cost function of the QAP, i.e. min Z = min F (π). The flow chart of the proposed algorithm is shown in Figure 1.

A. Crossover Operation

Though many researchers followed different perturbation tools, an evolutionary operator perturbation tool, recombination crossover (RC) is used in this proposed evolutionary search heuristic algorithm. The recombination operator [36] for the cross over operation preserves the information contained in both parents in the sense that all alleles of the offspring are taken either from the first or from the second parent. The recombination operator works as, all facilities found at the same locations in the two parents are assigned to the corresponding locations in the offspring.

Starting with a randomly chosen location that has no facility assigned yet, a facility is randomly chosen from the two parents. After that, additional assignments are made to ensure that no implicit mutation occurs. Then next non-assigned location to the right is processed in the same way until all locations have been considered. From figure 2, all facilities that are assigned to the same location in the parents are inherited by the offspring.

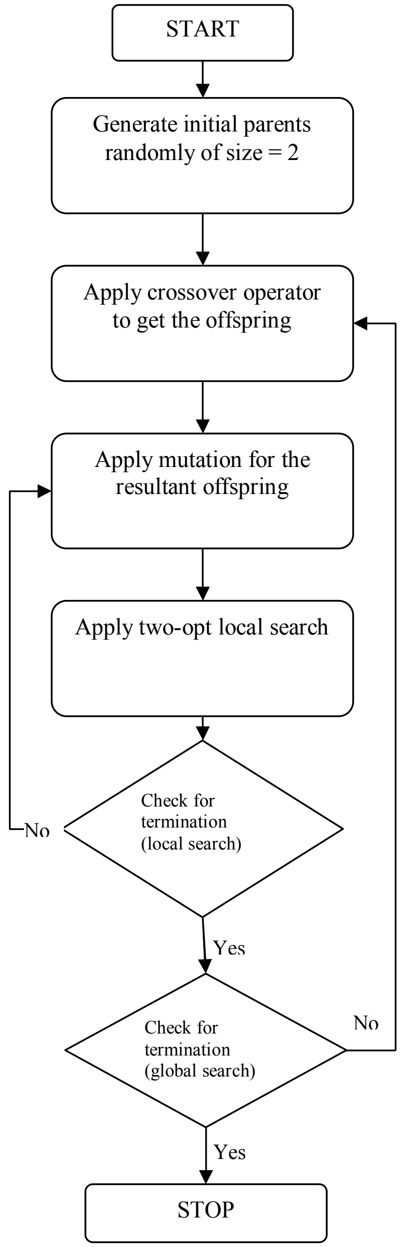

Figure 1. Flow chart of the ESH algorithm.

These are the facilities 5 and 4. Then, beginning with a randomly chosen location, in this case location 4, a facility is randomly selected from one of the parents and is assigned to the same location in the child. In this example facility 3 is assigned to location 4. Now other facilities have to be assigned to guarantee that no implicit mutation occurs. By assigning facility 3 to location 4, we prevent the possibility of assigning facility 2 of parent A and facility 6 of parent B to location 4. Hence we have to assign facility 2 in location 6, free location to the right of randomly selected location 4 already and facility 6 to location 1 by moving right side of the genome. Further we can proceed in choosing a facility for the next free location to the right in the offspring. We have to insert facility 1 to location 2, hence all facilities are assigned, the recombination operator ends the cross over operation. The advantage with this operator is that, it takes minimum distance between individuals of the population during evolution that gives increased performance in the local search.

Parent A: 1	3	5	2	4	6
Parent B: 2	1	5	6	4	3
_	_	5	_	4	_
_	_	5	3	4	_
_	_	5	3	4	2
Offspring: 6	1	5	3	4	2

Figure 2. Recombination crossover operation

B. Mutation Operation

The mutation operation adopted in this paper uses a self-improvement technique, which is explained as follows: This self-improvement technique is the method to improve the solution for predetermined trials of iterations and is developed from local search method and searches the solution space non-systematically until a specific stop criterion is satisfied. In this paper three mechanisms are used for finding the improved neighborhood solutions of offspring π^* obtained from RC operator. The three mechanisms are *randomized Adjacent interchange Mutation (AM), randomized Pair wise interchange Mutation (PM)* and *randomized Sliding Mutation (SM*).

C. Randomized AM

This is a randomized version of adjacent interchange neighborhood structure. This operator will generate a random number R in the range [1, n] and just interchange the machine present in the position R with the next machine in the sequence (R+1) and represented as:

$$\varphi_{am}(2314) = 2\underline{13}4$$

D. Randomized PM

This operator may also be termed as random swap operator and similar to swap neighborhood structure. Two random numbers are generated R_1 and R_2 in the range [1, n] and applied to machines present in the position R_1 and R_2 then the machines according to the random values generated are swapped and represented as:

$$\varphi_{pm}(2314) = \underline{4}31\underline{2}$$

E. Randomized SM

This is a randomized version of inert neighborhood structure. This operator may be also termed as randomized extraction and backward shift insertion operator. Sliding mutation refers to "moving a machine from the j^{th} place and placing it before the i^{th} position". Two values are generated randomly (R_1 and R_2) in the range [1, n] in such a way that $R_1 < R_2$ and applied to machines present in the positions in between R_1 and R_2. The machine in position R_2 is placed before the machine in position R_1 and all the machines in between R_1 and R_2 are pushed one position and represented as:

$$\varphi_{sm}(2314) = 2\underline{4}31$$

The following schemes of the iterated self-improved evolutionary algorithms are proposed in this paper. The scheme RCAM described as recombination crossover and randomized adjacent interchange mutation, RCPM described as recombination crossover and randomized pair wise interchange mutation and third one RCSM was recombination crossover and randomized sliding mutation.

F. Two-opt local search

A critical issue in the design of a neighborhood search approach is the choice of the neighborhood structure that is the manner in which the neighborhood is defined. The local search used in this paper is the two-opt heuristic with first improvement strategy. Let π be a solution (permutation), in two-opt neighborhood N (π) is defined as the set of all possible solutions resulting from π by swapping two of the elements. The two-opt heuristic searches the neighborhood of the current solution for a lower cost solution. If such a solution is found, it replaces the current solution and the search continues.

In our experiments, the entire neighborhood is searched and a solution with the lowest cost is selected. If a local search is completed by performing all possible swaps without improvement, then it will move to next iteration. If any improvement in solution cost is obtained, then it will continue the local search from the first improvement solution. The local search ends at n(n-1)/2 number of swaps. In this paper, the proposed search procedure is, once one iteration of a mutation operator was done, then it will move to a local search (figure: flow chart). If a local search is not improved, then it will move to next iteration of mutation. The number of iteration for local search is also equal to MT_CNT = 50 iteration in the proposed all three schemes of the ESH.

VI. The Proposed Evolutionary Search Heuristic Algorithm

Step 1: Randomly generate two permutations π_1 and π_2. Initialize RC_CNT = 0. Assign π^b = min[F(π_1), F(π_2)]

Step 2: Apply crossover operator among π_1 and π_2. Obtain the offspring π^*.

Step 3: Check for termination. If (RC_CNT = 50) then go to STEP 11

Step 4: Initialize MT_CNT = 0.

Step 5: Apply mutation operator on π^* and get the resultant offspring $\hat{\pi}$.

Step 6: Apply 2-opt local search on $\hat{\pi}$ and get improved offspring $\tilde{\pi}$.

Step 7: If F ($\tilde{\pi}$) < F (π^*) then replace $\pi^* = \tilde{\pi}$.

Step 8: Assign MT_CNT = MT_CNT+1

Step 9: Check for termination. If (MT_CNT = 50) then replace π_1 and π_2 as π^b and $\tilde{\pi}$ respectively, else go to STEP 2.

Step 10: Assign RC_CNT = RC_CNT +1 and then go to STEP 2.

Step 11: The algorithm is terminated.

VII. Experimental Setting

A large number of computational experiments were carried out in order to test the performance of the proposed ESH. The run time of ESH is controlled by termination criteria. The counters used in our proposed all three schemes of ESH algorithms is RC_CNT = 50 iterations and MT_CNT = 50 iterations. The initial population size is equal to two, randomly generated. Unless other wise stated the proposed algorithms is coded in JAVA and run on a Pentium P-IV (XT) machine, 2.8 GHz processor with 1 GB ram.

VIII. Computational Results

To demonstrate the efficiencies of the proposed three schemes of the ESH algorithms, we solved the problem instances from the QAPLIB available wide http://www.opt.math.tu-graz.ac.at/qaplib. The solution qualities were defined as the Relative Percentage Deviation (RPD) from the best known solution from the literature and, calculated according to $\mathbf{[(Z-Z_{best})/Z_{best}]*100}$, where Z is the obtained solution and Z_{best} is the best known solution of the corresponding problem. Table 1 to Table 8 gives the solution qualities (RPD) obtained by three schemes of the proposed ESH with local search and without local search for the four classes of QAP instances. The average relative percentage deviation obtained for each schemes of the ESH for all classes of QAP are also reported.

IX. Conclusion

In this paper, an attempt is made to optimize quadratic assignment formulations in facility layout design. We proposed a evolutionary search heuristic algorithm for solving QAP and presented its performance of solution qualities on four classes of QAP instances in literature. We incorporate three schemes of ESH, (i) RCAM with local search and without local search, (ii) RCPM with local search and without local search and (iii) RCSM with local search and without local search. The principal findings of the paper are, the scheme of RCSM with local search, outperforms the other two schemes of the ESH with in the termination criteria. This shows that, if the tenure of mutation cycle is carefully examined, there is scope for further development in the ESH procedures.

The solution procedure presented here may be extended to solve other formulations in facility layout problems of different manufacturing environments.

TABLE 1
Solution qualities obtained by ESH for Class (i) type of QAPs

Problem Name	ESH without local search		
	RCAM	RCPM	RCSM
Tai20a	15.404	15.174	15.028
Tai25a	12.311	13.756	11.431
Tai30a	13.327	13.518	14.035
Tai35a	12.898	13.200	14.911
Tai40a	12.788	14.060	14.122
Tai60a	14.129	13.395	13.353
Tai80a	11.773	11.973	12.852
Rou20	11.309	10.960	13.842
Ave. RPD	12.992	13.255	13.697

TABLE 2
Solution qualities obtained by ESH for Class (i) type of QAPs

Problem Name	ESH with local search		
	RCAM	RCPM	RCSM
Tai20a	7.047	2.221	0.304
Tai25a	2.925	3.459	1.184
Tai30a	3.808	4.198	1.509
Tai35a	4.197	3.514	1.717
Tai40a	4.185	3.086	1.871
Tai60a	4.043	3.525	1.611
Tai80a	3.234	3.403	2.158
Rou20	1.439	1.732	0.000
Ave. RPD	3.860	3.142	1.294

TABLE 3
Solution qualities obtained by ESH for Class (ii) type of QAPs

Problem Name	ESH without local search		
	RCAM	RCPM	RCSM
Nug30	20.020	18.125	18.909
Tho30	15.048	20349	20.383
Tho40	21.602	23.465	25.151
Sko42	17.784	15.785	14.584
Sko49	15.129	14.530	17.763
Sko56	16.751	14.238	14.714
Sko64	15.440	12.211	15.324
Sko72	15.422	13.146	15.202
Ave. RPD	17.149	16.481	17.754

TABLE 4
Solution qualities obtained by ESH for Class (ii) type of QAPs

Problem Name	ESH with local search		
	RCAM	RCPM	RCSM
Nug30	1.502	1.666	0.359
Tho30	5.145	3.875	0.345
Tho40	2.932	2.655	0.986
Sko42	3.289	1.670	0.493
Sko49	2.899	2.335	0.145
Sko56	1.811	2.879	0.267
Sko64	1.047	0.685	0.169
Sko72	1.808	1.600	0.537
Ave. RPD	2.554	2.170	0.413

TABLE 5
Solution qualities obtained by ESH for Class (iii) type of QAPs

Problem Name	ESH without local search		
	RCAM	RCPM	RCSM
Bur26a-g	4.729	2.629	3.337
Chr25a	159.536	177.608	250.474
Els19	71.470	41.085	68.380
Kra30a	29.201	25.861	26.963
Kra30b	30.431	27.357	33.516
Ste36a	67.164	57.359	69.137
Ste36b	159.198	156.472	236.639
Ave. RPD	74.533	69.767	98.349

TABLE 6
Solution qualities obtained by ESH for Class (iii) type of QAPs

Problem Name	ESH with local search		
	RCAM	RCPM	RCSM
Bur26a-g	0.147	0.133	0.032
Chr25a	47.734	31.085	2.055
Els19	14.445	0.000	0.000
Kra30a	7.244	5.006	0.000
Kra30b	3.544	2.778	0.186
Ste36a	5.186	4.703	3.002
Ste36b	20.994	13.348	4.656
Ave. RPD	14.185	8.151	1.0419

TABLE 7
Solution qualities obtained by ESH for Class (iv) type of QAPs

Problem Name	ESH without local search		
	RCAM	RCPM	RCSM
Tai20b	47.688	31.024	53.792
Tai25b	71.054	49.180	47.939
Tai30b	56.705	30.245	39.702
Tai35b	48.013	35.519	35.408
Tai40b	48.939	36.930	40.391
Tai50b	50.864	43.505	42.757
Tai60b	52.794	35.114	40.046
Tai80b	43.670	31.570	37.023
Tai100b	42.709	30.804	34.620
Ave. RPD	51.382	35.988	41.298

TABLE 8
Solution qualities obtained by ESH for Class (iv) type of QAPs

Problem Name	ESH with local search		
	RCAM	RCPM	RCSM
Tai20b	3.460	1.436	0.000
Tai25b	6.412	6.155	5.592
Tai30b	16.631	16.468	1.400
Tai35b	7.701	8.481	5.084
Tai40b	6.974	6.898	12.464
Tai50b	4.835	6.422	0.656
Tai60b	11.844	10.555	3.861
Tai80b	1.849	4.013	2.606
Tai100b	7.875	5.397	3.362
Ave. RPD	7.509	7.314	3.892

REFERENCES

[1] Heragu, S.S. (1997), "*Facilities Design*", Boston, MA: PWS Publishing.

[2] Koopmans, T.C., and M. Beckman (1957), "Assignment problems and the location of economic activities", *Econometrica* 25, 53-76.

[3] Anstreicher, K.M. (2003), "Recent advances in the solution of quadratic assignment problems", Mathematical Programming, Series B vol.97, pp. 27-42.

[4] Merz P and B. Freisleben (2000), "Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem", IEEE Transactions on Evolutionary Computation, vol. 4, no. 4, pp. 337-352.

[5] Garey M.R. and D.S. Johnson (1979), "*Computers and Intractability: A Guide to the Theory of NP-Completeness",* W.H. Freeman Company.

[6] Burgard, R.E., and T. Bonniger, (1983), "A Heuristic for Quadratic Boolean Programs with Applications to Quadratic Assignment Problems", *European Journal of Operational Research*, vol, 13, pp. 374-386

[7] Malucelli, F. (1993), "Quadratic Assignment Problems: Solution Methods and Applications", unpublished Doctoral Dissertation, Departmento Informatica, Universita Di Pisa, Italy.

[8] Pardalos, P.M., F. Rendl, and H. Wolkowicz (1994), "The Quadratic Assignment Problem: A Survey and Recent Developments, in Quadratic Assignment and Related Problems", P.M. Pardalos and H. Wolkowicz, eds., *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 16, 1-42, AMS.

[9] Burkard, R.E., E. Cela, P.M. Pardalos, and L.S. Pitsoulis. (1998), "The quadratic assignment problem. In handbook of combinatorial optimization 3, Edited by D.Z. Zhu and P.M. Pardalos, Kluwer, 241-337.

[10] Lawler, E.L. (1963), "The Quadratic Assignment Problem", *Management Science* 9, 586-599.

[11] Pardalos, P. M., and J. Crouse. (1989), "A parallel algorithm for the quadratic assignment problem", *Proceedings of the Supercomputing 1989 Conference*, ACM Press, pp. 351-360.

[12] Bazara, M.S., and M.D. Sherali (1980), "Bender's Partitioning Scheme applied to a new Formulation of the Quadratic Assignment Problem", *Naval Research Logistics Quarterly*, vol. 27, pp. 29-41.

[13] Christofides, N., and E. Benavent. (1989), "An exact algorithm for the quadratic assignment problem", *Operations Research* 37, 760-768

[14] Carraresi, P. and F. Malucelli (1992), "A New Lower Bound for the Quadratic Assignment Problem", Technical report TR-34/92, Universita di Pisa.

[15] Gilmore, P.C. (1962), "Optimal and suboptimal algorithms for the quadratic assignment problem", SIAM Journal on Applied Mathematics 10, 305-313.

[16] Armour, G.C. and E.S. Buffa (1963), "Heuristic Algorithm and Simulation Approach to Relative Location of Facilities", *Management Science* 9, 294-309.

[17] Pitsoulis, L. (1994), "A sparse GRASP for solving the Quadratic Assignment Problem", M. Eng. Thesis, The University of Florida.

[18] Glover, F. (1989), "Tabu Search - Part 1", *ORSA Journal on Computing* 1, no. 3, 190-206.

[19] Glover F. (1989), "Tabu Search - Part 2", *ORSA Journal on Computing* 2, no. 1, 4-32.

[20] Connolly D T (1990), "An improved annealing scheme for the QAP", *European Journal of Opreational Research* 46, 93–100.

[21] Cerny, V. (1985), "Thermodynamical Approach to Traveling Salesman Problem: An Efficient Simulation Algorithm", Journal of Optimization Theory and Applications, 45, pp. 41-51.

[22] Li, Y., P.M. Pardalos, and M.G.C. Resende (1994), "A greedy randomized adaptive search procedure for the quadratic assignment problem. In quadratic assignment and related problems, edited by P.M. Pardalos and H. Wolkowicz, DIMACS Series in Ciscrete Mathematics and Theoritical Computer Science, American Mathematical Society, 237-261.

[23] Buffa, E. S., G. C. Armour, and T. E. Vollmann. 1964., "Allocating facilities with CRAFT", *Harvard Business Review* 42, 136-158.

[24] Muller-Merbach, H. (1970), "Optimale Reihenfolgen", Springer Verlag, 158-179.

[25] West, D.H. (1983), "Algorithm 608: Approximate Solution of the Quadratic Assignment Problem", ACM Transactions on Mathematical Software, vol.9, pp. 461-466.

[26] Wilhelm, M.R. and T.L. Ward (1987), "Solving Quadratic Assignment Problems by Simulated Annealing", IIE Transactions, vol. 19, pp. 107-119.
[27] Skorin-Kapov, J. (1990), "Tabu Search Applied to the Quadratic Assignment Problem", ORSA Journal of Computing, vol. 2, pp. 33-45.
[28] Taillard É D (1991), "Robust taboo search for the quadratic assignment problem", *Parallel Computing* **17**: 443–455.
[29] Fleurent C and Ferland J (1994), "Genetic hybrids for the quadratic assignment problem", *DIMACS Series in Mathematics and Theoretical Computer Science **16**: 190–206.*
[30] Tate., D.M., A.E. Smith (1995), "A Genetic Approach to the Quadratic Assignment Problem", Computers and Operations Research, vol. 22, (1), pp. 78-83.
[31] Ahuja, K.R., J.B. Orlin, and A, Tiwari (2000), "A Greedy Algorithm for the Quadratic Assignment Problem", *Computers & Operational Research*, vol.27, pp. 917-934.
[32] Misevicius, A (2004), "An improved hybrid genetic algorithm: new results for the quadratic assignment problem", *Knowledge-Based Systems,* 17, pp. 65-73.
[33] Maniezzo, V., M. Dorigo, and A. Colorini (1994), "The Ant System Applied to the Quadratic Assignment Problem", Technical Report IRIDIA/94-28, Universite Libre de Bruxelles, Belgium.
[34] Dorigo M, Maniezzo V and Colorni A (1996), "The ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics–Part B* 26, *29–41*.
[35] Taillard É D (1995), "Comparison of iterative searches for the quadratic assignment problem", *Location Science* **3**: 87–105.
[36] Merz P and B. Freisleben (1999), "A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem", in Proceedings of the International Congress on Evolutionary Computation, IEEE press, pp 2063-2070.