Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 4, Issue, 02, pp.216-221, February, 2012

INTERNATIONAL JOURNAL
OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

SOLVING QUADRATIC ASSIGNMENT PROBLEMS USING A HYBRID NATURE-INSPIRED TECHNIQUE

Hashni, T*. and Amudha, T.

Department of Computer Application, Bharathiar University, Coimbatore, Thiruchirapalli, Tamilnadu, India

ARTICLE INFO

Article History:
Received 27th November, 2011
Received in revised form
15th December, 2011
Accepted 19th January, 2011
Published online 29th February, 2012

Key words:
Nature- Inspired algorithms, Consultant Guided Search algorithm, Genetic algorithm, Consultant Guided Search - Genetic algorithm.

ABSTRACT

Nature provides motivation to scientists in many ways. Scientists have started to realize that nature is a great source of inspiration to develop intelligent systems and techniques. Nature- Inspired algorithms is a kind of algorithms that imitate the problem solving behavior from nature. Consultant Guided Search algorithm (CGS) and Genetic algorithm (GA) are some of the Nature-Inspired Metaheuristic Algorithms inspired from Nature. In this paper, Consultant Guided Search algorithm (CGS) was hybridized with Genetic algorithm (GA) and a new technique was proposed. The proposed Consultant Guided Search - Genetic algorithm (CGS-GA) was implemented to solve the benchmark instances of Quadratic Assignment Problem (QAP). The performance of the proposed CGS-GA was compared with CGS algorithm. Results have shown that the proposed CGS-GA has outperformed CGS in arriving at improved optimal solutions for various test instances of Quadratic Assignment Problem.

INTRODUCTION

Nature has inspired researchers in many different ways. Nature-Inspired Algorithms is a category of algorithms that imitate the way nature performs. Nature-inspired metaheuristic algorithms are characterized by algorithmic operators mimicking computationally useful aspects of various natural phenomena [5][1]. Metaheuristics are algorithmic templates used to specify problem-independent optimization strategies, which can be instantiated in order to define problem-specific heuristics. Some of the most successful metaheuristic conceived in the last two decades are swarm intelligence techniques like Ant Colony Optimization (ACO) [2][3], Bee Colony Optimization (BCO) [4][12]. They are population-based methods that make use of the global behavior that emerges from the local interaction of individuals with one another and with their environment [9][17]. Consultant-Guided Search (CGS), is a new population-based metaheuristic for solving combinatorial optimization problems. CGS takes inspiration from the way people make decisions based on advice received from consultants. Human behavior is complex, but Consultant-Guided Search uses virtual people that follow only simple rules [9]. Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms (GA) belong to the larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by nature. The Quadratic Assignment Problem (QAP) was introduced by Koopmans and Beckmann back in 1957 to model a plant location problem [18].It is the most difficult NP-hard combinatorial optimization problem. It is mainly to find an assignment (n! Possible Permutations) of all facilities to all locations, such that the total cost of the assignment is minimized [6][19].

The Objective of the Quadratic Assignment Problem:

$$\min_{p \in N} f(P) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{p(i)p(j)}$$

Where, ***F= (fij)*** be the flow matrix, whose ***(i,j)-th*** element represents the flow between facilities ***i*** and ***j*** .

D = (dij) be distance matrix, whose ***(i,j)-th*** element represents the distance between locations i and j.

P = the total number of permutations or arrangements for assigning facilities to locations (***n!*** possible assignments) [15] [24].

In this paper, Consultant Guided Search (CGS) algorithm was hybridized with Genetic algorithm (GA) and a new algorithm Consultant Guided Search–Genetic algorithm (CGS-GA) was proposed. Both CGS and CGS-GA were applied to the benchmark instances of QAP.The results obtained from the proposed CGS-GA algorithm and the existing CGS algorithm were compared and analyzed. This paper is organized as follows: Section 2 describes the objectives of this paper and the methodology for solving QAP using CGS and proposed CGS-GA. Section 3 presents the implementation results and analyzes the results obtained by CGS and proposed CGS-GA

***Corresponding author:** *hashnithirumoorthy@gmail.com*, *amudhaswamynathan@buc.edu.in*,

for various QAP benchmark instances. Section 4 concludes the whole paper and presents future research directions.

Objectives and Methodology

The main objectives of this paper are:

- To propose a new, modified CGS by applying Genetic algorithm (GA) as consultant search technique (CGS-GA).
- To analyze and compare the performance of the proposed Consultant Guided Search-Genetic Algorithm (CGS-GA) and CGS algorithm in solving QAP.

Consultant Guided Search (CGS) algorithm

CGS is a new metaheuristic algorithm for solving combinatorial optimization problems. CGS takes inspiration from the way people make decisions based on suggestion received from consultants. In CGS Virtual persons represent as agents, which collaboratively solve complex combinatorial optimization problems. It is a population-based method. An individual of the CGS population is a virtual person, which can simultaneously act both as a client and as a consultant [10][11]. At the beginning of each iteration, a client chooses a consultant based on its personal preference and on the consultant's reputation. The reputation of a consultant increases with the number of successes achieved by its clients. A client achieves a success, if it constructs a solution better than all solutions found until that point by any client guided by the same consultant. Each time a client achieves a success, the consultant adjusts its strategy in order to reflect the sequence of decisions taken by the client. The exact details of how reputation and personal preference are used in order to select a consultant are specific to each application of CGS to a particular class of problems. Because the reputation fades over time, a consultant needs that its clients constantly achieve successes, in order to keep its reputation. If the consultant's reputation sinks below a minimum value, it will take a sabbatical leave, during which it will stop offering advice to clients and it will instead start searching for a new strategy to use in the future[10][11].

Pseudo code of Consultant Guided Search algorithm

```
procedure CGSMetaheuristic ( )
  create the set P of virtual persons
  foreach p ∈ P do
      setSabbaticalMode (p)
  end foreach
  while (termination condition not met) do
    foreach p ∈ P do
        if  actionMode[p] = sabbatical then
            currStrategy[p] ← constructStrategy (p)
        else
            currCons[p] ← chooseConsultant (p)
            if currCons[p] ≠ null then
                currSol[p] ← constructSolution (p, currCons[p])
            end if
        end if
    end foreach
    applyLocalOptimization ( )                    // optional
    foreach p ∈ P do
      if actionMode[p] = sabbatical then
          if currStrategy[p] better than bestStrategy[p] then
          bestStrategy[p] ← currStrategy[p]
      end if
    else
      c ← currCons[p]
      if c ≠ null and currSol[p] is better than all solutions
          found by a client of c since last sabbatical then
              successCount[c] ← successCount[c] + 1
              strategy[c] ← adjustStrategy(c, currSol[p])
          end if
      end if
    end foreach
        updateReputations ( )
        updateActionModes ( )
  end while
end procedure
```

Genetic Algorithm (GA)

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem [14].Genetic algorithms were formally introduced in the United States in the 1970s by John Holland at University of Michigan. The continuing price/performance improvements of computational systems have made them attractive for some types of optimization. In particular, genetic algorithms work very well on mixed (continuous and discrete), combinatorial problems. They are less susceptible to getting 'stuck' at local optima than gradient search methods [13][20]. Genetic algorithms has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solve problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs [7][8]. However, because of its outstanding performance in optimization, GAs has been wrongly regarded as a function optimizer [16]. Genetic algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce [21][22]. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

The genetic algorithm process consists of the following steps:

Selection : Selecting two parent chromosomes.

Crossover : Crossover the parent chromosomes to form new offspring.

Mutation : Mutate new offspring

Pseudo code of Genetic Algorithm

*1. [**Start**] Generate random population of n chromosomes (suitable solutions for the problem)*
*2.[**Fitness**] Evaluate the fitness f(x) of each chromosome x in the population*
*3.[**New population**] Create a new population by repeating following steps until the new population is complete*
> *1.[**Selection**] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)*
> *2.[**Crossover**] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.*
> *3.[**Mutation**] With a mutation probability mutate new offspring at each locus (position in chromosome).*
> *[Accepting] Place new offspring in a new population*

*4.[**Replace**] Use new generated population for a further run of algorithm*
*5.[**Test**] If the end condition is satisfied, stop, and return the best solution in current population*
*6.[**Loop**] Go to step 2*

Proposed CGS-GA Technique

The foremost aim of the proposed CGS-GA algorithm is to discover the feasible assignment cost for the given set of "n" facilities and the set of "n" locations. In the Quadratic Assignment Problem, there are n! possible assignments based on size of facilities and locations. To measure the cost of each possible assignment, the prescribed flow between each pair of facilities will be multiplied with the distance between their assigned locations, and sum over all the pairs. The main objective is to find the assignment of all facilities to all locations, such that the total cost of the assignment is to minimize:

$$\underset{p\in N}{Min}\ f(P) = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\cdot d_{p(i)p(j)} \longrightarrow \quad (1)$$

Where, "n" indicate set of the facilities and set of the locations, a flow matrix F= {fij}, whose (i,j)-th elements represents the flow between facilities i and j, a distance matrix D= {dij},(i,j)=th elements represents the distance between locations i and j,P indicates the total number of possible assignments for the facilities to the locations.

In this proposed CGS-GA algorithm, at the beginning of each iteration a client chooses consultants that will advice it during the construction of the current feasible assignment cost. A client decides which consultant to choose for construction based on consultant's reputation. The reputation of a consultant depends on the number of successes achieved by its clients. A clients achieves a success, if it constructs an assignment cost that is better than all assignment cost found up to that point by any client guided by the same consultant. In this case, the consultant's reputation will be incremented and they will also receive a supplementary bonus. On the other hand consultant's reputation fades over at each iteration time. The consultant's reputation fades as given by formula (2.2):

$$\boldsymbol{Reputation_k} = \boldsymbol{Reputation_{k\ (1-r)}} \longrightarrow \quad (2)$$

where, the parameter "r" represents the reputation fading rates," k" indicates which consultant reputation wants to fade over.

To keep its reputation, a consultant needs that its clients constantly achieve success. For each consultant, the algorithm keeps track of the best assignment cost obtained by any client working under its guidance. Based on this best assignment of all facilities to all locations, the proposed CGS-GA maintains a ranking of the consultants. For a small number of consultants appearing at the top of this ranking, the algorithm prevents their reputations from sinking below by initial reputation value. Besides reputation, another factor contributing to the choice of a consultant is clients' personal preference. In this proposed CGS-GA, the personal preference for a consultant is determined by the cost of its advertised assignment. Together with reputations, it gives the suitability of consultants by formula (2.3):

$$\boldsymbol{Suitability_k} = \frac{\boldsymbol{Reputation_k}}{\dfrac{\boldsymbol{\beta + Cost_{k-}\ Cost_{bsf}}}{\boldsymbol{Cost_{bsf}}}} \longrightarrow \quad (3)$$

where, the parameter "β" indicates the influence of the personal preference, "$\boldsymbol{Cost_K}$" is the cost of the assignment advertised by consultant k, "$\boldsymbol{Cost_{bsf}}$" is the cost of the best so far assignment. The probability to choose consultant k is:

$$P_K = \frac{\boldsymbol{Suitability_k}}{\sum_{c\ \in C} \boldsymbol{Suitability_k}} \longrightarrow \quad (4)$$

where "c" is the set of all available consultants. A Client is allowed to choose itself as a consultant, because the probabilities given by formula (2.4) do not depend on the client making the choice. In this proposed CGS-GA algorithm, each consultant used genetic algorithm as search technique (GA) to provide feasible assignment for its client. In this proposed CGS-GA, the genetic algorithm search the possible assignment for all facilities to all location based on its three operators namely:

- Selection
- Crossover
- Mutation

Selection phase is the foremost process in genetic algorithm. In this proposed CGS-GA algorithm, selection stage of genetic algorithm selects the two random possible assignments as a parent for the given set of "n" facilities and the set of "n" locations and store it for mating process. After the selection procedure of genetic algorithm in CGS-GA, the selected parent1 possible assignment and parent 2 possible assignments are placed in mating pool for crossover process. The crossover process is the next stage of selection phase. In this proposed CGS-GA algorithm, the two parent possible assignments in selection phase (i.e. Chromosomes) are made to perform a single-point crossover. Single crossover points on both parents' possible assignments are selected. All data beyond that point in either assignment are swapped between the two parent assignments. The resulting assignments are the children. The work adopts the Nug12 as an example and the Fig 1 presents the two possible assignments from the instances Nug12 and its crossover process.

Crossover point

Parent 1	3	8	5	6	10	7	11	9	1	4	2	12
Parent 2	6	12	10	7	8	1	4	9	3	11	5	2
Offspring 1	6	12	10	7	8	1	11	9	1	4	2	12
Offspring 2	3	8	5	6	10	7	4	9	3	11	5	2

Fig: 1 Single-Point Crossover in proposed CGS-GA

In above process two parents possible assignments are randomly selected from population for crossover phase and generated single cross point to exchange two substrings, which defined by crossover point, between parents parent1 and parent2 to product offspring1 and offspring2.But after the crossover phase in this proposed CGS-GA, some repetition of gene are presented in Offspring1 and Offspring2. To avoid that repetition, exchange the genes which are already in the substring from the parent 1 and parent 2.The mutation is the next stage of crossover. In this proposed CGS-GA, swap mutation is implemented. The swap mutation randomly selects two positions in off springs and swaps those two values of its position.

After searching the possible assignment using search technique (GA), the consultant will compute the cost for that assignment to its client. If the consultants' reputation is below a minimum value, then that particular consultant will be place in sabbatical mode, during that time that particular consultant will stop using search technique (GA). After a predefined period of time, the sabbatical mode ends and the virtual persons are set to normal mode and it guide regarding the location to its clients by search technique (GA). At the end of the sabbatical, the consultants' reputation is reset to predefined value. At each step of iteration in this proposed CGS-GA algorithm, a client receives from the consultant chosen for this iteration a suggestion regarding the next action to be taken. Solution construction is a stochastic process in proposed CGS-GA; therefore the client will not always follow the consultant's recommendation. Usually, at each step there are several variants the client can choose from. The variant recommended by the consultant has a higher probability to be chosen, but the client may option for one of the other variants, which it selects based on heuristic. The following Table 2.1 depicts the parameter settings for proposed CGS-GA:

Table 1: Parameter settings for proposed CGS-GA

Parameter	Values	Description
P	5	Number of Virtual persons
β	0.002	Influence of the advertised cost
maxReputation	40	Maximum reputation value
initialReputation	15	Reputation after sabbatical
bonus	6	Best-so-far reputation bonus
r	0.1	Reputation fading rate
sabbaticalduration	5	Sabbatical leave count
Generations	1000	Number of Generations in GA

Test Problems of QAP

The test problems are taken from the QAPLIB-Library [23]. QAPLIB-Library is a collection of test data sets for quadratic assignment problems.QAP has remained one of the great challenges in combinatorial optimization. It is still considered a computationally nontrivial task to solve modest size problems, say of size n=25. The benchmark Instances of Quadratic Assignment Problem are classified based on authors names with instances size. The instances size starts from 12 to 100.The format of the problem data in QAPLIB-Library [23]:

n , A , B

where n is the size of the instance A and B are either flow or distance matrix. This corresponds to a QAP of the form

$$\min_{p} \sum_{i} \sum_{j} a_{ij} \, b_{p(i),p(j)}$$

where p is a permutation. (i.e. possible assignments of set of "n" facilities and set of "n" Locations).

Implementation Results and Discussion

Chr12c, Scr12, Rou12, Tai12a, Had12, Nug12, Chr12a, Chr12b, Esc16a, Esc16b, Esc16c, Esc16e, Esc16d, Nug16a, Nug16b, Had12, Nug30, Tai30a and Tai30b benchmark QAP Problems were solved in this research work. The proposed CGS-GA algorithm gave feasible solution for all the runs. The result obtained by proposed CGS-GA algorithm was compared with CGS algorithm. The CGS-GA algorithm gave a minimum cost when compared with the cost obtained by CGS algorithm. For instances, Esc16b, Esc16c, Esc16e, Nug16a the proposed CGS-GA algorithm gave best-known solution. The CGS Gap and CGS-GA gap are identified to calculate the difference between best solution and the best-known value. The gap value is the relative gap between best feasible solutions, and best known lower bound in percent:

$$\textit{Gap} = (\textit{solution} - \textit{bound}) / (\textit{solution}) * 100\%. \longrightarrow \quad (3)$$

where "solution" indicates best solution of Instance, "Bound" indicates best-known solution of Instance. The computational results are given below in Table 2, 3, 4 and Figure 2, 3, 4.

Table 2: **Optimal cost and gap obtained for 12 size Instances**

Instance Size	Optimal Cost		Best-Known Solution	CGS Gap (%)	CGS-GA Gap (%)
	CGS	CGS-GA			
Chr12c	7098	6035	5578	0.21	0.07
Scr12	17565	16030	15705	0.1	0.02
Rou12	134853	125862	117764	0.06	0.06
Tai12a	128239	126392	112208	0.12	0.11
Had12	865	840	826	0.04	0.01
Nug12	335	307	289	0.13	0.05
Chr12a	7704	6213	4776	0.38	0.23
Chr12b	11907	7150	4871	0.59	0.31

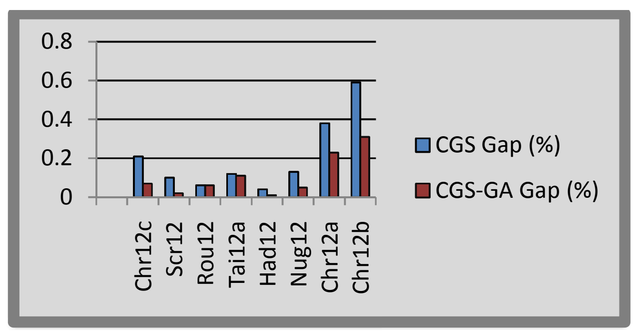


Fig 2: Comparison graph of Gap obtained by 12 size Instances

Table 3: Optimal cost and gap obtained for 16 size Instances

Instance Size	Optimal Cost		Best-Known Solution	CGS Gap (%)	CGS-GA Gap (%)
	CGS	CGS-GA			
Esc16a	43	37	34	0.20	0.08
Esc16b	154	146	146	0.05	0.00
Esc16c	92	80	80	0.13	0.00
Esc16d	24	15	14	0.41	0.06
Esc16e	19	8	8	0.57	0.00
Nug16a	898	805	805	0.10	0.00
Nug16b	755	695	620	0.17	0.1
Had16	1991	1937	1860	0.06	0.03

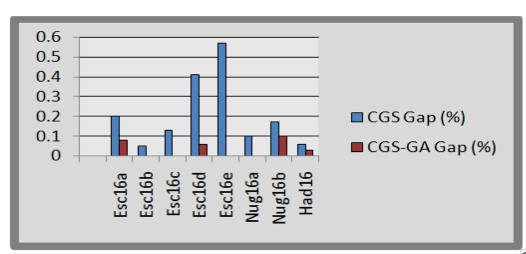


Fig 3: Comparison graph of Gap obtained by 16 size Instances

Table 4: Optimal cost and gap obtained for 30 sizes Instances

Instance Size	Optimal Cost		Best-Known Solution	CGS Gap (%)	CGS-GA Gap (%)
	CGS	CGS-GA			
Nug30	3361	3101	3062	0.08	0.01
Tai30a	1773124	1650186	909073	0.48	0.44
Tai30b	742123101	524582483	318558556	0.57	0.39

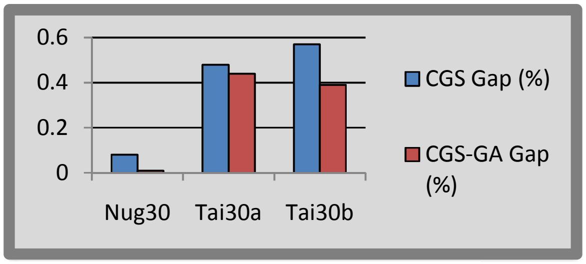


Fig 4: Comparison graph of Gap obtained by 30 size Instances

From the implementation of this research work, it was observed that the proposed CGS-GA was very effective than CGS in solving Quadratic Assignment problem. The proposed CGS-GA can also be used to much higher instances of size ranging from 35 to 100 and this hybrid algorithm will surely be able to achieve the best possible solutions, as it was quite evident from the implementation results. CGS-GA algorithm can also be easily and effectively modified to make it suitable for solving various other types of combinatorial optimization problems.

Conclusion and Future work

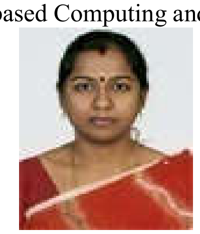
In this paper, the Consultant Guided Search algorithm is hybridized with Genetic Algorithm and a new technique was proposed for solving the benchmark instances of QAP. The ability of the proposed CGS-GA was investigated through the performance of several runs on well-known test problems. The results obtained by the proposed CGS-GA for Quadratic Assignment Problem are much better when compared to the result obtained by CGS algorithm. CGS-GA algorithm can be modified and be made suitable for other types of combinatorial optimization problems such as Vehicle routing, Scheduling , Travelling Salesman Problem, Bin packing etc., CGS-GA can be improved by making each consultant to opt for a different search technique. This work can be extended by designing and implementing different types of crossover and mutation operators to improve CGS-GA performances and also to implement for large size Instances.

REFERENCES

1. Batista B.M., José A., Pérez M. and Vega J.M.M., Nature-inspired Decentralized Cooperative Metaheuristic Strategies for Logistic Problems, 2006.
2. Dorigo, M., St¨utzle, T., Ant Colony Optimization, MIT Press, Cambridge (2004).
3. Dorigoa M., Blumb C., Ant Colony Optimization Theory: A Survey, Theoretical Computer Science, 2005.
4. Dusan Teodorovic., Bee Colony Optimization(BCO), Innovations in Swarm Intelligence
5. Elisa Valentina Onet and Ecaterina Vladu., Nature Inspired Algorithms and Artificial Intelligence, Journal of Computer Science, 2005.
6. Elshafei.A.N., Hospital Layout as a Quadratic Assignment Problem, Operations Research Quarterly 28, 1977, 167–179.
7. Farhad Djannaty, Hossein Almasi.A., Multi Hybrid Genetic Algorithm for the Quadratic Assignment Problem, Intelligent Systems Scientific Society of Iran, 2007.
8. Goldberg D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison Wesley, 1990
9. Iordache.S., Consultant-Guided Search - A New Metaheuristic for Combinatorial Optimization Problems, In Proceedings of the 2010 Genetic and Evolutionary Computation Conference (GECCO 2010). ACM Press, New York (2010).
10. Iordache.S., Consultant-Guided Search Combined With Local Search For The Quadratic Assignment Problem, GECCO (Companion), Springer-Verlag Berlin Heidelberg, 2010.
11. Iordache.S., Consultant-Guided Search Combined With Local Search For The Traveling Salesman Problem, GECCO (Companion), Springer-Verlag Berlin Heidelberg, 2010.
12. Karl von Frisch., The Dance Language and Orientation of Bees, Harvard University Press, Cambridge, Massachusetts, 1967.
13. Kaur B. and Mittal U., Optimization of TSP using Genetic Algorithm, Advances in Computational Sciences and Technology, 2010.
14. Moscato, P., On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program (report 826), 1989.
15. Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl., Qaplib-a quadratic assignment problem library. European Journal of Operational Research, 55:115–119,1991
16. Ravindra K. Ahuja, James B. Orlin, Ashish Tiwari., A Greedy Genetic Algorithm for the Quadratic Assignment Problem, European Journal of Operations Research 13, 374-386,1997.
17. Shigeyoshi Tsutsui., Parallel Ant Colony Optimization for the Quadratic Assignment Problems with Symmetric Multi Processing, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL) Report No. 2008006, 2008Studies in Computational Intelligence, Volume 248/2009, pg.39-60,2009.
18. Stutzle T., Max-Min Ant System for the Quadratic Assignment Problem, Sophia-Antipolis, France, July, 1997.
19. Tjalling C. Koopmans and Martin J. Beckmann., Assignment problems and the location of economic activities,Cowles Foundation Discussion Papers 4, Cowles Foundation, Yale University, 1957.
20. http://en.wikipedia.org/wiki/Genetic_algorithm
21. http://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)
22. http://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)
23. http://www.opt.math.tugraz.at/qaplib/
24. http://en.wikipedia.org/wiki/Quadratic_assignment_problem

AUTHORS

Ms.T.Hashni received her B.Sc degree in Computer science, M.Sc degree in Information Technology, M.Phil in Computer Science in 2008, 2010 and 2011respectively, from Bharathiar University, Coimbatore, India. She awarded as Rank Holder in Bharathiar University for her Post Graduation. She has attended National Conferences. Her area of interest includes Distributed Computing; Agent based Computing and Nature-inspired computing.



Mrs.T.Amudha received her B.Sc Degree in Physics, Masters Degree (MCA) in Computer Applications and M.Phil in Computer Science in 1995, 1999, and 2003 respectively, from Bharathidasan University, India. She has qualified UGC-NET for Lectureship in 2003 and is currently pursuing her doctoral research at Bharathiar University in the area of Agent Systems. She is currently working as Asst. Professor in the Department of Computer Applications, Bharathiar University, and Coimbatore and has 12 years of academic experience. She has more than 20 research publications for her credit in International/ National Journals & Conferences. Her area of interest includes Software Agents, Bio-inspired computing and Grid computing.