ISSN:0975-9646

Santosh Kumar Sahu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5950-5956

Hybrid Ant System Algorithm for Solving Quadratic Assignment Problems

Santosh Kumar Sahu, Manish Pandey

Department of Computer Science and Engineering

Maulana Azad National Institute of Technology, Bhopal, India

Abstract--**In this paper a hybrid variant of meta-heuristic algorithm ant colony optimization (ACO) is used. Approximate solutions to quadratic assignment problem have been proved very efficient. Different variants of ant colony optimization have been applied to QAP. But in this paper a hybrid approach is proposed which is combination of Ant system and Max-Min Ant system to take benefits of both the methods. In this approach solution construction phase is in accordance with Max-Min system and pheromone updation phase is according to Ant System. This hybrid approach is accompanied by local search technique. In this paper a comparative analysis is done using QAPLIB and it is found that results are improved and are comparable with Ant system and Max-Min ant system algorithm.**

Keywords: Quadratic Assignment Problem; Meta-Heuristic; Ant Colony Optimization; 2-opt iterative local search

I. INTRODUCTION

In last few couple of years the Quadratic Assignment Problem (QAP) which is one of the hardest optimization problems to solve in a reasonable time, has drawn attention of various researchers.

QAP finds a lot of application in real world applications such as Facility layout design (Campus planning) [18], Electric circuit board wiring (locating modules on the board to minimize total wire length) [6], control panel and keyboard design (optimized for a language) [6], machine scheduling (minimizing average job completion time) [17] and many more. Many techniques are proposed to solve QAP one of the most famous and efficient technique is Ant colony optimization. In this paper we have used ACO to solve QAP.

A. Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) was originally introduced in 1957 through Tjalling C. Koopmans and Martin Beckman [21] has been trying to allocate a set of 'n' facilities to a set of 'n' locations [10]. The Quadratic assignment problem (QAP) is associated with optimizing the allocation of a set of facilities to a set of locations with given flows between the facilities and given costs (or distance) between the locations with the objective to minimize the sum of the product between flows and costs.

B. The Problem Statement

Mathematically, the problem consists of 3, n x n matrices as follow:

D= $[d_{kh}]$ = (n x n) Distance matrix (between site 'k' and site 'h')

F= $[f_{ij}]$ = (n x n) Flow matrix (traffic intensity between buildings 'i' and 'j')

C= $[c_{ik}]$ = is the cost of allocate building 'i' at location 'k'

The QAP can be stated as follows:

$$\text{Min}_{\psi \in S(n)} \left[\sum_{i,j=1}^{n} \sum_{k,h=1}^{n} f_{ij}\, d_{\psi(k)\psi(h)} + \sum_{i=1}^{n} c_{ik} \right] \quad 1.1$$

Where S(n) is the set of all possible permutations (corresponding to the allocation) and C_{ik} cost of allocating facility 'i' to location 'k' is generally neglected as it does not make a considerable contribution to the complexity of solving the problem.

The term $f_{ij}\, d_{\psi(k)\psi(h)}$ represents the cost participation of concurrently allocating building 'i' to the site ψ_k and building 'j' to the site ψ_h in the current solution $\psi \in S(n)$. A tangible illustration would be a planning of departments building in University Campus. The problem consists of planning the n buildings to n sites on a campus, where d_{kh} is the distance from site 'k' to site 'h', and f_{ij} is the traffic intensity(flow) between buildings 'i' and 'j'. To obtain the allocation cost of each possible assignment there are n! Ways to allocate them. We multiply prearranged flow between each pair of facilities by the distance between their assigned locations, and sum over all the pairs.

The objective is to minimize the total weekly walking distance between the buildings.

For example Table I and Table II shows a QAP with problem size n=4. The distance matrix d_{kh} (4 x 4) and flow matrix f_{ij} (4 x 4). Generally these matrices are asymmetric but for simplicity we consider symmetric matrices.

TABLE I
Distance Matrix

	Distance			
Distance	0	1	2	3
	1	0	4	5
	2	4	0	6
	3	5	0	0

TABLE II
Flow Matrix

	Flow			
Flow	0	60	50	10
	60	0	30	20
	50	30	0	50
	10	20	50	0

Now we assume the possible best permutation among n! is ψ= {2, 0, 1, 3} means that the facility '2' is allocate at location '0', facility '0' is allocate at location '1', facility '1' is allocate at location '2', and facility '3' is allocate at location '3'. Fig. 1 shown below represents the allocation of locations to facilities.

Facilities →	2	0	1	3
Locations →	0	1	2	3

$$\psi = \text{cost}(\psi) = \sum_{i,j=1}^{n} \sum_{k,h=1}^{n} f_{ij}\, d_{\psi k\, \psi h} \quad 1.2$$

Cost = (0 x 0) + (60 x 2) + (50 x 4)+(10 x 6) + (60 x 2) + (0 x 0) + (30 x 1) + (20 x 3) + (50 x 4) + (30 x 1) + (0 x 0) + (50 x 5) + (10 x 6) + (20 x 3) + (50 x 5)+ (0 x 0) = 1440

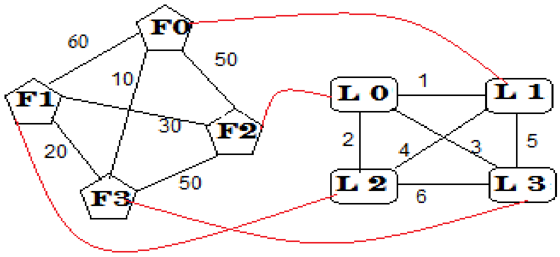

Fig. 1 Representing Allocation of Location to Facilities

The remainder of this paper is organized as follows. Section II gives literature review and brief overview of Ant colony optimization algorithm for QAP. Section III provides a proposed algorithm and local search technique. Section IV provides discussion of the computational results and parameter setting. Section V provides paper conclusion and future scope of this paper.

II. Related Work

QAP belongs to the complex combinatorial discrete optimization problem. Most of the combinatorial optimization problems come under the class of NP-Hard problems. To solve large sized problems exact methods are not suitable for finding the solution in satisfactory time. In order to resolve this problem many approximation algorithms are designed called as heuristic and Meta-heuristic Optimization methods.

A. *Exact Methods:*

The exact methods are used to find the best solution for limited size of problems and their computational cost is very high for high value of n.(i) Brute Force method, (ii) Branch and bound [9], (iii) Cutting planes [8], (iv) Dynamic programming [7]. Nystrom [30] in 1999 proposed "The branch and bound using tree elaborating strategies" that provides the best possible solutions for the QAP problem of size 36.

B. *Approximation Methods:*

Approximation algorithms would be the Method to approaching NP-hard problems. Many optimization problems are NP-hard problems that cannot be solved in polynomial time. To solving optimization problems some heuristic algorithms were proposed the obtained solution by heuristics algorithms need not be necessary same as exact solution but is optimal one which is nearer to best. Heuristic methods are further classified into two types a) Meta-heuristic and b) problem specific heuristic. Meta-heuristic is also of two types: Single solution based and Population based Meta-heuristic. Heuristic or probabilistic methods begin the solution by generating initial random solutions, and apply some heuristic approaches to modifying the result until the solution found is optimal. Several heuristic approaches have been proposed: Greedy Randomized Adaptive search procedure (GRASP)[3], Construction Methods[5], simulated annealing[13], tabu search[14], genetic algorithms[15,16,17], evolution strategies[4,18], scatter search[19] and ant-based algorithms(ant colony optimization)[20], particle swarm optimization[19], artificial bee colony algorithm[2], differential evolution[29], migrating birds optimization[24]. The latest and most efficient approach among these is Ant Colony Optimization algorithms, has proven itself as one of the best performing algorithms for structured, real-life instances.

Burkard and Rendl in 1984 proposed a heuristic Simulated Annealing (SA) approach, the basic idea is comes from physical annealing process in industries, "Simulated Annealing" mean annealing of metal on high temperature, cooling a metal progressively in a specified time schedule. In QAP the initial solution, then generate annealing parameters (temperature, number of iteration, termination condition), update temperature, generate neighbors to select next solution and apply local search to optimize the acceptable results.

Glover & Lagunna in (1989-90) proposed Tabu search to guide the search process which is integrated with several heuristic and metaheuristic algorithm to avoid local optima problem.

Birattari M., Dicaro G. and Dorigo M. (1991) proposed metaheuristic algorithm called "Ant Colony Optimization" is inspired by the real nature of ants and their foraging behavior in finding the shortest path from their nest to food[1]. Several variants of ACO algorithms proposed for solving QAP, such as the Ant System [20] (Dorigo et al., 1996), ACO with elitist ants [20](Dorigo M, 1999), the Max-Min Ant System(MMAS) [21] (Stützle & Hoos, 2000), ANTS[27] (Vittorio Maniezzo , Colorni, 1999), and Hybrid Ant System[25](HAS). The main difference between them is the pheromone updating process during solution construction by individual ants.

1) Ant Colony Optimization Algorithm: ACO algorithms which are based on population based search method are motivated by the behavior of real ants looking for food from the source selects the shortest path. To apply ACO the problem must be represented in a graph as G(S, L) where S is the sets of states and L is the sets of connection to every connected state. To construct solution, ants randomly travel on the graph and they leave some amount of chemical substance called pheromone on the visited edges. This pheromone attracts other ants to follow same path and find the solution after few iterations.

General framework for ACO:
Set parameters & initialize pheromone trails
while (end-condition = false) do
1. Construct_Ant_Solutions
2. Apply LocalSearch(optional)
3. Update Pheromone
end while

In the solution construction, each ant initially construct their solution for the given problem using probabilistic rule which uses pheromone and heuristics information to decide the choosing the next component in the graph.

In the pheromone update, each ant deposits the few amount of pheromone to their constructed solution. The last phase, local search which is optional, which is used to improve the result and it has several types such local search techniques used such as neighborhood iterative exchange, 2-opt local search and tabu search.

C. Ant System for QAP

First initialize the pheromone trails and heuristic information and control parameters.

1) Solution construction: In the main loop, each ant constructs feasible solutions, and then improves the solution using local search and the pheromone trails are updated. In solution construction each ant 'k' randomly selects unassigned facility 'j' to an available location 'i' according to a probabilistic rule specified as follows.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{l \epsilon N_i^k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} \qquad if\ j \epsilon N_i^k$$

1.3

Where $P_{ij}^k(t)$ is the probability of ant 'k' to assigned the facility 'j' to location 'i' at iteration 't' and $\tau_{ij}(t)$ is the chemical pheromone trail of the assignment, indicating how profitable it has been in the past to make that particular assignment at iteration 't', $\eta_{ij}(t)$ is the attractiveness of assigning the facility 'j' to location 'i' at iteration t referred as heuristic information, indicating a priori desirability of that assignment. Thus, more the value of the pheromone content and the heuristic information, the more it will be beneficial to choose that assignment in order to reach feasible solution.

2) Pheromone update: each ant 'k' evaporate the pheromone for all assignment of location and facility (τ_{ij}) as (1- ρ) where ρ is (0 < ρ < 1), and using the equation below pheromone updation is done by every ant.

$$\tau_{ij}(t+1) = \rho\,.\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}{}_{ij}^{k} \qquad 1.4$$

Where ρ is the pheromone evaporation rate which is used to reduce the effect of previously done bad assignment by any ant, m is the number ants and $\Delta\tau_{ij}^{k}$ is the amount of pheromone added by the ant 'k' on the coupling of (i, j) as given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{f(\psi)^{k}} & \text{if facility j is assigned on location i} \\ 0, & otherwise \end{cases}$$

Where, ψ^k represent k^{th} ant permutation and $f(\psi)^k$ is the cost of the permutation and Q is the fixed quantity of pheromone deposited by ant.

D. MAX MIN Ant System

Max-Min ant system (MMAS) is modified from traditional ant system (AS) in various aspects as,

1. In every iteration, only the ant having best solution is allowed to update pheromone, best ant can be iterative best or global best.
2. To avoid search stagnation the pheromone quantity is limited within the specified range [τ_{min}, τ_{max}], that is pheromone should be within interval [$\tau_{min} \leq \tau_{ij} \leq \tau_{max}$].

MMAS does not use any heuristic information because of the construction solution by each ant is improved by local search technique. MMAS algorithm comprises of following phases:

Solution construction: In each iteration of solution construction every ant 'k' randomly chooses an unassigned location 'i' and then place an available facility j based on the following probabilistic rule as:

$$p_{ij}^{k}(t) = \frac{\tau_{ij}(t)}{\sum_{l \epsilon N_i^k}\tau_{il}(t)} \qquad if\ j\ \epsilon\ N_i^k \qquad 1.5$$

1) Pheromone update: After all the ants of single iteration have completed the assignment than pheromone is evaporated with the same concept as the pheromone is evaporated in Ant system.

$$\tau_{ij}(t+1) = \rho.\tau_{ij}(t) + \Delta\tau_{ij}^{best} \qquad 1.6$$

Here, $\Delta\tau_{ij}^{best}$ is defined as

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{Q}{f(\psi)^{k}} & \text{if facility j is assigned on location i in solution } \psi^k \\ 0, & otherwise \end{cases}$$

Where $f(\psi)^k$ is the cost of solution ψ^k of k^{th} ant. In MMAS-QAP the ψ^k is might be either best of the current iterations solution or the global best cost of the solution and Q is the constant amount of pheromone deposited by an ant.

III. HYBRID ANY SYSTEM FOR QAP

In this paper the proposed hybrid ant system algorithm with 2-opt iterative local search. This algorithm has adopted the basic concept of population based ant system algorithms and Max-Min Ant system with some improvement and apply 2-opt iterative local search.

Consider a Graph G(C, L) where C= components are the set of facilities and Locations and L= connection between components are set of flow and distance of facilities and locations respectively. Constraints, each ant build their feasible solution randomly such as every facility is allocated to exactly one location and every location allocate to exactly one facility. Feasible solution ψ consists of n pairs (i, j) of facilities and locations. Assumption: before the start the solution construction the allocation order is fixed in way of allocation facilities to the locations. In this proposed algorithm the allocation order is fixed by each ant by randomly selecting an unassigned location and using probability rule to assigns it to an unassigned available facility.

In the algorithm each artificial "ant" has the following abilities:

1. If a facility 'j' is assigned to location 'i' the ant will lay a pheromone on the pair (i, j).

2. To keep an account of allocated facility to location, each ant has a vector of problem size which stores the assigned facilities. The index of the vector indicates the location number and the element at that index on the vector represent the facility number.
3. For each, new iteration the vector which is associated with every ant must be empty and ant now again constructs the allocation and stores the order in their respective vector.

Algorithm HAS

1. Initialize parameters:
   Rho (ρ), Q, max-Iterations, alpha (α), Pheromone matrix ($\tau_{ij=}1$) and read the flow and distance matrix separately in two distinct n x n arrays.
2. Each ant initially generates their solution using Fisher-Yates-Shuffle and roulette wheel selection algorithm.
3. Find the best ant solution and their cost and store for the further used in iterations.
4. Locate m ants at first location
5. While(iteration<max-Iteration) do
6. Each ant construct solution ψ^k ($1\leq k\leq m$) do
7. For each location ant k randomly select a Unassigned location 'i ($1\leq i\leq n$)
   Allocate a facility j ($1\leq j\leq n$) from *N* (k) on the location i with a probability

$$P_{ij}^{k}=\frac{\tau_{ij}\ (t)}{\sum_{a\varepsilon N(k)}\tau_{ij}\ (t)}$$

   Facilities of ant 'k'.
   End for
8. Improve the constructed solution of each ant k using 2-opt iterative local search
9. Update the global pheromone matrix as
   i. First evaporate the pheromone
   τ_{ij} (t+1)=(1 – ρ) * τ_{ij} (t) for all pair(i, j)
   ii. All ants k deposit the pheromone on those pair (i, j)
   where ant k is assigned the facility j on location i as:

$$\tau_{ij}\ (t+1) = \tau_{ij}\ (t+1) + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

   Where $\Delta\tau_{ij}^{k}$ is the amount of pheromone ant k deposit on the pair (i, j) as:

$$\Delta\tau_{ij}^{k} = \begin{cases}\frac{Q}{f(\psi)^k} & \text{if facility j is assigned on location i}\\ 0, & otherwise\end{cases}$$

   Where, $f(\psi)^k$ is the cost of ant k solution in the current iteration.
10. After update pheromone set the limits of pheromone on each pair of facility and location to avoid search stagnation as below.
    If $\tau_{ij}^{k} < 0.0001$ then $\tau_{ij}^{k} = 0.0001$
    if $\tau_{ij}^{k} > 100000$ then $\tau_{ij}^{k} = 100000$
11. Find the best permutation and its cost among k ants.
12. If (iteration best cost < best cost so far) than
13. Set: Best cost so far = iteration best cost
14. End while
16. Apply local search on the final best permutation and print the final cost and permutation.
   End algorithm

A. *Local search for the QAP*

In the proposed HAS algorithm we have use a 2-opt iterative Local search procedure is applied on the permutation to improve the obtained permutation. It explores all the neighborhood n (n-1)/2 possible swaps using move cost [22] where n is the problem size. The move cost for neighborhood n (n-1)/2 pairs of facility 'i' and 'j' in the permutation are computed as Delta (ψ, i , j)

Algorithm for local search

```
moveCost(permutation, flow matrix, distance matrix)
{
  For i = 0 to n do {
    For j = i+1 to n   do {
        Delta[i][j] = (a[i][i] - a[j][j]) * (b[p[j]][p[j]] -
                       b[p[i]][p[i]]) + (a[i][j] - a[j][i]) *
                       (b[p[j]][p[i]] - b[p[i]][p[j]]);
  For (k = 0; k < n; k = k + 1)
   {  If (k! = i && k! = j)
         Delta[i][j] += (a[k][i] - a[k][j]) * (b[p[k]][p[j]] –
                        b [p[k]][p[i]]) + (a[i][k] - a[j][k]) *
                        (b [p[j]][p[k]] - b[p[i]][p[k]]);
   }
  If (Delta[i][j] < 0)
      Swap (ψ, i , j);
 }  }
```

Where p= permutation, a=flow matrix and b= distance matrix.

If (Delta[i][j] < 0) than swap (ψ, i, j) otherwise no changes. Means if the move cost is negative than exchanging the facility at index 'i' with index 'j' has the cost is less than before computed cost at location 'i' and 'j' in ψ. Local search compute all neighborhood move cost is of complexity $O(k*n^2)$.

IV. Experimental Results

According to the instances of QAPLIB problem can be classified in four categories of instances:

1. Unstructured, randomly generated instances: In this category of instances are generated randomly according to consistent allocation (instances Taixxa) for distance and flow matrix.
2. Unstructured instances with grid-distance: In this category of instances with the distance matrix defined as the Manhattan distance between grid points on n_1 x n_2 grid and with random flow.
3. Real-life instances: In this category of instances from practical applications of the QAP.
4. Real-life like instance: In a QAPLIB real life instances mainly have a small size, so for big size of real life application of QAP the instances in this category are randomly generated in such a way that the matrix entries resemble the distribution found in real life problems (instances Taixxb).

In this section the results obtained from HAS (Hybrid Ant System) has been compared with Improved MMAS with 2-opt

local search (called New MMAS) [26] and Improved MMAS with Tabu search (called New MMAST) [26].
In Table1 proposed HAS algorithm has performed best results; the best results are shown in bold font. Table2 shows Comparative analysis of Ant System [27] and population based Hybrid Ant System (PHAS) [28] with proposed hybrid Ant system algorithm (HAS).

TABLE III
Experimental Results for Improved MMAS with 2-Opt Local Search (Called New MMAS) [26] and Improved MMAS with Tabu Search (Called New MMAST) [26] Compare with Proposed Hybrid Ant System Algorithm (HAS).

Problem Name	Size N	Best Known cost	New-MMAS	New MMAST	Number of Ants	Cost of HAS	RPD %
Unstructured Random generated instances							
Tai25a	25	1167256	1.381	0.665	35	1167256	**0**
Tai30a	30	1818146	1.228	0.657	50	1825384	**0.398**
Tai35a	35	2422002	1.633	**1.029**	50	2456836	1.438
Tai40a	40	3139370	1.733	**1.288**	40	3191706	1.66
Tai50a	50	4938796	2.027	**1.629**	50	5022456	1.693
Tai80a	80	13499184	NA	NA	80	13848642	2.588
Grid based distance matrix							
Nug25	25	3744	0.060	0	25	3744	**0**
Nug30	30	6124	0.276	0.133	30	6124	**0**
Sko42	42	15812	1.135	0.405	52	15842	**0.189**
Sko49	49	23386	1.111	0.501	49	23436	**0.213**
Sko56	56	34458	1.238	0.558	56	34624	**0.481**
Real life instances							
Chr25a	25	3796	3.934	4.284	25	3866	**1.844**
Kra30a	30	88900	NA	NA	50	88900	**0.0**
Kra30b	30	91420	0.125	**0.056**	30	91580	0.175
Ste36b	36	15852	0.137	1.019	36	15892	**0.252**
Bur26a	26	5426670	0.027	0.049	26	5426670	**0.0**
Real life like instances							
Tai25b	25	344355646	0.009	0.018	25	344355646	**0.0**
Tai30b	30	637117113	0.003	0.392	40	637117113	**0.0**
Tai35b	35	283315445	0.085	0.208	35	283611793	**0.1046**
Tai40b	40	637250948	0.035	0.388	50	637307091	**0.0088**
Tai150b	150	498896643	NA	NA	150	505087945	**1.24**

TABLE IV
Experimental Results Comparison by Ant System [27] with Proposed HAS.

Problem Name	Size N	Best known cost	ANT cost % RPD	PHAS % RPD	Number of Ant in HAS	Cost of HAS	RPD %
Chr15c	15	9504	NA	6.355	15	9504	**0.0**
Chr20b	20	2298	2.79	4.96	20	2298	**0.0**
Lipa30a	30	13178	0.0	NA	30	13178	**0.0**
Lipa40a	40	31538	1.02	NA	40	31704	**0.526**
Ste36a	36	9526	0.76	0.251	36	9550	**0.251**
Bur26b	26	3817852	0.0	0.0	26	3817852	**0.0**
Els19	19	17212548	NA	NA	19	17212548	**0.0**

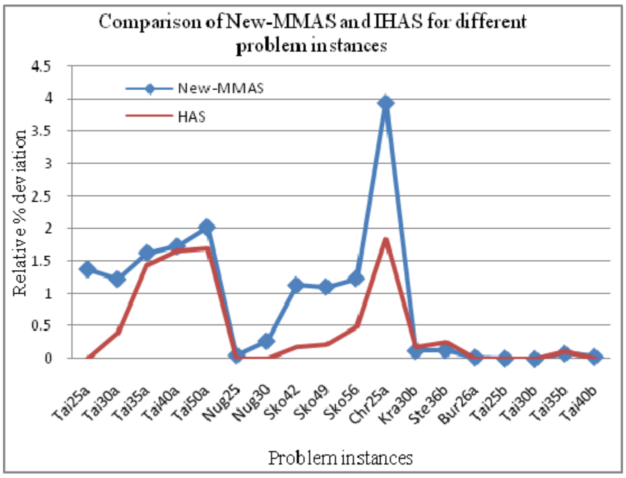

Fig. 2 Graph representing comparison of New-MMAS and HAS in terms of RPD for different Problem instances.

Table1 shows that the proposed HAS algorithm perform better performance for all four type categories problem except some problem from unstructured randomly generated instances category. Table2 shows that the proposed algorithm obtained better results than compared algorithms. We obtained best result for problem chr15c, chr20b and kra25a.

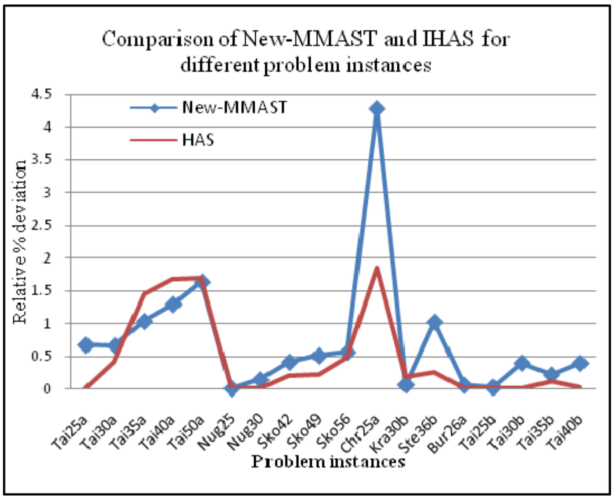

Fig. 3 Graph representing comparison of New-MMAST and HAS in terms of RPD for different Problem instances.

A. Parameter setting

Suitable parameter setting for HAS was determined by preliminary experiments on different type of problems. The parameters are test over 10 times run for getting best result for indivisible problem. The number of ants used is equal to problem size (m = n) and pheromone evaporation rate ρ is problem dependent (0.05, 0.07, 0.08) and Q=1. The number of iteration for local search is 3. Table1 represent the Relative Percentage Deviation (RPD) from the best known cost, the calculation of RPD = $\frac{Z-Z(best)}{Z(best)} * 100$, where Z is the cost of the proposed algorithm and Z(best) is the best known cost of the problem.

The proposed algorithm is implemented in Microsoft visual C# and all the experiments were tested on Intel(R) Pentium(R) CPU G620 @ 2.60 GHz with 2 GB RAM and Windows7 (32 bit) Operating System.

V. CONCLUSION AND FUTURE WORK

In this paper hybrid ant system is proposed which is accompanied by 2-opt local search technique.

It is then compared with improved Max-Min ant system (MMAS) with 2-opt local search (new MMAS) and improved MMAS with tabu search (new MMAST) and results are found improved over these two algorithms for all the specified categories of test cases except some problem instances of large size which comes under Unstructured Random generated instances for new MMAST. In future a hybrid parallel implementation for the proposed algorithm will be proposed and some other variant of ACO combined with different local search techniques will be compared and proposed for solving QAP.

REFERENCES

[1] M. Dorigo, V. Manifesto, and A. Colorni," *The ant system: Optimization by a colony of cooperating Agents"*. IEEE Transactions on Systems, Mans, and Cybernetics, 1(26):29-41, 1996.

[2] D. karaboga, B. Basturk, "*A Powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm*", journal of global optimizationvol. 39 (3), pp. 459-471, 2007.

[3] Li, Y., P.M. Pardalos, and M.G.C. Resende, "*A Greedy Randomized adaptive search procedure for the quadratic assignment problem*", In: Pardalos, PM., Wolkowicz, H. eds., Quadratic Assignment and Related Problems. Vol. 16 of DIMACS series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1994, 237-261.

[4] Taillard E D,"*Robust tabu search for the quadratic assignment problem"*, Parallel Computing 1991; 17:443– 55.

[5] Burkard, R.E. and F.A.Rendl(1983) "*Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems. European Journal of Operational Research* 17, 169-174.

[6] L. Steinberg, "The backboard wiring problem: A placement algorithm", SIAM Review 3, 1961, 37–50.

[7] Christofides, N., and E. Benavent, "An Exact Algorithm for the Quadratic Assignment Problem", Operational Research, 1989, vol. 37, pp. 760-768.

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

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

[10] Cela E. (1998). "*The Quadratic Assignment Problem: Theory and Algorithms"*, Kluwer.

[11] S. sahni and T. Gonzalez, "*P-complete approximation problem*", Journal of the ACM, vol. 23, 555-565, and 1976

[12] A Survey of the Quadratic Assignment Problem, with Applications Clayton W. Commander, the University of Florida April 4, 2003

[13] Wilhelm, M.R. and T.L. Ward, "*Solving Quadratic Assignment Problems by Simulated Annealing*", IIE Transactions, 1987, vol. 19, pp. 107-119

[14] J. skorin-kapov, "*tabu search applied to the quadratic assignment problem*", ORSA Journal on Computing, vol. 2, 1990

[15] E. M. Tate, A.E. smith, "*A genetic approaches to Quadratic assignment problem*", Computer and Operation Research, Vol. 22, 1995

[16] Z. Drezner, "*A new genetic algorithm to the quadratic assignment problem*", INFORMS journal on Computing, Vol. 15, 2003

[17] Carlson, R.C. and G.L. Nemhauser (1966), "*Scheduling to minimize cost*", Operations Research 14, 52-58.

[18] J.W. Dickey and J.W. Hopkins, "*Campus building Arrangement using TOPAZ*", Transportation Research 6, 1972, 59–68.

[19] R.C. Eberhart, J. Kennedy, "*A new optimizer using particle swarm theory*", in: Proceedings of the Sixth International Symposium on Micro machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[20] T. Stutzle and Marco Dorigo. "*ACO Algorithms for the Quadratic Assignment Problem"*. In D.Corne, M.Dorigo and F.Glover, editors, New Ideas in Optimization, McGraw-Hill, 1999.

[21] Koopmans, T.C., and M. Beckman (1957). "*Assignment Problems and the Location of Economic Activities*." Econometric 25, 53-76.

[22] E.D. Taillard. Comparison of Iterative Searches for the Quadratic Assignment Problem. Location Science, 3:87{105, 1995.

[23] QAPLIB benchmarks problem instance: www:opt.math.tu-graz.ac.at/qaplib.html

[24] Ekrem Duman, Mitat Uysal, and Ali Fuat Alkaya, "*Migrating Birds Optimization: A New Meta-heuristic Approach and Its Application to the Quadratic Assignment Problem*", C. Di Chio et al. (Eds.): EvoApplications 2011, Part I, LNCS 6624, pp. 254–263, Springer 2011.

[25] L. M. Gambardella; E. D. Taillard; M. Dorigo "*Ant Colonies for the Quadratic Assignment Problem*", The Journal of the Operational Research Society, Vol. 50, No. 2. (Feb.,1999), pp. 167-176.

[26] Malek Mouhoub and Zhijie Wang, "*Improving the Ant Colony Optimization Algorithm for the Quadratic Assignment Problem*", Conference on Tools with Artificial Intelligence(ICTAI08) IEEE 2008

[27] Vittorio Maniezzo and Alberto Colorni, "*The Ant System Applied to the Quadratic Assignment Problem*", IEEE transaction on knowledge and data engineering, vol. 11, NO. 5, 1999

[28] A.S. Ramkumar and S.G. ponnambalam, "*Hybrid Ant Colony System for solving Quadratic Assignment Formulation of Machine Layout Problems*", IEEE, 2006

[29] R. Storn, K. Price, "*Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces"*, Journal of Global Optimization
-11 (1997) 341–359.

[30] M. Nystrom, "Solving certain large instances of the quadratic assignment problem: Steinberg's examples," California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech CSTR: 2001.010, 1999.