8-2014

Hybrid Metaheuristics for Solving the Quadratic Assignment Problem and the Generalized Quadratic Assignment Problem

Aldy GUNAWAN
Singapore Management University, aldygunawan@smu.edu.sg

Kien Ming Ng

Kim Leng Poh

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Hybrid Metaheuristics for Solving the Quadratic Assignment Problem and the Generalized Quadratic Assignment Problem

Aldy Gunawan, Kien Ming Ng, Kim Leng Poh and Hoong Chuin Lau

Abstract—**This paper presents a hybrid metaheuristic for solving the Quadratic Assignment Problem (QAP). The proposed algorithm involves using the Greedy Randomized Adaptive Search Procedure (GRASP) to construct an initial solution, and then using a hybrid Simulated Annealing (SA) and Tabu Search (TS) algorithm to further improve the solution. Experimental results show that the hybrid metaheuristic is able to obtain good quality solutions for QAPLIB test problems within reasonable computation time. The proposed algorithm is extended to solve the Generalized Quadratic Assignment Problem (GQAP), with an emphasis on modelling and solving a practical problem, namely an examination timetabling problem. We found that the proposed algorithm is able to perform better than the standard SA algorithm does.**

I. Introduction

The Quadratic Assignment Problem (QAP) is identified as the problem of finding a minimum cost for allocating facilities into locations, with the costs being the sum of all possible distance-flow products (Loiola et al., 2007). This problem belongs to the class of *NP*-hard problems.

Some of the surveys of the QAP in the literature were presented by Drezner et al. (2005) and Loiola et al. (2007). There are many practical problems that can be presented as a QAP, such as problems dealing with the facility layout design problem (Benjaafar, 2002) and the placement of electronic components (Duman and Ilhan, 2007). The QAP can be formulated in different ways, such as pure integer programming formulations (Fedjki and Duffuaa, 2004), mixed integer linear programming formulations (Frieze and Yadegar, 1983), graph formulations (White, 1995) and permutation problems (Lim et al., 2000).

Both exact and heuristic methods have been used to solve the QAP. Exact algorithms, which include the branch-and-bound, dynamic programming and cutting plane techniques, can only be used to solve small-size instances of the problem. Thus, many heuristics have been proposed by researchers to find optimal or near optimal solutions for the QAP. These heuristics range from simple iterative improvement procedures to metaheuristic implementations, such as Ant Colony Optimization (Puris et al., 2010), Genetic Algorithm (Lim et al., 2002), Tabu Search (Drezner, 2005) and Simulated Annealing (Tian et al., 1996). Loiola et al. (2007) highlighted the development of hybrid algorithms for solving the QAP. These hybrid algorithms for the QAP include a combination of Tabu Search with Simulated Annealing as presented by Misevicius (2004).

This paper presents a new hybrid metaheuristic for the QAP. It involves three different algorithms: GRASP (Greedy Randomized Adaptive Search Procedure), Simulated Annealing (SA) and Tabu Search (TS). An extensive computational testing of this hybrid metaheuristic has been carried out with the benchmark instances in the QAPLIB, a well-known library of QAP instance (Burkard et al., 1997).

We also consider a modification of the proposed hybrid metaheuristic to solve the Generalized Quadratic Assignment Problem (GQAP), which arises in many applications, such as the service allocation problem (Cordeau et al., 2007) and the examination timetabling problem (Bullnheimer, 1998). The GQAP focuses on assigning all objects to locations so as to minimize the overall distance covered by the flow of materials moving between different objects subject to the resource limitation at each location. Pessoa et al. (2010) proposed exact algorithms that combine Lagrangean decomposition and the Reformulation-Linearization Technique. The performance of the algorithms heavily depends on a good initial upper bound for the heuristic. GRASP with path-relinking heuristics have been proposed by Mateus et al. (2011) to solve some benchmark instances. Enhancing the performance by using randomization was also implemented.

Finally, parameter sensitivity analysis for QAP utilizing one-at-a-time sensitivity measures and the linear regression analysis are conducted to further assess the influences of parameters to the quality of the solutions.

II. Problem Description

A. Quadratic Assignment Problem (QAP)

The QAP is the problem of assigning n facilities to n different locations. Given two $n \times n$ matrices, $F = [f_{ij}]$ and $D = [d_{kl}]$, where f_{ij} is the flow between facilities i and j and d_{kl} is the distance between locations k and l, the problem can be formulated as follows (Loiola et al., 2007):

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl} \quad (1)$$

subject to :

$$\sum_{i=1}^{n} x_{ik} = 1 \qquad 1 \le k \le n \quad (2)$$

A. Gunawan* and H.C. Lau are with the Living Analytics Research Centre, School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 78902 (*corresponding author, phone: 65 6808 5227; fax: 65 6828 0901; e-mail: aldygunawan@smu.edu.sg, hclau@smu.edu.sg).

K.M. Ng and K.L. Poh are with Industrial and Engineering Department, National University of Singapore (e-mail: isenkm@nus.edu.sg, isepohkl@nus.edu.sg).

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad 1 \le i \le n \tag{3}$$

$$x_{ik} \in \{0,1\} \qquad 1 \le i,k \le n \tag{4}$$

(1) represents the total cost of assignment of all facilities to all locations, which is the product of the flow between facilities i and j and the distance between locations k and l. The constraints (2) and (3) ensure that exactly n facilities are to be assigned to exactly n locations.

The QAP can also be represented as a permutation problem. Let $d_{\pi(i)\pi(j)}$ be the distance between locations $\pi(i)$ and $\pi(j)$. The QAP problem then becomes:

$$\min_{\pi \in \Pi(n)} Z(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)} \tag{5}$$

where $\Pi(n)$ is the set of all permutations of {1, 2, …, n}.

In this paper, a solution to the QAP is represented by the vector: $\pi = [\pi(1), \pi(2), \pi(3), \ldots, \pi(n)]$, where the element $\pi(i) = k$ denotes that facility i is assigned to location k.

B. *Generalized Quadratic Assignment Problem (GQAP)*

The GQAP is a generalization of the QAP that allows multiple facilities to be assigned to a single location subject to the resource limitation at each location. In this paper, the examination timetabling problem is selected and explained in greater detail to illustrate the GQAP (Leong and Yeong, 1987). The examination timetabling problem is defined as the problem of allocating a number of examinations to a certain number of time periods in such a way that there would be no conflict or clash, i.e., no student are required to attend more than one examination at the same time. This conflict is being categorized as a hard constraint and it is defined as the first order conflict in Bullnheimer (1998). Besides the first-order conflict, other order conflicts can also be taken into consideration, for instance, the second-order conflict which refers to the situation where two consecutive exams have to be taken by a student, room capacity constraint and so forth.

In many real applications, more than one examinations may be assigned to a given time period when there are ample resources available (rooms and personnel), especially when the number of examinations is greater than the number of time periods available. Thus, constraint (2) is modified and the examination timetabling problem can then be formulated as a GQAP.

Let e and n be the number of examinations and time periods respectively and $e \ge n$. Given ($e \times e$) and ($n \times n$) matrices, $F' = [\ f'_{ij}\ ]$ and $C' = [\ c'_{kl}\ ]$, where f'_{ij} is the number of students taking examinations i and j and c'_{kl} is the cost between time periods k and l, the problem can be formulated as follows:

[*EP Model*]

$$\text{Minimize} \quad Z_{EP} = \sum_{i=1}^{e}\sum_{j=1}^{e}\sum_{k=1}^{n}\sum_{l=1}^{n} f'_{ij} c'_{kl} x_{ik} x_{jl} \tag{6}$$

subject to:

$$\sum_{i=1}^{e} x_{ik} = Cap_k \qquad 1 \le k \le n \tag{7}$$

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad 1 \le i \le e \tag{8}$$

$$x_{ik} \in \{0,1\} \qquad 1 \le i \le e, 1 \le k \le n \tag{9}$$

Let Cap_k be the number of classrooms available in time period k $(1 \le k \le n)$. (7) ensures that the maximum number of examinations scheduled at any time period is Cap_k. In this paper, it is assumed that Cap_k = Cap for $1 \le k \le n$. (8) ensures that each examination is assigned to a particular time period.

Since the *EP Model* assumes that each time period can accommodate more than one examination, there is a possibility that both the first-order and second-order conflicts may occur. In order to minimize both conflicts, different values of c'_{kl} are introduced:

$$c'_{kl} = \begin{cases} 1/(d_{kl})^{\eta} & |k-l| > 1 \\ \mu & |k-l| = 1 \\ M & k = l \end{cases} \tag{10}$$

where

M = a very large positive number

η = a marginal product of time period

μ = a number between 1 and M, and is typically set to 10

d_{kl} = the distance between time periods k and l

d_{kl} is calculated by the actual time differences between two periods, for example, the difference between time periods 2 and 4 is 2 periods. The parameter η emphasizes the importance of the conflict. Setting η = 0 means that we only concern about the first-order and second-order conflicts and we treat other distances d_{kl} as equally important. On the other hand, the cost between two time periods is decreasing for η = 1. A higher value of d_{kl} will result in a lower value of c'_{kl}.

III. HYBRID METAHEURISTICS

The hybrid algorithms for solving QAP and GQAP are presented and described in detail in the following sub-sections.

A. *Hybrid Metaheuristic for QAP*

The hybrid metaheuristic consists of two phases: construction and improvement phases. In the construction phase, an initial solution is built by implementing the Greedy Randomized Adaptive Search Procedure (GRASP) (Yong et al., 1994). We construct an initial solution by adding one new element at a time (e.g. allocation one facility at a time). This process is started by selecting the first 2 assignments based on the minimum cost of interaction $f_{ij}d_{kl}$, followed by assigning the remaining (n-2) facilities based on the cost of assigning a particular facility with respect to the already-made assignments. This process is performed until all the remaining (n–2) facilities have been assigned. The details of the construction process of GRASP can be referred to Yong et al. (1994).

The initial solution generated by GRASP, *initial_sol*, is then improved in the improvement phase using a hybridization of SA and TS algorithm (Algorithm SA-TS) (Figure 1). While it is mainly based on Simulated Annealing (Kirkpatrick et al., 1983), the main difference of the standard SA and the proposed SA lies in the additional elements or strategies added. Several features from TS, such as the tabu length, tabu list and the intensification strategy are incorporated in the algorithm for further improvement (Glover, 1989).

In order to improve the solution, a local search algorithm involving a partial sequential neighborhood search is also augmented. The basic idea of the search is to swap or exchange the locations of two facilities such that a better solution is derived. Assuming that $f_{ii} = f_{jj} = 0$, the objective function difference $\Delta(\pi,i,j)$ obtained by exchanging facilities $\pi(i)$ and $\pi(j)$ can be computed in $O(n)$ operations, using the following equation (Taillard and Gambardella, 1997):

$$\Delta(\pi,i,j) = f_{ij}\left(d_{\pi(j)\pi(i)} - d_{\pi(i)\pi(j)}\right) + f_{ji}\left(d_{\pi(i)\pi(j)} - d_{\pi(j)\pi(i)}\right) + \sum_{a=1,a\neq i,j}^{n}\left\{f_{ai}\left(d_{\pi(a)\pi(j)} - d_{\pi(a)\pi(i)}\right) + f_{aj}\left(d_{\pi(a)\pi(i)} - d_{\pi(a)\pi(j)}\right) + f_{ia}\left(d_{\pi(j)\pi(a)} - d_{\pi(i)\pi(a)}\right) + f_{ja}\left(d_{\pi(i)\pi(a)} - d_{\pi(j)\pi(a)}\right)\right\} \quad (11)$$

Algorithm SA-TS ( )
Initialize the parameters
Set the best solution, *best_sol* = *initial_sol*
Set the current solution, *current_sol* = *initial_sol*
Set the total number of iterations, *num_iter* = 0
Set the total number of iterations without improvement, *no_improv* = 0
While the total number of iterations, *num_iter* is less than the preset maximum number of iterations, *outer_loop* do:
  Repeat *inner_loop* times:
    Select a facility *i* randomly
    Apply a partial sequential neighborhood search
    Find the best permutation π' with the smallest value of $\Delta(\pi',i,j)$
    Check whether the best permutation is tabu or not
    If $\Delta(\pi',i,j) < 0$
      Update the current solution, *current_sol*
        If *current_sol* is better than *best_sol*
          Update the best solution, *best _sol* ← *current_sol*
      Update tabu list
    Else
      Choose a random number *r* uniformly from [0,1]
      no_improv ← *no_improv* + 1
      If $r < exp^{-\Delta(\pi',i,j)/T}$ and the new solution is not tabu
        Accept the new solution, *new_sol*
        Update the current solution, *current_sol*
        Update tabu list
      Else
        Return to the current solution, *current_sol*
        Update tabu list
  Update temperature $T \leftarrow \alpha \times T$
  If (*no_improv* > *limit*)
    Apply the intensification strategy
    Set *no_improv* ← 0
  num_iter ← *num_iter* +1
End while
Report the best solution, *best_sol*

Figure 1. SA-TS for QAP

Instead of selecting two facilities randomly as was commonly done in SA, we start by selecting one facility *i* randomly followed by examining all other potential pair-swaps sequentially in the order $\{(i,j): j \neq i\}$. The selected move is the one with the best $\Delta(\pi,i,j)$ value. The new permutation is then evaluated by the acceptance-rejection procedure in SA. The acceptance of the selected move depends on its $\Delta(\pi,i,j)$ value. For minimization problem, if $\Delta(\pi,i,j) < 0$ or $\Delta(\pi,i,j) \geq 0$, with a probability of $exp^{-\Delta(\pi,i,j)/T}$, the selected move is accepted.

The tabu list contains pairs (*i*, *j*) that have been visited in the last *length* iterations. For a given iteration, if a pair (*i*, *j*) belongs to the tabu list, it is not allowed to accept the exchange of facilities *i* and *j*, unless this exchange gives an objective function value strictly better than the previous one (*aspiration level criteria*). At any temperature *T*, the neighborhood search is repeated until a certain number of iterations, *inner_loop*, has been performed.

If there is no improvement of the solution obtained within a certain number of iterations (*limit*), we apply an intensification strategy of Tabu Search. This strategy focuses the search once again starting from the best permutation obtained. Finally, the entire algorithm will be terminated if the total number of iterations of the outer loop reaches the preset maximum number of iterations, *outer_loop*.

B. *Hybrid Metaheuristic for GQAP*

In the previous section, GRASP is implemented to construct an initial solution for the QAP. Due to some of the general requirements of the examination timetabling problem differing from that of QAP, such as more than one examination to a time period can be assigned, the GRASP is then modified.

The construction phase is started by selecting the first 2 assignments based on the minimum cost of interaction $f'_{ij}c'_{kl}$, followed by assigning the remaining (*e*-2) examinations based on the cost of assigning a particular examination with respect to the already-made assignments, i.e. we select the one that has the minimum cost. This process is made until all the remaining (*e*-2) examinations are assigned. The algorithm applied in the improvement phase is adapted from the Algorithm SA-TS described in Figure 1.

The neighborhood is defined by reallocating an examination of the current solution π to another different time period (single move) such that a better solution π' is derived. It is also necessary to ensure that the maximum number of examinations scheduled at any time period, *Cap*, is not being exceeded. Instead of a random neighborhood search, a partial sequential neighborhood search is used, which involves examining all other potential moves sequentially with respect to time periods for an examination of the current solution π.

The objective function difference $\Delta(\pi,\pi',i)$ obtained by exchanging the time periods of examination *i*, $\pi(i)$ and $\pi'(i)$, is shown in (12):

$$\Delta(\pi,\pi',i) = 2\sum_{a=1,a\neq i}^{e} f_{ai}\left(c'_{\pi(a)\pi'(i)} - c'_{\pi(a)\pi(i)}\right) \quad (12)$$

The selected move is the one with the best $\Delta(\pi,\pi',i)$ value. The new permutation is then evaluated by the acceptance-rejection procedure in SA. We also incorporate features from Tabu Search, such as tabu length, tabu list and intensification strategy in the algorithm.

The tabu list contains examination-time period pairs that have been visited in the last *length* iterations. For a given iteration, if a pair $(i,\pi'(i))$ belongs to the tabu list, it is not allowed to accept the exchange of the time periods $\pi(i)$ and $\pi'(i)$, unless this exchange gives a strictly better objective function value (*aspiration level criteria*). At any temperature *T*, the neighborhood search is repeated until a certain number of iterations, *inner_loop*, has been performed.

IV. COMPUTATIONAL RESULTS

The computational results and comparisons for the proposed hybrid metaheuristics are provided below. The values of the parameters used in the computational experiments are determined experimentally to ensure a compromise between the computation time and the solution quality which are summarized in Table I. The algorithms were implemented using C++ and executed on a 2.67 GHz Intel Core 2 Duo CPU with 3 GB of RAM under the Microsoft Windows Vista Operating System.

TABLE I. PARAMETER SETTINGS FOR QAP, GQAP

Parameter	Value (QAP)	Value (GQAP)
Maximum number of iterations, *outer_loop*	300n	50e
Initial temperature, *T*	5,000	1000
Number of neighborhood moves at each temperature *T*, *inner_loop*	100n	100n
Cooling factor, α	0.9	0.9
Number of non-improvement iterations prior to intensification, *Limit*	0.02*outer_loop*	0.01*outer_loop*
Length of tabu list, *length*	$n/2$	$e/2$

A. QAP Results

In order to evaluate the performance of our proposed approach, we have solved some benchmark problems from the QAPLIB. For each benchmark problem, the proposed algorithm was executed 20 times with different random seeds. All are solved within reasonable CPU time. Due to the space limitation, we did not report the computation time.

For all problem instances, the best known/optimal solutions are also obtained within reasonable computation time. The objective function values of the optimal/best known solutions given in Burkard et al. (1997) are also presented for comparison purposes. The heading Φ_1 refers to % deviation between the average objective function value of the solutions obtained and the best known/optimal solution, while the heading Φ_2 refers to % deviation between the best objective function value of the solutions obtained and the best known/optimal solution.

Table II summarizes the computational results for the *chr* problem instances. The difficulty level in solving the *chr* problem instances is considered significant (Lim et al., 2002). On the whole, the proposed hybrid algorithm is able to find solutions with values of Φ_1 not exceeding 1.50% from the known optimum.

TABLE II. SA-TS RESULTS FOR *chr* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*chr*12a	9552	9552	9552	0.00	0.00
*chr*12b	9742	9742	9742	0.00	0.00
*chr*12c	11156	11156	11156	0.00	0.00
*chr*15a	9896	9896	9896	0.00	0.00
*chr*15b	7990	7990	7990	0.00	0.00
*chr*15c	9504	9504	9504	0.00	0.00
*chr*18a	11098	11098	11098	0.00	0.00
*chr*18b	1534	1534	1534	0.00	0.00
*chr*20a	2192	2224.9	2192	1.50	0.00
*chr*20b	2298	2306.7	2298	0.38	0.00
*chr*20c	14142	14142	14142	0.00	0.00
*chr*22a	6156	6181.3	6156	0.41	0.00
*chr*22b	6194	6265.2	6194	1.15	0.00
*chr*25a	3796	3811	3796	0.40	0.00

Tables III and IV summarize the results of testing on *had* and *kra* problem instances. The average gaps of the solutions are less than 0.75%. The hybrid algorithm is again able to obtain the best known/optimal solutions.

TABLE III. SA-TS RESULTS FOR *had* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*had*12	1652	1652	1652	0.00	0.00
*had*14	2724	2735	2724	0.40	0.00
*had*16	3720	3721	3720	0.03	0.00
*had*18	5358	5358	5358	0.00	0.00
*had*20	6922	6927.2	6922	0.08	0.00

TABLE IV. SA-TS RESULTS FOR *kra* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*kra*30a	88900	89554.5	88900	0.74	0.00
*kra*30b	91420	91420	91420	0.00	0.00
*kra*32	88700	88700	88700	0.00	0.00

Table V is a summary of the results for the *nug* problem instances. The results indicate that these problem instances do not pose much difficulty for the proposed hybrid algorithm to obtain good solutions as the values of Φ_1 are not more than 0.02%.

Tables VI, VII and VIII show the results of testing on *rou*, *scr* and *sko* problem instances. The values of Φ_1 are not more than 0.03% for *rou* and *scr* problem instances, while the maximum value of Φ_1 is only 0.18% for *sko* problem instance. For *sko*49 and *sko*56, the values of Φ_2 are about 0.1% from the optimal/best known solution.

While the proposed hybrid algorithm is unable to obtain the best known/optimal solutions for the *tai* and *wil* problem instances when $n > 20$ as shown in Tables IX and X, the values of Φ_1 and Φ_2 do not exceed 3.72% and 3.58% respectively.

TABLE V. SA-TS RESULTS FOR *nug* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*nug*12	578	578	578	0.00	0.00
*nug*14	1014	1014	1014	0.00	0.00
*nug*15	1150	1150	1150	0.00	0.00
*nug*20	2570	2570	2570	0.00	0.00
*nug*21	2438	2438	2438	0.00	0.00
*nug*22	3596	3596	3596	0.00	0.00
*nug*24	3488	3488	3488	0.00	0.00
*nug*25	3744	3744	3744	0.00	0.00
*nug*27	5234	5234	5234	0.00	0.00
*nug*28	5166	5166.9	5166	0.02	0.00
*nug*30	6124	6124.4	6124	0.01	0.00

TABLE VI. SA-TS RESULTS FOR *rou* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*rou*12	235528	235528	235528	0.00	0.00
*rou*15	354210	354210	354210	0.00	0.00
*rou*20	725522	725742.7	725522	0.03	0.00

TABLE VII. SA-TS RESULTS FOR *scr* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*scr*12	31410	31410	31410	0.00	0.00
*scr*15	51140	51140	51140	0.00	0.00
*scr*20	110030	110030	110030	0.00	0.00

TABLE VIII. SA-TS RESULTS FOR *sko* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*sko*42	15812	15833.8	15812	0.14	0.00
*sko*49	23386	23424.5	23410	0.16	0.10
*sko*56	34458	34520.4	34494	0.18	0.10

TABLE IX. SA-TS RESULTS FOR *tai* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*tai*10a	135028	135028	135028	0.00	0.00
*tai*12a	224416	224416	224416	0.00	0.00
*tai*15a	388214	388214	388214	0.00	0.00
*tai*17a	491812	491812	491812	0.00	0.00
*tai*20a	703482	704610.2	703482	0.16	0.00
*tai*25a	1167256	1182462.3	1175490	1.30	0.71
*tai*30a	1818146	1845611.7	1833020	1.51	0.82
*tai*35a	2422002	2484348.1	2477054	2.57	2.27
*tai*40a	3139370	3228315.1	3207852	2.83	2.18
*tai*50a	4938796	5122386.6	5115612	3.72	3.58
*tai*60a	7205962	7463484.2	7417240	3.57	2.93
*tai*80a	13515450	13997867.4	13938662	3.57	3.13
*tai*100a	21054656	21788679.9	21689698	3.49	3.02

TABLE X. SA-TS RESULTS FOR *wil* PROBLEM INSTANCES

Benchmark problem	Optimal/Best known Sol.	Average Sol.	Best Sol.	Φ_1 (%)	Φ_2 (%)
*wil*50	48816	48867.7	48850	0.11	0.07
*wil*100	273038	273406.3	273240	0.13	0.07

In summary, we observe that the proposed hybrid algorithm is able to obtain very good or optimal solutions to benchmark problem instances drawn from the QAPLIB.

B. GQAP Results

For the GQAP, the computational results are focused on solving the examination timetabling problem. The random data sets have sizes that are comparable to an examination timetabling problem arising in a university of Indonesia (Table XI). The value of the parameter η is set to 1.

The algorithm is also repeated for 20 runs, with the average objective function value, the best objective function value and the average computation time being tabulated. To see if the proposed hybrid algorithm is an improvement over a standard SA algorithm, we also applied the standard SA algorithm to data sets.

TABLE XI. CHARACTERISTICS OF THE EXAMINATION PROBLEM

Data set	Number of examinations *e*	Number of periods *n*	Number of classrooms *Cap*
20×20	20	20	3
40×20	40	20	4
60×20	60	20	5
80×20	80	20	6
100×20	100	20	7
200×20	200	20	12

Table XII reports the results obtained by both algorithms and it indicates that the performance of the hybrid algorithm is better than the standard SA in terms of the average and the best objective function values obtained. The computation times needed by both algorithms are relatively comparable. For example, the CPU times for 100×20 instance are 340.90 seconds (by standard SA) and 350.52 seconds (by SA-TS), respectively.

TABLE XII. COMPUTATIONAL RESULTS FOR EXAMINATION PROBLEM

Data set	Algorithm SA		SA-TS	
	Average obj. value	Best obj. value	Average obj. value	Best obj. value
20×20	123.36	121.68	121.84	121.68
40×20	642.24	628.48	634.82	627.66
60×20	2743.22	2657.32	2699.42	2652.08
80×20	7723.47	7410.62	7620.33	7281.68
100×20	20275.46	18167.34	19456.78	17583.36
200×20	32334.50	30127.32	31297.83	29604.61

C. Sensitivity Analysis

Hutter et al. (2009) presented the importance of finding good parameter settings that affects the performance of an algorithm. We conduct a sensitivity analysis utilizing one-at-a-time sensitivity measures and linear regression analysis. Figure 2 shows an example of varying the value of the initial temperature T for *chr* instances of QAP. We observe that the higher the value of temperature T, the lower the average percentage deviation from the optimal/best known solutions.

A linear regression function is also built in order to provide the comprehensive sensitivity measure of the average percentage deviation from the optimal/best known solutions (Y). Two parameters, T and α, are selected to build the regression, as shown in (14). It shows that the coefficient for T is -0.0005, which indicates that for every additional degree in temperature *T*, the average percentage deviation *Y* will decrease by an average of 0.0005, by keeping α constant.

$$Y = 20.4 - 0.0005T - 22.4\alpha \quad (14)$$

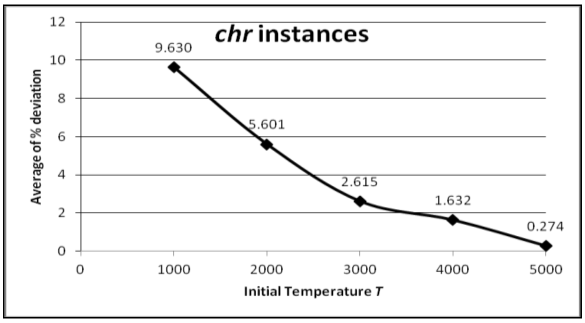

Figure 2. Sensitivity Analysis for *chr* instances

V. Conclusion

In this paper, hybrid metaheuristics combining GRASP, Simulated Annealing and Tabu Search are proposed to solve the QAP and GQAP. The proposed algorithm is able to obtain the optimal or best known solutions for problem instances drawn from the QAPLIB. A modification of the algorithm also performs better than the standard SA algorithm in solving the examination timetabling problem.

The Tabu Search framework has been designed primarily with short term memory. As part of future research work, the possibility of implementing other Tabu Search strategies, such as long term memory and diversification strategy, can be considered. Comparison with other metaheuristics, such as Tabu Search and Genetic Algorithm can also be performed.

One-at-a-time sensitivity measures and linear regression analysis do not examine the possibility of interaction effects between parameters. Mateus et al. (2011) studied the effect of changing single parameter values and fixing the values of all other parameters. Extending their work using the framework proposed by Gunawan et al. (2011) for fine-tuning algorithm parameters considering the interaction effects among parameters is another area of future work.

Acknowledgment

This research is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

References

[1] Benjaafar, S. (2002). Modeling and analysis of congestion in the design of facility layouts. *Management Science*, 48(5), 679-704.

[2] Bullnheimer, B. (1998). An examination scheduling model to maximize students' study time. In E. K. Burke, & M. W. Carter (Eds.). Practice and Theory of Automated Timetabling: Second International Conference. Selected Papers. *Lecture Notes in Computer Science* (Vol. 1408, pp. 78-91). Springer.

[3] Burkard, R. E., Karisch, S. E., & Rendl, F. (1997). QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 10, 391-403.

[4] Cordeau, J.-F., Gaudioso, M., Laporte, G. & Moccia, L. (2007). The service allocation problem at the Gioia Tauro maritime terminal. *European Journal of Operational Research*, 176, 1167-1184.

[5] Drezner, Z. (2005). The extended concentric tabu for the quadratic assignment problem. *European Journal of Operational Research*, 160, 416-422.

[6] Drezner, Z., Hahn, P. M., & Taillard, É. D. (2005). Recent advances for the QAP problem with special emphasis on instances that are difficult for metaheuristic methods. *Annals of Operations Research*, 139, 65-94.

[7] Duman, E. & Ilhan, O. (2007). The quadratic assignment problem in the context of the printed circuit board assembly process. *Computers & Operations Research*, 34, 163-179.

[8] Fedjki, C. A., & Duffuaa, S. O. (2004). An extreme point algorithm for a local minimum solution to the quadratic assignment problem. *European Journal of Operational Research*, 156(3), 566-578.

[9] Glover, F. (1989). Tabu search – part I. *ORSA Journal on Computing*, 1, 190-206.

[10] Gunawan, A., Lau, H. C., & Lindawati (2010). Fine-tuning algorithm parameters using the design of experiment approach. In C. A. Coello Coello, (Eds.). Learning and Intelligence Optimization Conference, LION5. *Lecture Notes in Computer Science* (Vol. 6683, pp. 278-292). Springer.

[11] Hutter , F., Hoos, H. H., Leyton-Brown, K. & Stützle, T. (2009). ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1), 267-306.

[12] Kirkpatrick, S., Gellatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.

[13] Leong, T. Y., & Yeong, W. Y. (1987). Examination scheduling: A quadratic assignment perspective. In Proceedings of International Conference on Optimization: Techniques and Applications, Singapore, pp. 550-558.

[14] Lim, M. H., Yuan, Y., & Omatu, S. (2000). Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem. *Computational Optimization and Applications*, 15, 249-268.

[15] Lim, M. H., Yuan, Y., & Omatu, S. (2002). Extensive testing of a hybrid genetic algorithm for solving quadratic assignment problems. *Computational Optimization and Applications*, 64, 47-64.

[16] Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., & Querido, T. (2007). A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176, 657-690.

[17] Mateus, G. R., Resende, M. G. C., & Silva, R. M. A. (2011). GRASP with path-relinking for the generalized quadratic assignment problem. *Journal of Heuristics*, 17, 527-565.

[18] Misevicius, A. (2004). An improved hybrid optimization algorithm for the quadratic assignment problem. *Mathematical Modelling and Analysis*, 9(2), 149-168.

[19] Pessoa, A. A., Hahn, P. M., Guignard, M., & Zhu, Y-R. (2010). Algorithms for the generalized quadratic assignment problem combining Lagrangean decomposition and the Reformulation0Linearization Technique. *European Journal of Operational Research*, 206, 54-63.

[20] Puris, A. Bello, R., & Herrera, F. (2010). Analysis of the efficacy of a two-stage methodology for ant colony optimization: case of study with TSP and QAP. *Expert Systems with Applications*, 37, 5443–5453.

[21] Sahni, S., & Gonzales, T. (1976). P-Complete approximation problems. *Journal of the Association for Computing Machinery*, 23, 555-565.

[22] Tian, P., Wang, H. C., & Zhang, D. M. (1996). Simulated annealing for the quadratic assignment problem: a further study. *Computers and Industrial Engineering*, 31(3-4), 925-928.

[23] White, D. J. (1995). Some concave-convex representations of the quadratic assignment problem. *European Journal of the Operational Research*, 80(2), 418-424.

[24] Yong, L., Pardalos, P. M., & Resende, M. G. C. (1994). A greedy randomized adaptive search procedure for the quadratic assignment problem. In P. M. Pardalos, & H. Wolkowicz (Eds.). Quadratic assignment and related problems: DIMACS Workshop. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (Vol. 16, pp. 237-261). American Mathematical Society, Providence.