An Elitist GRASP Metaheuristic for the Multi-objective Quadratic Assignment Problem

Hui Li and Dario Landa-Silva

Automated Scheduling Optimisation and Planning Research Group
School of Computer Science, The University of Nottingham, United Kingdom
{hzl,jds}@cs.nott.ac.uk

Abstract. We propose an elitist Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic algorithm, called mGRASP/MH, for approximating the Pareto-optimal front in the multi-objective quadratic assignment problem (mQAP). The proposed algorithm is characterized by three features: elite greedy randomized construction, adaptation of search directions and cooperation between solutions. The approach builds starting solutions in a greedy fashion by using problem-specific information and elite solutions found previously. Also, mGRASP/MH maintains a population of solutions, each associated with a search direction (i.e. weight vector). These search directions are adaptively changed during the search. Moreover, a cooperation mechanism is also implemented between the solutions found by different local search procedures in mGRASP/MH. Our experiments show that mGRASP/MH performs better or similarly to several other state-of-the-art multi-objective metaheuristic algorithms when solving benchmark mQAP instances.

1 Introduction

The quadratic assignment problem (QAP) models many real-world optimization problems in diverse areas such as operations research, economics, etc. One of its major applications is facility location, where a set of facilities should be assigned to different locations. The objective is to find an assignment of all facilities to all locations, such that the total cost is minimized. The QAP is a **NP**-hard combinatorial optimization problem [1]. So, there is no known exact algorithm for solving the QAP in polynomial time. Recently, the multi-objective QAP (mQAP) has been investigated by researchers in the multi-objective optimization community [2,3]. Unlike the single-objective QAP, the mQAP involves multiple types of flows between any two facilities.

Over the last decades, research on multi-objective metaheuristics, such as evolutionary algorithms, simulated annealing, and tabu search, has attracted a lot of attention from the scientific community. A majority of these algorithms use either Pareto dominance or weighting method for fitness assignment. For example, two representative Pareto-based evolutionary multi-objective (EMO) algorithms - NSGA2 [4] and SPEA2 [5] rank the members of the population by comparing them in terms of Pareto domination while MOEA/D [6] defines the

M. Ehrgott et al. (Eds.): EMO 2009, LNCS 5467, pp. 481–494, 2009.

fitness of individuals by using weighted functions. To find a well-distributed set of solutions, some strategies, such as estimating the density of non-dominated solutions and maintaining a set of uniform weights, have been used to maintain the diversity of population in these algorithms.

It is well-known that well-designed genetic operators play an important role in improving the performance of evolutionary algorithms. The proximate optimality principle (POP) [7] assumes that good solutions share some similarities in the decision space. This principle holds for many real-world problems. Based on this principle, Zhang and Sun [8] proposed a genetic operator, called guided mutation, to sample solutions in promising areas of the search space. This is achieved by modifying the elite solutions found previously and then using global information from a probabilistic model. The combination of guided mutation with iterated local search produced competitive results for solving the QAP in [8].

GRASP [9] is one of the most successful metaheuristics for combinatorial optimization. It is a multi-start local search approach. In each iteration of GRASP, two procedures are involved: greedy randomized construction of starting solutions and a local search procedure. A multi-objective version of GRASP was proposed in [10] to handle multi-objective knapsack problem. In that algorithm, each solution is improved along a certain direction by local search. However, the local optima obtained in different iterations do not interact with each other. As shown in [6] and [11], cooperation between solutions with similar search directions and the adaptive change of these search directions is beneficial. In this paper, we propose an elitist multi-objective GRASP metaheuristic called mGRASP/MH. We assess the performance of mGRASP/MH by applying it to a number of benchmark mQAP instances and comparing its performance to that of some existing multi-objective algorithms.

The remainder of this paper is organized as follows. Section 2 formulates the mQAP and discusses fast local search for this problem. Section 3 discusses some important issues of the basic GRASP algorithm for single objective optimization. Section 4 presents the proposed mGRASP/MH for the mQAP. Experimental results are presented and discussed in Section 5 while Section 6 concludes the paper.

2 The Multi-objective Quadratic Assignment Problem

2.1 Mathematical Formulation

Given a location matrix $A = \{a_{ij}\}_{n\times n}$ and flow matrices $B^k = \{b^k_{rs}\}_{n\times n\times m}, k = 1,\ldots,m$, the mQAP is to minimize the following objective functions simultaneously:

$$C(\pi) = \{C^1(\pi),\ldots,C^m(\pi)\}, \pi \in \Omega \tag{1}$$

with

$$C^k(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b^k_{\pi_i\pi_j}, k = 1,\ldots,m \tag{2}$$

where

- n is the number of locations/facilities, m is the number of objectives (i.e. types of flows), $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation of $L = \{1, \ldots, n\}$, Ω is the set of all permutations, $C(\pi)$ is a vector of m objective functions $C^k(\pi), i = 1, \ldots, m$.
- a_{ij} is the distance between locations i and j, and $b^k_{\pi_i \pi_j}$ is the k-th flow between facilities π_i and π_j.

In the case of conflicting objectives, there is no solution π^* which is optimal for all objective functions $C^k(\pi), k = 1, \ldots, m$. Instead, the optimal solution π^* to the mQAP in (1) is often defined as the trade-off solution in terms of Pareto optimality. Assume u and v are objective vectors, u is said to *dominate* v if and only if $u_k \leq v_k$ for all $k = 1, \ldots, m$, and $\exists s \in \{1, \ldots, m\}$, $u_s < v_s$. A solution π^* is said to be Pareto-optimal to (1) if $C(\pi^*)$ is not dominated by $C(\pi)$ for any $\pi \in \Omega$. The *Pareto-optimal front* (POF) is the set of objective vectors of all Pareto-optimal solutions.

In the mathematical programming community, multi-objective optimization problems are often tackled using some form of weighted sum method that combines multiple objective functions into a single scalar function as follows:

$$f(\pi|\lambda) = \sum_{k=1}^{m} \lambda_k \cdot C^k(\pi) \tag{3}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is the weight vector with $\lambda_k \geq 0, k = 1, \ldots, m$ and $\sum_{k=1}^{m} \lambda_k = 1$. Each component of λ can be regarded as the preference w.r.t each objective. The global minima of $f(\pi)$ in (3) is also Pareto-optimal to the mQAP in (1). By minimizing the scalar functions (3) with appropriate weight vectors, a good approximation of the POF is likely to be obtained. However, the weighted sum method cannot solve the multi-objective optimization problems with non-convex POF. Despite this, the weighed sum method has been successfully applied to solve many multi-objective combinatorial optimization problems.

2.2 Fast Local Search

Local search based on 2-opt operator has been widely used to tackle some permutation-based combinatorial optimization problems. In the QAP, the neighborhood of the current solution consists of all solutions obtained by exchanging the positions of two elements in its permutation [12] (i.e., 2-opt swap). Since all elements in the new solution, except the exchanged ones, remain the same, the computation of the objective function value for neighboring solutions can be done quickly by considering only those exchanged elements. In the case of the mQAP, the computation of the function values of neighboring solutions is very similar. Assume that i and j are two positions exchanged in permutation π, the difference $\Delta(\pi, k, i, j)$ of function values regarding the k-th flow before and after exchanging elements i and j can be stated as:

$$\begin{aligned}\Delta(\pi,k,i,j) = &(a_{jj}-a_{ii})(b^k_{\pi_i\pi_i}-b^k_{\pi_j\pi_j})+\\ &(a_{ji}-a_{ij})(b^k_{\pi_i\pi_j}-b^k_{\pi_j\pi_i})+\\ &\textstyle\sum_{s=1,s\neq i,j}^{n}((a_{sj}-a_{si})(b^k_{\pi_s\pi_i}-b^k_{\pi_s\pi_j})+\\ &(a_{js}-a_{is})(b^k_{\pi_i\pi_s}-b^k_{\pi_j\pi_s}))\end{aligned} \tag{4}$$

When A and $B^k, k = 1, \ldots, m$, are symmetric,

$$\Delta(\pi,k,i,j) = 2\sum_{s=1,s\neq i,j}^{n}(a_{sj}-a_{si})(b^k_{\pi_s\pi_i}-b^k_{\pi_s\pi_j}) \tag{5}$$

Then, the function value of the neighboring solution $\bar{\pi}$ after swapping the elements i and j is

$$C^k(\bar{\pi}) = C^k(\pi) + \Delta(\pi,k,i,j), k = 1, \ldots, m. \tag{6}$$

The computational complexity in (6) is only $O(n)$, which is much less than the complexity of evaluating $C(\bar{\pi})$ in (1) (i.e. $O(n^2)$).

3 Greedy Randomized Adaptive Search Procedure

GRASP is a multi-start metaheuristic algorithm, which repeatedly improves starting solutions by local search. At each iteration of GRASP, a greedy randomized constructive procedure and a local search procedure are involved. The best local optimum collected over all local searches is retained and returned as the final solution of GRASP.

3.1 Greedy Randomized Construction

A greedy randomized construction procedure for building starting solutions is shown in Fig. 1. Initially, a partial solution S is set as an empty set. Then, the greedy function values of all unselected components in E are evaluated. To make better contribution to the partial solution S, a restricted candidate list (RCL) is formed by the components with low g values in E. One of the commonly-used strategies to determine RCL is to select the elements with g values between

$$[g^{min}, g^{min} + \alpha \times (g^{max} - g^{min})],$$

where $g^{min} = \min\{g(e)|e \in \mathrm{E}\}$ and $g^{max} = \max\{g(e)|e \in \mathrm{E}\}$. Here, $\alpha \in [0, 1]$ is a parameter to balance the greediness and randomness of the partial solution S. When $\alpha = 0$, only the component with the minimal g value will be selected. This component should make the biggest contribution to the partial solution. On the contrary, when $\alpha = 1$, all candidate components in E have equal chance to be selected. That is, the construction procedure will pick unselected components randomly. In practice, α is set to be either fixed or adaptive.

```
1  begin
2      S := ∅ and E := {all components of solution}.
3      while E is not empty do
4          foreach e in E do compute greedy function value g(e);
5          Define RCL as the set of elements in E with low g values;
6          Select an element ē ∈ RCL randomly;
7          Add ē to partial solution (i.e., S := S ∪ {ē});
8          Remove ē from E (i.e., E := E\{ē}).
9      end
10 end
```

Fig. 1. Greedy Randomized Constructive Procedure of GRASP

3.2 Local Search Procedure

Following the construction step, local search is applied to improve starting solutions. Two basic strategies - first improvement and best improvement, are often considered to accept local search moves. In first improvement, the first neighbor with better objective function value examined is accepted as the new current solution. In contrast, best improvement examines all neighbors and accepts the best one as the new current solution. More sophisticated local search methods with good global search ability, such as simulated annealing and tabu search, have also been suggested to improve the starting solutions in GRASP [13].

4 The Proposed mGRASP/MH Algorithm

4.1 Motivation

In [10], a GRASP algorithm, denoted mGRASP here, was developed to tackle the multi-objective knapsack problem. Like single-objective GRASP algorithms, mGRASP uses a greedy randomized construction step and a local search step. At each iteration, a weighted sum function is defined as the utility function for selecting greedy elements in the construction step and accepting better neighbors in the local search step.

To find a diverse set of Pareto-optimal solutions, mGRASP uses multiple distinct weight vectors evenly spread. According to the experimental setting reported in [10], up to one thousand weight vectors are used in one thousand iterations of mGRASP. Note that each iteration of mGRASP is independent from the other iterations. As shown in [6,11], the adaptation of finite weight vectors and the cooperation between solutions with similar weight vectors could benefit the diversity and convergence in multi-objective search. These strategies can be easily used in mGRASP.

Inspired by the POP principle, the guided mutation operator generates solution in a different way to greedy randomized construction [8]. This operator uses the global information in a probabilistic model to disturb the elite solutions

```
1  Algorithm 1: mGRASP/MH
   input  : N: population size, α: balance factor between greediness and
            randomness, β: proportion of components from elite solution
   output: NDS: the set of all non-dominated solutions
2  Initialize P = {π^(1), ..., π^(N)} and W = {λ^(1), ..., λ^(N)}.
3  begin
4      repeat
5          foreach i ∈ {1, ..., N} do
6              Step 1: Generate greedy solution π based on λ^(i) and π^(i);
7              Step 2: Apply local search on π to produce π' and update NDS;
8              Step 3: Replace the worse members in P with π';
9              Step 4: Modify the search direction λ^i adaptively.
10         end
11     until stopping condition is satisfied ;
12 end
```

Fig. 2. Framework of mGRASP/MH

found during the search. This idea has not yet been used in multi-objective algorithms. Then, we improve the performance of mGRASP by constructing promising starting solutions based on elite solutions.

4.2 mGRASP/MH for the QAP

We propose an elitist multi-objective GRASP metaheuristic in this paper, called mGRASP/MH. At each iteration, a population $P = \{\pi^{(1)}, \ldots, \pi^{(N)}\}$ of solutions and a set of corresponding weight vectors $W = \{\lambda^{(1)}, \ldots, \lambda^{(N)}\}$ are maintained. The framework of mGRASP/MH is shown in Fig. 2. The four main steps in lines 6-9 are involved in the main loop of mGRASP/MH. In the following, each of these steps is detailed.

Step 1: Elitist-based Greedy Construction. Unlike the greedy randomized construction algorithm in Fig. 1, the construction algorithm shown in Fig. 3 uses not only problem-specific greedy information but also the elite solution $\pi^{(i)}$ found in the previous local search. Parameter α is used to balance the greediness and the randomness of the partial solution. The parameter $\beta \in [0, 1]$ is used to control the proportion of components copied from the elite solution $\pi^{(i)}$. n_0 is the number of elements copied from $\pi^{(i)}$. ϕ is a random order of locations. L' denotes the set of locations assigned. In lines 4-6, n_0 components in $\pi^{(i)}$ are directly copied into a new solution π. Line 7 calculates the cost of the partial solution containing the components only from elite solutions. LOC and FAC in line 8 are the set of locations and facilities unassigned.

The ground set E is composed of all unassigned (location, facility) pairs. For each pair, the growth in cost is computed in lines 11-13. The associated g value is obtained in line 14. In line 16, RCL is formed by selecting a set of (location,

```
1  Algorithm 2: ElitistGreedyConstruction(α, β, λ^(i), π^(i))
   input  : λ^(i): current weight vector, π^(i): elite solution
   output: π: greedy randomized elite solution
2  begin
3      Set φ = {φ_1, ..., φ_n} to be a random permutation of L = {1, ..., n},
       n_0 = ⌊β × n⌋, and L' = ∅;
4      for c = 1 to n_0 do
5          π_{φ_c} = π^(i)_{φ_c}; L' = L' ∪ {φ_c};
6      end
7      for k = 1 to m do C^k = Σ_{i∈L'} Σ_{j∈L'} a_ij · b^k_{π_i π_j};
8      Set LOC = {φ_{n_0+1}, ..., φ_n} and FAC = {π^(i)_{φ_{n_0+1}}, ..., π^(i)_{φ_n}};
9      while FAC is not empty do
10         foreach (lc, fc) ∈ M = LOC × FAC do
11             for k = 1 to m do
12                 Δ(lc, fc, k) = Σ_{p∈L'} a_{p,lc} b^k_{π_p,fc} + Σ_{q∈L'} a_{lc,q} b^k_{fc,π_q} + a_{lc,lc} b^k_{fc,fc}
13             end
14             g(lc, fc) = Σ_{k=1}^{m} λ^(i)_k · (C^k + Δ(lc, fc, k));
15         end
16         RCL = {(lc, fc) | g^min ≤ g(lc, fc) ≤ g^min + α(g^max − g^min)};
17         Randomly select a pair (lc', fc') from RCL and set π_{lc'} = fc' and
           L' = L' ∪ {lc'};
18         for k = 1 to m do C^k = C^k + Δ(lc', fc', k);
19         Set LOC = LOC\{lc'} and FAC = FAC\{fc'}.
20     end
21     return π;
22 end
```

Fig. 3. Elitist-based Greedy Construction Procedure for the mQAP

facility) pairs with the g values between $[g^{min}, g^{min} + \alpha(g^{max} - g^{min})]$, where $g^{min} = \min\{g(lc, fc) | (lc, fc) \in M\}$ and $g^{max} = \max\{g(lc, fc) | (lc, fc) \in M\}$. One pair (lc', fc') of (location, facility) is randomly selected from RCL and updates the partial solution in line 17. In line 18, the total cost of partial solution with the pair selected in the previous step is computed. Line 19 removes lc' and fc' from the sets of unassigned locations and facilities respectively. This procedure is repeated until the set FAC is empty. Finally, a complete solution is returned.

Step 2: Local Search. After constructing an elite greedy solution, a local search procedure is triggered and guided by the weighted sum function with $\lambda^{(i)}$ in (3). In mGRASP/MH, 2-opt local search with first improvement is used for the mQAP. Each local search procedure is terminated if there is no solution in its neighborhood with better fitness. Since all members of the population have different weight vectors (i.e. search directions), the set of all local optima found for all search directions is likely to cover the POF reasonably well. The set *NDS*

is updated when a successful local move is made. On the one hand, the current solution is added to *NDS* if it is not dominated by any member of *NDS*. On the other hand, any members of *NDS* dominated by the current solution are removed from this set.

Step 3: Selection. As discussed in [11,6], optimal solutions obtained with similar weight vectors should be similar in the objective space and decision space. Cooperation between solutions with similar weighted sum functions can be very helpful for finding good approximations to the POF. Therefore, the local optima obtained in **Step 2** is very likely to be better than the solutions in the population with similar weight vectors. In this paper, we compare π with all $\pi^{(i)} \in P, i = 1, \ldots, N$. If $f(\pi|\lambda^{(i)}) < f(\pi^{(i)}|\lambda^{(i)})$, then set $\pi^{(i)} = \pi$. In mGRASP, solutions found in different iterations do not interact.

Step 4: Modification of Search Direction. Ideally, finding the optimal solutions of all weighted sum functions leads to a good approximation of the POF. However, this is impossible in mGRASP/MH since a population of fixed size is used. In [11], we have suggested an adaptive mechanism to tune the weight vector of each solution according to the locations of some solutions previously examined. In this mechanism, the non-dominated neighboring solution π' that is nearest to $\pi^{(i)}$ is identified. For each objective k, if $C^k(\pi') < C^k(\pi^{(i)})$, then decrease $\lambda_k^{(i)}$ by δ (> 0); otherwise, increase by δ. If $\lambda_k^{(i)}$ exceeds the bounds, then use the nearest bound to replace it. As a result, the optimal solution of the weighted sun function with the modified weight vector should be moved away from π' in the objective space. In such a way, the sparse part of POF can be explored more intelligently and efficiently. In this paper, we use this strategy in a slightly different manner. Each search direction is modified with a probability.

5 Computational Experiments

5.1 Performance Assessment

To quantitatively evaluate the non-dominated solutions found by each algorithm, we use both the generational distance (GD) metric and the inverted generational distance (IGD) metric. Assume S is the final set of non-dominated solutions found by multi-objective algorithm and S^* is a set of reference solutions, either the true POF or a very good approximation. The GD metric measures the average distance from S to S^*, while the IGD metric measures the average distance from S^* to S [14]. These two metrics can be formulated as follows: $\mathrm{GD}(S, S^*) = \frac{1}{|S|}\sum_{u \in S} \min\{dist(u, v)|v \in S^*\}$ and $\mathrm{IGD}(S^*, S) = \frac{1}{|S^*|}\sum_{u \in S^*} \min\{dist(u, v)|v \in S\}$, where $dist(u, v)$ is the Euclidean distance between two objective vectors. The smaller the GD or IGD values, the better quality of the set S. In this paper, the reference set for each instance is formed by collecting all non-dominated solutions found by five algorithms in 20 runs.

5.2 Experimental Settings

We used a set of 18 benchmark mQAP instances to test the performance of mGRASP/MH. These test instances were generated by Knowles [15] and are available at *http://dbkgroup.org/knowles/mQAP/*. The correlation values between flow matrices of these test instances are shown in Table 1.

Table 1. Correlations between the flows of the 18 benchmark mQAP test instances

Instance	$c(B^1, B^2)$	Instance	$c(B^1, B^2)$	Instance	$c(B^1, B^2)$,$c(B^1, B^3)$
KC10-2fl-1uni	0	KC20-2fl-1uni	0	KC30-3fl-1uni	(0, 0)
KC10-2fl-2uni	0.8	KC20-2fl-2uni	0.7	KC30-3fl-2uni	(0.4,0.4)
KC10-2fl-3uni	-0.8	KC20-2fl-3uni	-0.7	KC30-3fl-3uni	(-0.4, -0.4)
KC10-2fl-1rl	0	KC20-2fl-1rl	0	KC30-3fl-1rl	(0.4, 0)
KC10-2fl-2rl	0.7	KC20-2fl-2rl	0.4	KC30-3fl-2rl	(0.7, -0.5)
KC10-2fl-3rl	-0.7	KC20-2fl-3rl	-0.4	KC30-3fl-3rl	(-0.4, -0.4)

We compared mGRASP/MH to mGRASP and to three state-of-the-art EMO algorithms - MOEA/D, NSGA2, and SPEA2. In MOEA/D, the mQAP is converted into a number of single objective subproblems. These subproblems are optimized by an evolutionary algorithm simultaneously. The best solutions to all subproblems found so far are retained in its population. The distribution of these solutions is controlled by the diversity of weight vectors. Each offspring solution in MOEA/D is improved by local search. In both NSGA2 and SPEA2, the non-dominated solutions found so far have priority to survive in the population. The diversity of these non-dominated solutions is maintained by estimating their density. In this paper, we use cycle crossover [16] and mutation based on the 2-opt swap for the MOEA/D, NSGA2, and SPEA2 algorithms.

In both mGRASP and mGRASP/MH, α is set to 0.1. Parameter β is set to 0.5. That is, half of the components in elite solutions are copied to the construction procedure of mGRASP/MH. The population size (N) in mGRASP/MH is 50 for all instances. The δ value for changing weight is 0.01. The population size in NSGA2, SPEA2, and MOEA/D is 100. In MOEA/D, the neighborhood size of each subproblem is 20 for all test instances.

We run each algorithm on each instance 20 times. All algorithms are coded in C++ and executed on a PC with CPU (Intel (R) Core (TM) 2, 1.86GHZ) and RAM (2GB). Every algorithm uses the same computational time for the same test instance. The computational times used for the instances with 10, 20, and 30 locations are set to 10, 20, and 30 seconds, respectively.

5.3 Discussions of Results

The mean GD and IGD values found by the five algorithms are summarized in Table 2 and Table 3. It is evident that mGRASP/MH and MOEA/D clearly outperform the other three algorithms on all test instances. Among the five algorithms, NSGA2 and SPEA2 show the worst performance with respect to

Table 2. The mean GD values of non-dominated solutions found in 20 runs

Instance	mGRASP/MH	mGRASP	MOEA/D	NSGA2	SPEA2
KC10-2fl-1uni	0	592	1730	4462	6152
KC10-2fl-2uni	5305	0	5490	11800	13845
KC10-2fl-3uni	0	1	111	1357	2893
KC10-2fl-1rl	0	1129	22132	236966	321468
KC10-2fl-2rl	22086	16300	34471	157128	151661
KC10-2fl-3rl	0	1129	14979	244293	285310
KC20-2fl-1uni	9225	21758	11269	48813	53635
KC20-2fl-2uni	9138	58660	16364	65180	61904
KC20-2fl-3uni	3758	6966	4934	22133	29537
KC20-2fl-1rl	580688	2069384	509229	2996725	2565999
KC20-2fl-2rl	205812	1124948	155082	1372892	1117776
KC20-2fl-3rl	168651	476440	145244	1194632	1251489
KC30-3fl-1uni	41072	55178	18945	132735	163554
KC30-3fl-2uni	64156	111067	26085	153182	156566
KC30-3fl-3uni	30308	36855	14684	94685	123557
KC30-3fl-1rl	1302906	2491688	302268	3264761	3667731
KC30-3fl-2rl	877695	1931606	297531	3038431	3281139
KC30-3fl-3rl	917218	1427153	313038	3450839	3880325

Table 3. The mean IGD values of non-dominated solutions found in 20 runs

Instance	mGRASP/MH	mGRASP	MOEA/D	NSGA2	SPEA2
KC10-2fl-1uni	7	460	2211	6590	7795
KC10-2fl-2uni	4715	0	4915	11284	13196
KC10-2fl-3uni	0	6	147	2393	4387
KC10-2fl-1rl	266	3555	45512	318513	382993
KC10-2fl-2rl	8414	10460	128988	212026	226922
KC10-2fl-3rl	14	2403	37239	300822	357818
KC20-2fl-1uni	8509	21360	12058	53492	58575
KC20-2fl-2uni	10500	58830	16987	66425	64604
KC20-2fl-3uni	3526	6677	4878	35764	44289
KC20-2fl-1rl	467232	1980738	433020	2914559	2623621
KC20-2fl-2rl	280650	1259521	192956	1895681	1520627
KC20-2fl-3rl	205030	653760	153859	1594337	1534329
KC30-3fl-1uni	38396	54552	20578	141325	167422
KC30-3fl-2uni	63583	110308	26415	161061	163284
KC30-3fl-3uni	29342	36927	16133	127932	154106
KC30-3fl-1rl	1519861	3350333	474028	5962007	7018525
KC30-3fl-2rl	1062987	2837723	421962	4538068	4986717
KC30-3fl-3rl	974208	1658247	395310	4072323	4503648

minimizing the GD and IGD values. The main reason for this might be that no local search is used to improve offspring solutions in these two approaches.

The non-dominated solutions found by all five algorithms after 20 runs on the four 2-objective instances with zero correlation between flow matrices are

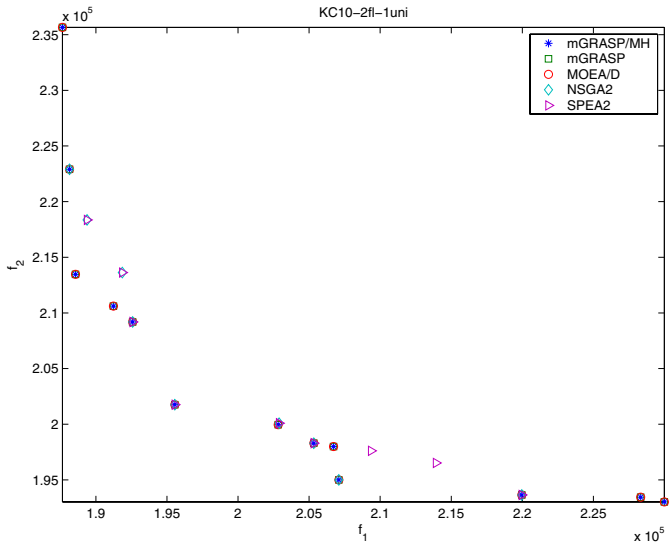

Fig. 4. Non-dominated solutions found by mGRASP/MH, mGRASP, MOEA/D, NSGA2, and SPEA2 on KC10-2fl-1uni in 20 runs

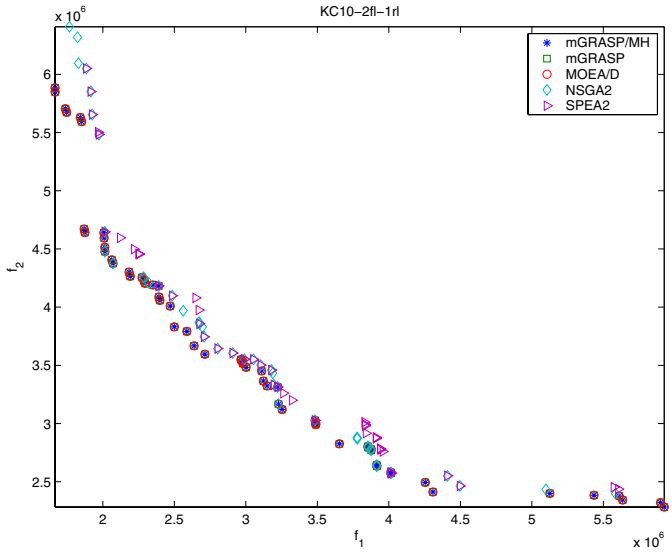

Fig. 5. Non-dominated solutions found by mGRASP/MH, mGRASP, MOEA/D, NSGA2, and SPEA2 on KC10-2fl-1rl in 20 runs

plotted in Figs. 4-7. It can be observed from Fig. 4 that all five algorithms find almost the same set of non-dominated solutions on instance KC10-2fl-1uni. The results in Fig. 6 and Fig. 7 show that both mGRASP/MH and MOEA/D clearly perform better than mGRASP on KC20-2fl-1uni and KC20-2fl-1rl. Figs. 5-7

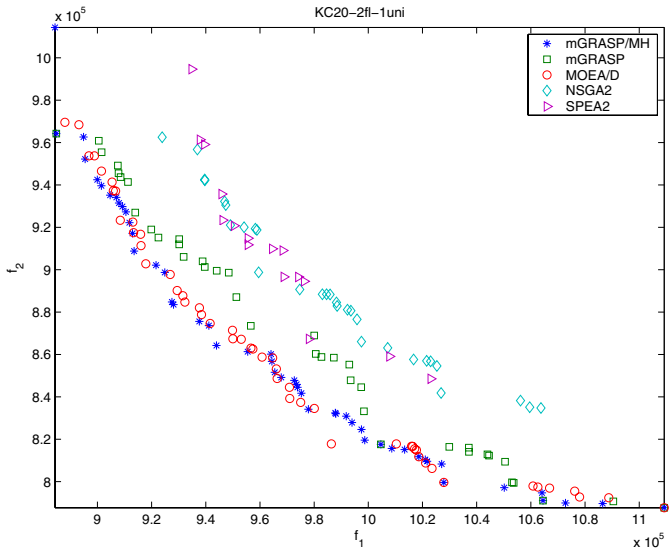

Fig. 6. Non-dominated solutions found by mGRASP/MH, mGRASP, MOEA/D, NSGA2, and SPEA2 on KC20-2fl-1uni in 20 runs

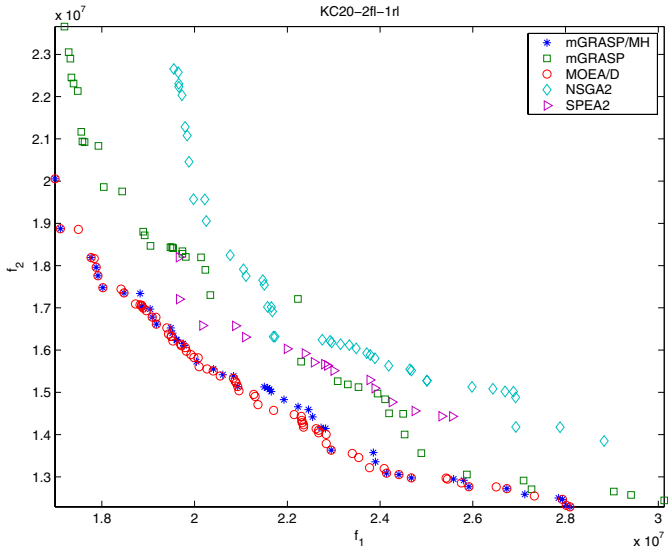

Fig. 7. Non-dominated solutions found by mGRASP/MH, mGRASP, MOEA/D, NSGA2, and SPEA2 on KC20-2fl-1rl in 20 runs

show that three local search-based metaheuristics - mGRASP/MH, mGRASP, and MOEA/D, find better solutions than two Pareto-based EMO algorithms - NSGA2 and SPEA2 on three instances - KC10-2fl-1rl, KC20-2fl-1uni, and KC20-2fl-1rl.

Results in Table 2 and Table 3 on six 3-objective test instances show that both mGRASP/MH and MOEA/D perform better than the other three algorithms in terms of the GD and IGD metrics. It can also be seen that mGRASP/MH finds the worse GD and IGD values than MOEA/D on these instances. It is easy to understand the reason behind the worse performance of mGRASP/MH. The greedy randomized construction procedure in mGRASP/MH has higher computational complexity than the crossover and mutation operators used in MOEA/D. Within the restricted computational time, MOEA/D could improve more new solutions by local search. This is also part of the reason that mGRASP/MH performs better than mGRASP. The former only builds half of the starting solution by greedy randomized construction procedure. Therefore, mGRASP/MH needs less time in the construction of a starting solution.

6 Conclusions

We proposed an elitist GRASP metaheuristic algorithm called mGRASP/MH to tackle the mQAP (multi-objective quadratic assignment problem). In the proposed approach, elitist-based greedy randomized construction, cooperation between solutions, and weight-vector adaptations are used to accelerate convergence and diversify the search. Our experimental results show that mGRASP/MH is competitive with MOEA/D and outperforms mGRASP and two Pareto-based EMO algorithms - NSGA2 and SPEA2 on the benchmark problem instances considered here. It has also been shown that the multi-objective metaheuristic algorithms using local search perform better than those without local search for the mQAP.

In this paper, the construction of starting solutions copies parts or components from elite solutions. Under the framework of mGRASP/MH, it is very easy to use other advanced techniques, such as guided mutation [8], cooperative strategy [17,18] and path relinking [19]. Complex memory structure for storing historical information from the search, probability distributions in guided mutation, all these should benefit the global search ability of mGRASP/MH. The cooperation between solutions obtained by different local search procedures can be implemented by considering path relinking [19] or tabu mechanisms [18]. These are some of our future research directions.

Acknowledgement. The work was funded by the UK's EPSRC, under grant EP/E019781/1. The authors would like to thank anonymous reviewers for their helpful and constructive comments.

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Series of Books in the Mathematical Sciences. W. H. Freeman, New York (1979)
2. Knowles, J.D., Corne, D.W.: Towards landscape analyses to inform the design of hybrid local search for the multiobjective quadratic assignment problem. In: Soft Computing Systems - Design, Management and Applications (HIS 2002), pp. 271–279 (2002)

3. Paquete, L., Stützle, T.: A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. European Journal of Operational Research 169(3), 943–959 (2006)
4. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(2), 182–197 (2002)
5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100 (2001)
6. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evolutionary Computation 11(6), 712–731 (2007)
7. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1998)
8. Zhang, Q., Sun, J.: Iterated local search with guided mutation. In: The 2006 IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver (2006)
9. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8, 67–71 (1989)
10. Vianna, D.S., Arroyo, J.E.C.: A GRASP algorithm for the multi-objective knapsack problem. In: The First International Conference on Quantitative Evaluation of Systems. IEEE Computer Society, Los Alamitos (2004)
11. Li, H., Landa-Silva, J.D.: Evolutionary multi-objective simulated annealing with adaptive and competitive search direction. In: The 2008 IEEE Congress on Evolutionary Computation (CEC 2008), Hong Kong, pp. 3310–3317. IEEE Press, Los Alamitos (2008)
12. Taillard, É.D.: Robust taboo search for the quadratic assignment problem. Parallel Computing 17(4-5), 443–455 (1991)
13. Resende, M.G.C.: Metaheuristic hybridization with GRASP. In: Chen, Z.L., Raghavan, S. (eds.) Tutorials in Operations Research, INFORMS (2008)
14. Czyzak, P., Jaszkiewicz, A.: Pareto-simulated annealinga metaheuristic technique for multi-objective combinatorial optimization. J. Multi-Criteria Decis. Anal. 7(1), 34–47 (1998)
15. Knowles, J.D., Corne, D.W.: Instance generators and test suites for the multiobjective quadratic assignment problem. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 295–310. Springer, Heidelberg (2003)
16. Oliver, I.M., Smith, D.J., Holland, J.R.C.: A study of permutation crossover operators on the traveling salesman problem. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, Hillsdale, NJ, USA, pp. 224–230. L. Erlbaum Associates Inc., Mahwah (1987)
17. Burke, E.K., Landa-Silva, J.D.: The influence of the fitness evaluation method on the performance of multiobjective search algorithms. European Journal of Operational Research 169(3), 875–897 (2006)
18. Landa-Silva, J.D., Burke, E.K.: Asynchronous cooperative local search for the office space allocation problem. INFORMS Journal on Computing 19(4), 575–587 (2007)
19. Oliveira, C.A.S., Pardalos, P.M., Resende, M.G.C.: GRASP with path-relinking for the quadratic assignment problem. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 356–368. Springer, Heidelberg (2004)