

Instance Generators and Test Suites for the Multiobjective Quadratic Assignment Problem

Joshua Knowles^{1*} and David Corne²

¹ IRIDIA - CP 194/6, Université Libre de Bruxelles, Avenue F D Roosevelt, 50, 1050 Brussels, Belgium. E-mail: jknowles@ulb.ac.be

² School of Systems Engineering, University of Reading, Reading RG6 6AY, UK. E-mail: d.w.corne@reading.ac.uk

Abstract. We describe, and make publicly available, two problem instance generators for a multiobjective version of the well-known quadratic assignment problem (QAP). The generators allow a number of instance parameters to be set, including those controlling epistasis and inter-objective correlations. Based on these generators, several initial test suites are provided and described. For each test instance we measure some global properties and, for the smallest ones, make some initial observations of the Pareto optimal sets/fronts. Our purpose in providing these tools is to facilitate the ongoing study of problem structure in multiobjective (combinatorial) optimization, and its effects on search landscape and algorithm performance.

Keywords

Quadratic assignment problem, multiobjective combinatorial optimization, fitness landscapes, inter-objective correlations, flow-dominance, distance-dominance

1 Introduction

Configuring a search metaheuristic to work effectively on a specific problem class or instance depends upon being able to relate measurable properties of the problem to performance predictions of the metaheuristic and its configuration. In single objective optimization, some fairly general, measurable properties of problems and/or search ‘landscapes’, have already been proposed and analysed in the literature (e.g. see [15]), and progress towards a science of heuristic search is under way [7]. In *multiobjective* optimization (MOO), however, problem and landscape structures have additional degrees of freedom, necessitating the consideration of other unique factors. For example, in MOO, fitness landscapes have multiple ‘vertical’ dimensions and the desired optima form a Pareto front (PF) that can vary in dimension, cardinality, extent, connectedness, and convexity. Consequently, search algorithms operating on these landscapes have more potential modes of operation: when trying to obtain the PF, many search paths are possible but some will be far more efficient than others. It is therefore, perhaps,

* <http://iridia.ulb.ac.be/~jknowles/>

even more important to understand how to configure MOO metaheuristics than in the case of single objective optimization.

To facilitate empirical studies of problems/landscapes and their relationships to algorithm/configuration performance, test suites and problem generators are useful tools. Arguably, generators are better (particularly those with many controllable parameters) because they enable the effects of different properties to be investigated in isolation or in groups, whereas test suites (alone) can often be too limited in the instances, which can lead to a false impression of progress while, in fact, important problem properties remain uncharted.

In the field of evolutionary multiobjective optimization (EMOO), some good generators and test suites already exist. The frameworks proposed by Deb et al. [4–6] have been widely appreciated, although the test suite proposed in [21] had some drawbacks that have, arguably, somewhat constrained progress. The knapsack problems, originally used in [22] have also become a popular choice for benchmarking algorithms. Once again, there are advantages and difficulties with the popularity of this suite, however. It is advantageous for researchers to have a common standard of comparison and it is encouraging to see multiobjective combinatorial optimization (MOCO) problems being tackled in the EMOO literature, but the knapsack problems are limited for two main reasons: 1. a generator is not publicly available and the instances provided do not have many varying parameters, and 2. because, being a constrained problem, heuristic repair or other mechanisms must be introduced, affecting the landscape ‘seen’ by a metaheuristic, and clouding the important issue of measuring algorithm performance. Other MOCO problems exist in the literature, including our own mc-MST problems [12], but few have become useful benchmarks for investigating general problem characteristics.

Much more extensive study of problems and algorithm performance has been carried out in relation to single objective combinatorial problems. After the traveling salesman problem, perhaps the most studied of all is the quadratic assignment problem (QAP). QAP is both practically important and very difficult, making it particularly relevant for approximate search. In a recent paper [11] we proposed a multiobjective version of the QAP, where $m \geq 2$ distinct QAPs must be minimized simultaneously over the same permutation space. We believe the problem to have practical applications but our main purpose in proposing it is the opportunity it may provide for more general understanding of multiobjective combinatorial optimization (MOCO). There are several advantages to the QAP as a candidate to become a useful test-bed in multiobjective optimization. It has a very simple formulation, solutions being permutations of the integers from 1.. n , so that specialized heuristics and/or repair mechanisms are not needed (or used) to tackle the problem as, for example, they are in knapsack and graph-based problems. The objective function is fast to compute and it can also be delta-evaluated, enabling local search to be efficiently applied [19]. Furthermore, much of the knowledge about the problem, including global measures and landscape analysis tools can be adapted from the vast QAP literature, and ‘imported’ into the multiobjective domain.

Our aim here is to encourage study of the multiobjective QAP (mQAP), and of MOCO problems in general, through the provision of two mQAP instance generators which allow some important parameters to be controlled and investigated. Some initial test suites derived from these are also given to illustrate the effects of some parameters and to facilitate algorithm performance comparisons. Properties and measures for QAP instances are briefly discussed and, for the test suites, some simple measures are applied. For the smallest instances we compute the entire search space and present some observations of the Pareto front structures. It is our aim in future work to make much further investigations of this problem, and we have already begun this work in [11]. But our focus here is on providing what we hope is a useful new MOCO problem for the EMOO field in a form that makes it easy to use.

The rest of this paper is organized as follows. In section 2 we briefly review some of the literature related to the QAP, focusing on available instances, global properties and landscape measures. Section 3 describes the mQAP and reviews some other MOCO problems and what is known about their problem/landscape structures. Section 4 introduces our generators, while section 5 presents some test suites and provides some initial measures on some instances. Section 6 concludes.

2 The Scalar QAP

Many diverse planning tasks of practical importance can be formulated as instances of the *quadratic assignment problem* (QAP), an NP hard [17] combinatorial optimization problem that dates back to the early sixties. The QAP is a broad problem class embracing both the *graph partitioning problem* and *travelling salesman problem* as special cases [14]. It is also an unusually difficult problem, where even relatively small ($n \geq 20$) general instances cannot be solved to optimality, and it has thus been important in stimulating research in approximate methods. The great research effort attracted by the problem means that a relatively large amount is now known about QAP instance structures and how this relates to global and local statistical measures of their fitness landscapes. This knowledge has been put to effective use in designing and tuning metaheuristics for this problem, e.g. [19, 18, 14].

The quadratic assignment problem (QAP) entails the assignment of n facilities to n locations so as to minimize a sum of flow/distance products. It may be formulated as:

$$\text{Minimize } C(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi_i \pi_j} \quad (1)$$

where n is the number of facilities/locations, a_{ij} is the distance between location i and location j , b_{ij} is the flow from facility i to facility j , and π_i gives the location of facility i in permutation $\pi \in P(n)$ where $P(n)$ is the QAP search space: the set of all permutations of $\{1, 2, \dots, n\}$.

2.1 Instances and generators

A number of instances of the QAP are publicly accessible from QAPlib [3], and come in several categories: those with uniformly random distance and flow matrix entries; those that derive from real applications, which are generally more structured and where typically some of the off-diagonal flow matrix entries are zero; and, because the latter are quite small, random ‘real-like’ instances [19] that have been artificially constructed using a parameterized generator. The real-like instances are more practically interesting than the uniformly random instances and although it is easier to find the optimum (or best known solution) of them on small instances, they are generally more difficult in terms of reaching a given % above this value. The reason for this effect is that in the uniform instances there are far more solutions at a given % cost above the best-known (easy to find), but it is then difficult to find which of these is close (in permutation space) to the best-known solution, making it difficult to direct the search. These properties can be related to the ‘dominance’ of the flow and distance matrices, defined below.

2.2 Measures of QAP instances and landscapes

Several papers on QAP have conjectured that certain (measurable) characteristics of QAP instances can be related to measurable properties of their fitness landscapes. To merely *characterize* an instance it is sufficient to describe some properties of the distance and flow matrices, a computationally tractable task. To measure some property of the fitness landscape, however, it is necessary (for all but very small instances) to sample the search space. Sampling approaches can be broadly divided into global and local methods. The former operate by taking arbitrary solutions (e.g. uniformly at random) and measuring some properties such as parameter correlations between these. These global methods have been criticized, however, because when performing optimization, little time should be spent sampling these ‘average’ points [15]. Therefore, it may be better to use local measures that are based on biased sampling.

Vollmann and Buffa [20] introduced flow dominance as a basic means of characterizing QAP instances. Flow dominance is a measure of the flow matrix, A , given by

$$fd(A) = 100 \frac{\alpha}{\beta}, \quad \text{where } \alpha = \sqrt{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (a_{ij} - \beta)^2} \quad \text{and } \beta = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n a_{ij}. \quad (2)$$

When there is high flow dominance there is low epistasis, in general [14]. The distance dominance dd can be defined on the distance matrix in an analogous fashion.

Measures of the search landscape (whether global or local) often depend on a measure of distance between solutions in the parameter space. Bachelet [1] measures distance $\text{dist}(\pi, \mu)$ between two QAP solutions π and μ as the smallest number of 2-swaps that must be performed to transform one solution into the

other (this can be computed in $O(n)$ time); this distance measure has a range of $0..n - 1$.

From this, other measures can be easily defined. Bachelet gives the diameter of a population of solutions P as

$$\text{dmm}(P) = \frac{\sum_{\pi \in P} \sum_{\mu \in P} \text{dist}(\pi, \mu)}{|P|^2}. \quad (3)$$

The entropy [9], which is a further measure of the dispersion of solutions, is given by

$$\text{ent}(P) = \frac{-1}{n \log n} \sum_{i=1}^n \sum_{j=1}^n \left(\frac{n_{ij}}{|P|} \log \frac{n_{ij}}{|P|} \right) \quad (4)$$

where n_{ij} is the number of times facility i is assigned to location j in the population. Low values of entropy indicate highly clustered solutions; high values that they are more randomly distributed.

These measures can be used on a random sample of points (i.e. as global measures) but it is also possible to measure these properties on local optima, or on, say, the fittest $p\%$ of points found during an optimization. The latter can give a better picture of how desirable solutions are distributed in the search space.

The fitness-distance correlation [10] has also been widely used. Often a plot is shown, giving the fitness against distance from the nearest global optimum or best known solution.

3 The Multiobjective QAP Model

The multiobjective QAP (mQAP), with multiple flow matrices (defined below) naturally models any facility layout problem where we are concerned with the flow of more than one type of item or agent. For example, in a hospital layout problem we may be concerned with simultaneously minimizing the flows of doctors on their rounds, of patients, of hospital visitors, and of pharmaceuticals and other equipment.

3.1 Problem definition

The mQAP may be formulated as follows:

$$\text{'minimize' } \mathcal{C}(\pi) = \{C^1(\pi), C^2(\pi), \dots, C^m(\pi)\} \quad (5)$$

where

$$C^k(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi_i \pi_j}^k, \quad k \in 1..m \quad (6)$$

and where n is the number of facilities/locations, a_{ij} is the distance between location i and location j , b_{ij}^k is the k th flow from facility i to facility j , π_i gives the location of facility i in permutation $\pi \in P(n)$, and finally, 'minimize' means to obtain the Pareto front, or an approximation of it.

3.2 Fitness landscapes in MOCO

Although a fairly wide variety of MOCO problems have been defined and tackled in the literature (see [8] for a survey), little work has attempted to characterize the search landscapes of these problems. A rare and valuable foray into this area is [2], where the property of ‘global convexity’ in bi-objective TSPs was investigated; the distances in objective space between solutions near the PF were found to correlate with parameter space distances. One of the implications of this study was that applying wholly separate runs of a single-objective metaheuristic may not be as effective as making use of information from ‘nearby’ points in the objective or weight space. With the mQAP we expect global convexity to be less marked but similar techniques could be used to test this hypothesis, and to observe how global convexity varies with instance type.

The knapsack problems introduced in [22] have been used as benchmarks for studying the performance of various metaheuristics. Generalizing over a number of results, it now appears clear that population-based methods using only Pareto selection have the tendency to concentrate their solutions on the centre of the PF. This is the case, even when using advanced strategies such as the BMOA [13] that also incorporate modern archiving methods. This ‘central tendency’ may be due to a clustering in the parameter space of the central PF solutions, with the extremes more isolated. Further investigations of this tendency and its relation to MOCO problem structure is needed.

Finally, we have observed previously [12] that correlations between the weights making up different objectives in a multiobjective MST problem can strongly affect the PF shape, and this, in turn, has an effect on the success of weighted-sum based approaches. The introduction of the mQAP generators (described next) is intended to facilitate further study of the above relationships, and to this end we have also outlined some methods [11], making use of a fast local search for the QAP, aimed eventually at answering the following question: what problem features affect the relative difficulty of approaching the PF vs moving along it? This, we believe impacts strongly on the most appropriate choice of search strategy.

4 Instance Generators for the mQAP

One approach to obtaining instances of the mQAP (or any other MOCO problem) is simply to use and concatenate available single-objective instances. This approach has the advantage of allowing comparison between single-objective and multiobjective algorithm performance on the problem. The extremes of the Pareto front—or best known single-objective solutions—would also be known, although *only* the extremes. We do not criticize this approach and would encourage more researchers to use it (being explicit about where the instances came from). However, some aspects of multiobjective problems cannot be easily controlled by concatenating single-objective instances. In particular, we would like to control the correlations between corresponding components (flows in our case) of the different objectives of a problem instance. This is desirable because

Algorithm 1 Real-like mQAP instance generator with correlated flow elements and overlap parameter

```

1: Input:  $n \in \mathbb{Z}^+$ ,  $m \in \mathbb{Z}^+$ ,  $c[1]..c[m] \in [-1, 1]$ ,  $\eta \in [0, 1]$ ,  $A \in \mathbb{Z} < B \in \mathbb{Z}^+$ ,
    $r_{\max} \in \mathbb{Z}^+$ ,  $R_{\max} \in \mathbb{Z}^+$ ,  $N_{\max} \in \mathbb{Z}^+$ ,  $seed \in \mathbb{Z}^+$ 
2:  $i \leftarrow 1$ 
3: while  $i < n$  do
4:    $\Theta \leftarrow \mathcal{U}[0, 2\pi)$ ,  $R \leftarrow \mathcal{U}[0, R_{\max})$ ,  $N \leftarrow \lfloor \mathcal{U}[1, N_{\max} + 1) \rfloor$ 
5:   for  $j \leftarrow 1..N$  do
6:      $\theta \leftarrow \mathcal{U}[0, 2\pi)$ ,  $r \leftarrow \mathcal{U}[0, r_{\max})$ 
7:     if  $i < n$  then
8:        $loc[i] \leftarrow (R \cos \Theta + r \cos \theta, R \sin \Theta + r \sin \theta)$ 
9:        $i \leftarrow i + 1$ 
10:    end if
11:  end for
12: end while
13: for  $i \leftarrow 1..n$  do
14:   for  $j \leftarrow 1..n$  do
15:     $dmatrix[i][j] \leftarrow \text{Euclid}(loc[i], loc[j])$ 
16:   end for
17: end for
18:  $\text{print2d}(dmatrix)$ 
19: for  $i \leftarrow 1..n$  do
20:   for  $k \leftarrow 1..m$  do
21:     $fmatrix[k][i][i] \leftarrow 0$ 
22:   end for
23: end for
24: for  $i \leftarrow 1..n - 1$  do
25:   for  $j \leftarrow i + 1..n$  do
26:    for  $k \leftarrow 1..m$  do
27:     if  $k = 1$  then
28:        $R1 \leftarrow \mathcal{U}[0, 1)$ 
29:        $fmatrix[k][i][j] \leftarrow fmatrix[k][j][i] \leftarrow \lfloor 10^{(B-A) \cdot R1 + A} \rfloor$ 
30:     else
31:        $R2 \leftarrow \mathcal{U}[0, 1)$ 
32:       if  $R2 > \eta$  then
33:         if  $fmatrix[1][i][j] = 0$  then
34:            $R2 \leftarrow \mathcal{U}[0, 1)$ 
35:            $fmatrix[k][i][j] \leftarrow fmatrix[k][j][i] \leftarrow \lfloor 10^{(B \cdot R2)} \rfloor$ 
36:         else
37:            $fmatrix[k][i][j] \leftarrow fmatrix[k][j][i] \leftarrow 0$ 
38:         end if
39:       else
40:          $V \leftarrow \text{correl\_val}(R1, c[k])$  /* see equation (7) */
41:          $fmatrix[k][i][j] \leftarrow fmatrix[k][j][i] \leftarrow \lfloor 10^{(B-A) \cdot V + A} \rfloor$ 
42:       end if
43:     end if
44:   end for
45: end for
46: end for
47: for  $k = 1$  to  $m$  do
48:    $\text{print2d}(fmatrix[k])$ 
49: end for

```

realistic problems are likely to exhibit such inter-objective correlations, and describing how these affect the search landscape should be a key element of any useful exposition on MOCO problem/landscape relationships.

4.1 Uniformly random instance generator

In the first of our generators, `makeQAPuni.cc`¹, only one inter-objective correlation can be controlled. The generator makes symmetric² QAP instances with one distance and multiple flow matrices, and its basic parameters are for the instance size n and number of objectives m . All flows and distances are integers in $1..f_{\max}$ and $1..d_{\max}$, respectively where f_{\max} and d_{\max} are two further parameters. The desired correlation between corresponding entries in the first and all other flow matrices, is set using the parameter c . Correlated random variables are generated using:

$$p(r^k) = \mathcal{N}(r^1, 1 - \sqrt{c}) / ((1 - \sqrt{c}) \cdot \sqrt{2\pi}), \quad k \in 2..m \quad (7)$$

where $p(r^k)$ is the probability of accepting a random variable $r^k \in [0, 1)$ given the value of a uniform random variable, $r^1 \in [0, 1)$, and $\mathcal{N}(\bar{x}, \sigma^2)$ is a normal distribution with mean \bar{x} and variance σ^2 . The actual flows are made from the random variables using:

$$b^k = 1 + \lfloor r^k \cdot b \rfloor, \quad k \in 1..m \quad (8)$$

The generator makes the flow entries in all flow matrices, one at a time using the above procedure. Pseudocode for the generator is not given here but follows a similar structure as that for the real-like generator described next.

4.2 Real-like instance generator

The real-like instance generator, `makeQAPr1.cc`¹, makes instances where the distance and flow matrices have structured entries. The generator follows procedures for making the non-uniformly random QAP problems given the appellation TaiXXb in the literature, outlined in [19]. Pseudocode for the generator is presented in Algorithm 1.

The distance matrix entries generated are the Euclidean distances between points in the plane. The points are randomly distributed in small circular regions, with these regions distributed in a larger circle. The size and number of the small and larger circles can be controlled by the parameters $r_{\max} \in \mathbb{Z}^+$, $R_{\max} \in \mathbb{Z}^+$, and $N_{\max} \in \mathbb{Z}^+$.

The flow entries are non-uniform random values, controlled by two parameters, $A \in \mathbb{Z}$ and $B \in \mathbb{Z}^+$, with $A < B$. Let X be a random variable uniformly distributed in $[0, 1)$. Then a flow entry is given by

$$\lfloor 10^{((B-A)*X+A)} \rfloor. \quad (9)$$

¹ Available from <http://iridia.ulb.ac.be/~jknowles/mQAP/> . C code for reading instances is also provided

² Without loss of generality, since an asymmetric matrix can always be transformed into a symmetric one [14]

With negative values of A the flow matrix is sparse, i.e. it contains a number of off-diagonal zero entries. The non-zero entries have non-uniformly distributed values. Different values of A and B cannot be set for each different flow matrix but this is an easy extension that might later be added.

The entries in the k th flow matrix ($2 \leq k \leq m$) are generated using (9) but the random variable X is correlated with the value of X that was used in the corresponding entry in the first flow matrix. Here, correlations (in $[-1, 1]$) can be set between the first and *each* of the additional flow matrices using $m - 1$ further parameters to the generator.

A degree of ‘overlap’ between the matrices can also be specified using a parameter $\eta \in [0, 1]$. It controls the fraction of entries in the j th flow matrix that are correlated with the corresponding entries in the 1st flow matrix. With the overlap parameter $\eta = 0$, a random un-correlated value, calculated using

$$\lfloor 10^{(B * X)} \rfloor. \quad (10)$$

will be placed in each entry of the j th flow matrix that corresponds to a zero entry in the first flow matrix. Using this, (and not (9)) ensures that the flow is non-zero. Conversely, a zero will be placed in each entry of the j th flow matrix that corresponds to a non-zero value in the first flow matrix. Thus there is no overlap between the flows of the first and j th matrix when $\eta = 0$. With the $\eta = 1$ all the flows overlap and are correlated. With the overlap set to intermediate values some of the flows will overlap and others will not.

5 Test Suites

Table 1 describes three 10 facility, 2 flow matrix instances, produced by the uniform instance generator. All off-diagonal distances and flows in the matrices are from the set 1..100. The instances differ in the correlations between corresponding elements in the first and second flow matrix. All instances have very similar flow and distance dominance values so the epistasis present (when considering just one flow matrix at a time) should be about the same. However, the different correlations in the instances may affect difficulty for two reasons. First, the shape of the Pareto front may be changed. For example, in [12] it was found that positive correlations between weights in mc-MST problems led to more convex and smaller Pareto fronts, whereas negative correlations led to large, flat Pareto fronts. The latter were more difficult for weighted sum-based approaches in the sense that many weight vectors (in a naïve approach where these are uniformly randomly generated) tend to push the search to either extreme of the PF, making it difficult to find the intermediate optima. The mQAP instances here should allow these correlation effects to be isolated and studied further. The difficulty of moving toward the Pareto front may also be affected by the correlation. One might guess that if the correlation between objectives is strongly positive then the problem reduces to the single objective one. Whereas if there is little, or even negative correlation, then there should exist more optima and they are more likely to be spread out in the space. Therefore the search to find at least

Param/Property	Parameter/Property Values by Instance		
	KC10-2fl-1uni	KC10-2fl-2uni	KC10-2fl-3uni
$\max(d)$	100	100	100
$\max(f)$	100	100	100
$\text{corr}(f^1, f^2)$	0.0	0.8	-0.8
Spearman r	0.18	0.92	-0.75
fd^1, fd^2	76.7, 69.9	66.1, 63.2	64.8, 69.8
dd	57.9	622.3	69.8
#PO	27	4	135
#supported	7	3	14
diam(PO)	7	6	8
ent(PO)	0.71	0.39	0.78
seed	68203720	289073914	73892083

Table 1. Test suite of three 10 node, 2 flow matrix, uniformly random instances with different flow correlations and other parameters. The first figure in an instance name is the number of nodes, the second is the number of flow matrices and the third is just an index. The global parameters/properties are: $\max(d)$, the maximum distance in the distance matrix; $\max(f)$, the maximum flow in any of the flow matrices; $\text{corr}(f^i, f^j)$, the correlation parameter affecting corresponding flow matrix entries of the i th and j flow; Spearman r , the measured sample rank correlation between (off-diagonal) corresponding entries in the first and j th flow matrix; fd^j , the flow dominance of the j th flow matrix; dd , the distance dominance; #PO, the number of Pareto optima; #supported the number of supported Pareto optima; diam(PO), the diameter of the Pareto optimal set; ent(PO), the entropy of the Pareto optimal set. The random seed to the generator is also given for reference

one (but not a specific one) may be easier. The correlation parameter used to set the desired correlation may be compared with the value, given in the table, of the Spearman rank correlation measure [16] of the actual flow entries.

Some of the correlation effects can also be appreciated by observation of the values of the other measures given. The number of Pareto optima (found by exhaustive search) is smaller when the correlation is large and larger when it is negative. Similarly, the diameter and entropy of the Pareto optima vary with inter-objective correlation. From the values we see that the optima are in fewer, larger clusters (in parameter space) when the correlation is positive and more spread out, and less clustered when the correlation is negative. The number of supported solutions (those lying on the convex hull of the Pareto front) is small in these instances, indicating that it may be difficult for a weighted sum approach to find all the Pareto optima. However, this is an artifact of the problems being very small; with larger instances the number of unsupported Pareto optima would generally be smaller as the PF assumes a smoother shape due to the ‘central-limit’ effect from summing larger numbers of flow/distance products.

Five small, two objective instances of the real-like problems are presented in Table 2. Similar observations apply to these problems. The Spearman r values correlate well with the correlation parameter setting for the first three instances. However, when the overlap parameter is set to 0.6, in instances 4 and 5, the r values drop significantly, as one would expect. Looking at the number of Pareto

Param/Prop	Parameter/Property Values by Instance				
	KC10-2fl-1rl	KC10-2fl-2rl	KC10-2fl-3rl	KC10-2fl-4rl	KC10-2fl-5rl
$\max(d)$	155	111	145	136	138
$\max(f)$	9445	9405	9732	9419	95476
A	-2	-2	-2	-2	-5
B	4	4	4	-4	5
$\text{corr}(f^1, f^2)$	0.0	0.7	-0.7	0.7	0.7
overlap η	1.0	1.0	1.0	0.6	0.6
fd^1, fd^2	243.0, 194.1	230.8, 242.8	248.9, 219.6	304.7, 272.1	405.3, 561.99
dd	69.5	60.7	65.6	58.6	58.6
Spearman r	0.15	0.83	-0.77	0.39	-0.10
#PO	38	17	58	33	48
#supported	13	3	25	12	14
diam(PO)	8	7	8	8	8
ent(PO)	0.68	0.49	0.62	0.58	0.63
seed	35243298	18178290	7810398	48972324	2129704715

Parameter/Property Values Common to All Instances

$$N_{\max} = 1, r_{\max} = 100, R_{\max} = 0$$

Table 2. Test suite of five 10 node, 2 flow matrix, real-like instances with different flow correlations and other parameters. A and B control the distribution of flow values, in particular the fraction of off-diagonal zero entries, thus they influence the flow dominance and epistasis. The overlap parameter η indicates the fraction of entries in the first flow matrix that are correlated with the corresponding entries in the other flow matrices. The distance matrices of all these instances are the Euclidean distances between random points in a single circle of radius 100. Other parameters are as in Table 1

optima, and their entropy and diameter values, we see that the r value explains these quite well. For the fifth instance, which has the largest flow dominance (lowest epistasis), the number, distance and entropy of Pareto optima is somewhat surprising. We might expect these values to be lower, but because the r value is negative they lie between those of the first and third instances, underlining the importance of the inter-objective correlation in determining the distribution of Pareto optima. Of course, many further observations are needed to verify this effect.

In Figure 1 we compare the shape of the Pareto front in instances 4 and 5, which differ in parameters only in the value of A and B . It is seen that instance 5 (despite having a negative r correlation) has a much more convex-shaped PF.

In Table 3 we present the first of our two publicly available test suites, produced using the uniform instance generator. The first three instances are of size $n = 20$ and have two objectives. The second three instances are of size $n = 30$ and have three objectives. For these instances it is not practical to perform exhaustive search, so we are unable to give measures of the Pareto front. However, we are able to repeatedly (100 000 times) apply a deterministic local search to these instances, and can measure the number of internally nondominated optima found, as well as their entropy and diameter. For more information on our

Param/Property	Parameter/Property Values by Instance		
	KC20-2fl-1uni	KC20-2fl-2uni	KC20-2fl-3uni
$\max(d)$	100	100	100
$\max(f)$	100	100	100
$\text{corr}(f^1, f^2)$	0.0	0.7	-0.7
Spearman r	-0.14	0.79	-0.82
fd^1, fd^2	58.5, 64.6	58.6, 56.9	60.3, 61.8
dd	60.7	61.9	62.2
# nondom	80	19	178
diam(nd)	15	14	16
ent(nd)	0.828	0.43	0.90
seed	89749235	20983533	48927232

Param/Property	Parameter/Property Values by Instance		
	KC30-3fl-1uni	KC30-3fl-2uni	KC30-3fl-3uni
$\max(d)$	100	100	100
$\max(f)$	100	100	100
$\text{corr}(f^1, f^{k>1})$	0.0	0.4	-0.4
Spearman r	-0.02, -0.02	0.43, 0.50	-0.43, -0.42
fd^1, fd^2, fd^3	59.4, 61.3, 64.8	60.3, 57.2, 58.3	61.0, 59.2, 61.8
dd	59.9	58.7	60.1
# nondom	705	168	1257
diam(nd)	24	22	24
ent(nd)	0.97	0.92	0.96
seed	549852743	394908123	121928193

Table 3. Test suite of three 20 node, 2 flow matrix, and three 30 node, 3 flow matrix, uniformly random instances with different flow correlations and other parameters. Parameters and properties are as for previous tables, except that nondom replaces PO here, and refers to an internally nondominated set from a total of 100 000 local optima

procedure for doing this see [11]. We can observe that on these instances, where the fd and dd levels are fairly constant, the number of nondominated optima and their diameter and entropy follow closely the value of r .

Table 4 presents the second of our test suites and gives identical measures for the local search samples. Figure 2 summarises the effects of inter-objective correlations on the number of PO/nondominated solutions found, over otherwise homogeneous sets of instances.

6 Conclusion

We have presented instance generators and test suites for the mQAP, with the aim of providing new benchmarks for EMOO algorithms and to facilitate studies of the relationships between problem characteristics, search landscapes, and algorithm performance in MOCO. In the previous section we have observed some simple relationships between two problem characteristics, namely flow-dominance and inter-objective correlations, and the resulting search landscapes. In [11], we also defined some methods for measuring properties of *local* landscape features. Of particular interest to us, is obtaining some measure of the relative

Param/Prop	Parameter/Property Values by Instance			
	KC20-2fl-1rl	KC20-2fl-2rl	KC20-2fl-3rl	KC20-2fl-4rl
$\max(d)$	196	173	185	164
$\max(f)$	9954	9644	9326	96626
A	-2	-2	-2	-5
B	4	4	4	5
$\text{corr}(f^1, f^2)$	0.0	0.4	-0.4	0.4
Spearman r	0.08	0.43	-0.49	0.32
overlap η	1.0	1.0	1.0	1.0
fd^1, fd^2	206.8, 230.1	243.8, 367.0	246.2, 350.0	305.9, 477.5
dd	54.9	57.9	56.3	56.5
# nondom	541	842	1587	1217
diam(nd)	15	14	15	15
ent(nd)	0.63	0.6	0.66	0.51
seed	1213563	45767	8347234	1121932

Param/Prop	Parameter/Property Values by Instance			
	KC20-2fl-5rl	KC30-3fl-1rl	KC30-3fl-2rl	KC30-3fl-3rl
$\max(d)$	185	172	180	172
$\max(f)$	98776	9929	9968	9852
A	-5	-2	-2	-2
B	5	4	4	4
$\text{corr}(f^1, f^2)$	0.4	0.4	0.7	-0.4
$\text{corr}(f^1, f^3)$	—	0.0	-0.5	-0.4
$\text{Sp}.r(f^1, f^2)$	-0.25	0.10	0.46	-0.75
$\text{Sp}.r(f^1, f^3)$	—	-0.29	-0.67	-0.70
overlap η	0.5	0.7	0.7	0.2
fd^1, fd^2, fd^3	351.5, 391.1, —	235.3, 320.7, 267.5	233.7, 337.5, 341.8	251.7, 359.5, 328.3
dd	57.3	56.1	58.6	56.8
# nondom	966	1329	1924	1906
diam(nd)	15	24	24	24
ent(nd)	0.56	0.83	0.86	0.86
seed	3894673	20983533	34096837	9346873

Parameter/Property Values Common to All Instances

$$N_{\max} = 1, r_{\max} = 100, R_{\max} = 0$$

Table 4. Test suite of five 20 node, 2 flow matrix, and three 30 node, 3 flow matrix, real-like instances with different flow correlations and other parameters. Parameters and properties as in previous tables

difficulty of moving towards the Pareto front versus moving along it. This, we think, should bear strongly on the overall search strategy that is most effective on a given problem. Our future work will focus on this issue.

Acknowledgments

Joshua Knowles gratefully acknowledges the support of a Marie Curie research fellowship of the European Commission, contract number HPMF-CT-2000-00992.

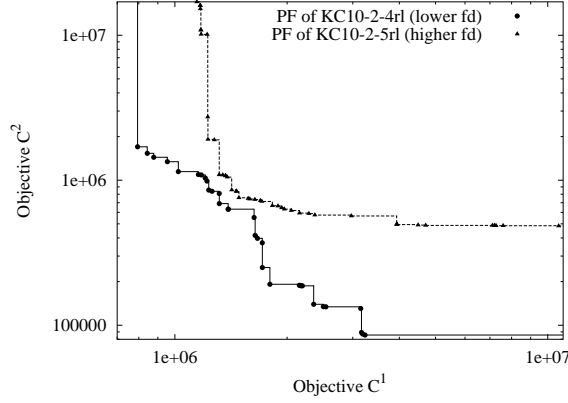


Fig. 1. The effect of different flow dominance (fd) on the Pareto front shape. Notice the logarithmic scale; with high flow dominance the PF is strongly convex

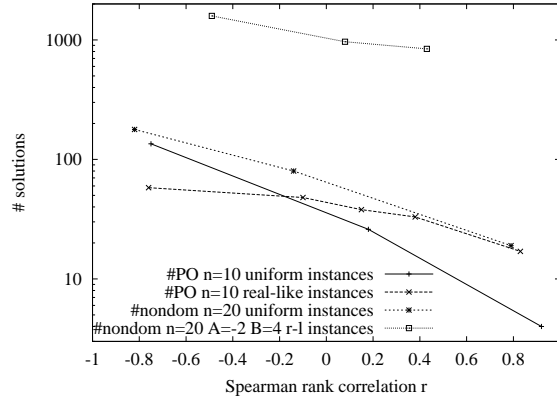


Fig. 2. The effect of inter-objective correlations on the number of nondominated solutions

References

1. V. Bachelet. *Métaheuristiques Parallèles Hybrides: Application au Problème D'affectation Quadratique*. PhD thesis, Université des Sciences et Technologies de Lille, December 1999.
2. P. C. Borges and M. P. Hansen. A basis for future successes in multiobjective combinatorial optimization. Technical Report IMM-REP-1998-8, Institute of Mathematical Modelling, Technical University of Denmark, March 1998.
3. R. E. Burkard, S. Karisch, and F. Rendl. QAPLIB—A quadratic assignment problem library. *European Journal of Operational Research*, pages 115–119, 1991.
4. K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.
5. K. Deb, A. Pratap, and T. Meyarivan. Constrained Test Problems for Multi-objective Evolutionary Optimization. In E. Zitzler et al., editors, *First Interna-*

- tional Conference on Evolutionary Multi-Criterion Optimization*, pages 284–298. Springer-Verlag. LNCS 1993, 2001.
6. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
 7. S. Droste, T. Jansen, K. Tinnefeld, and I. Wegener. A new framework for the valuation of algorithms for black-box optimization. In *FOGA 2002, Proceedings of the Seventh Foundations of Genetic Algorithms Workshop*, pages 197–214, 2002.
 8. M. Ehrgott and X. Gandibleux. A Survey and Annotated Bibliography of Multi-objective Combinatorial Optimization. *OR Spektrum*, 22:425–460, 2000.
 9. C. Fleurent and J.A. Ferland. Genetic hybrids for the quadratic assignment problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16:173–188, 1994.
 10. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
 11. J. Knowles and D. Corne. Towards Landscape Analyses to Inform the Design of Hybrid Local Search for the Multiobjective Quadratic Assignment Problem. In A. Abraham et al., editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279, Amsterdam, 2002. IOS Press.
 12. J. D. Knowles and D. W. Corne. A Comparison of Encodings and Algorithms for Multiobjective Minimum Spanning Tree Problems. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 1, pages 544–551, Piscataway, New Jersey, May 2001. IEEE Service Center.
 13. M. Laumanns and J. Ocenasek. Bayesian Optimization Algorithms for Multi-objective Optimization. In J. J. Merelo Guervós et al., editors, *Parallel Problem Solving from Nature—PPSN VII*, pages 298–307, 2002. Springer-Verlag. LNCS 2439.
 14. P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352, 2000.
 15. B. Naudts and L. Kallel. A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1):1–15, April 2000.
 16. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
 17. S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23:555–565, 1976.
 18. T. Stützle and M. Dorigo. Available ACO algorithms for the QAP. In D. Corne et al., editors, *New Ideas in Optimization*, chapter 3, pages 33–50. McGraw Hill, 2000.
 19. E. D. Taillard. A comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3:87–105, 1995.
 20. T. E. Vollmann and E. S. Buffa. The facilities layout problem in perspective. *Management Science*, 12(10):450–468, 1966.
 21. E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
 22. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.