Multiobjective Quadratic Assignment Problem Solved by an Explicit Building Block Search Algorithm – MOMGA-IIa

Richard O. Day and Gary B. Lamont

Department of Electrical Engineering,
Graduate School of Engineering & Management,
Air Force Institute of Technology,**
WPAFB (Dayton) OH, 45433, USA
{Richard.Day, Gary.Lamont}@afit.edu

Abstract. The multi-objective quadratic assignment problem (mQAP) is an non-deterministic polynomial-time complete (NPC) problem with many real-world applications. The application addressed in this paper is the minimization of communication flows in a heterogenous mix of Organic Air Vehicles (OAV). A multi-objective approach to solving the general mQAP for this OAV application is developed. The combinatoric nature of this problem calls for a stochastic search algorithm; moreover, two linkage learning algorithms, the multi-objective fast messy genetic algorithm (MOMGA-II) and MOMGA-IIa, are compared. Twenty-three different problem instances having three different sizes (10, 20, and 30) plus two and three objectives are solved. Results indicate that the MOMGA-IIa resolves all pareto optimal points for problem instances < 20.

1 Introduction

The scalar quadratic assignment problem (QAP) was introduced in 1957 by Koopmans and Beckmann. In 2002, Knowles and Corne extended the QAP to be multi-objective and it became the multi-objective quadratic assignment problem (mQAP) [11]. Explicit Building Block (BB) search Algorithms are good at solving a multitude of NPC problems [3, 10, 21], including the mQAP. This investigation illustrates our latest achievement in finding a better building block builder by way of a good competitive template selection mechanism added into the multi-objective fast messy GA (MOMGA-II) [22]. The new MOMGA-II is called the MOMGA-IIa. MOMGA-IIa originated as a single objective messy GA

** The views expressed in this article are those of the authors and do not reflect the official policy of the United States Air Force, Department of Defense, or the U.S. Government. The authors also wish to acknowledge the following individuals: Jesse Zydallis for the use of his MOMGA-II code; Mark Kleeman, Todd Hack and Justin Kautz for their persistent help and discussions.

G.R. Raidl and J. Gottlieb (Eds.): EvoCOP 2005, LNCS 3448, pp. 91–100, 2005.

(mGA) and evolved into a multi-objective mGA called the MOMGA [6]. Many different Multi-objective Evolutionary Algorithms (MOEAs) were produced during this time period; however, the MOMGA is the only MOEA explicitly using good BBs to solve problems – even the Bayesian optimization algorithm (BOA) uses a probabilistic model to find good building blocks. The MOMGA has a population size limitation: as the BB size increases so does the population size during the Partially Enumerative Initialization (PEI) phase. This renders the MOMGA less useful on large problems. To overcome this problem, the MOMGA-II, based on the single objective fmGA, was designed. The fmGA is similar to the mGA in that it specifically uses BBs to find solutions; however, it requires smaller population sizes and has a lower run time complexity when compared to the mGA. MOMGA-II includes many different repair, selection, and crowding mechanisms. Unfortunately, the MOMGA-II is found to be limited when solving large problems [5]. This called for the development of basis function diversity measures in the MOMGA-IIa which are designed for smart BB searching in both the geno- and pheno-type domains. The problem under investigation is the mQAP. Test instances used in this study for the mQAP were designed by Knowles and Corne [12]. Results are compared with deterministic results (where available) and previously published attempts at solving these test instances. This paper begins with this introduction and then is followed by a description of the Organic Air Vehicles (OAV) problem mapped to the mQAP problem domain. Next the algorithm domain is presented. A short discussion of the deterministic approach is also included in the algorithm domain discussion. This is followed by the design of experiments section which includes the resources and parameter settings. Finally, results are presented and conclusions are drawn.

2 Problem Domain

Today, OAVs are operated in an independent role where they each have their own mission and a single controller. Future operation of OAVs must include collaboration and autonomous operation of a package (heterogeneous mix) of OAVs. During flight operations of an autonomous package of OAVs, vehicles must communicate in an efficient manner. Flight vehicle patterns play an important role in communication effectiveness (power consumption) during long range missions. In this investigation, the communication and flight pattern of a heterogeneous set of OAVs is mapped to the mQAP.

The QAP was originally designed to model a plant location problem [2]. Mapping the OAV problem into a QAP is accomplished with replacement. By inserting OAVs for plants, flight formation positions for plant locations, and communication traffic for supply flow, the OAVs problem is mapped directly onto the QAP. The mQAP is similar to the scalar QAP[1], with the exception of having multiple types of flows coming from each object.

---

[1] See `http://www.seas.upenn.edu/qaplib/` for more info about the QAP.

For example, the OAVs may use one communication channel for passing reconnaissance information, another channel for target information, and yet another channel for OAV operational messages. The end goal is to minimize all the communication flows between OAVs. The mQAP[2] is defined mathematically in Equations 1 and 2.

$$minimize\{C(\pi)\} = \{C^1(\pi), C^2(\pi), \ldots, C^m(\pi)\} \tag{1}$$

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

> where n is the number of objects/locations, a_{ij} is the distance between location i and location j, b^k_{ij} is the kth flow from object i to object j, π_i gives the location of object i in permutation $\pi \in P(n)$, and 'minimize' means to obtain the Pareto front [12].

Many algorithm approaches have been used on the QAP. QAP researchers can only optimally solve for problems that are of size < 20. Furthermore, problem sizes of 15 are extremely difficult [2]. When feasible, optimal solutions are found using branch and bound methods [8, 2]. However, since many real-world problems are larger than 20 instances, other methods need to be employed in order to find a good solution in a reasonable amount of time. The use of Stocastic Local Searches and Ant Colonies has been explored. These have been found to do well when compared to some of the best heuristics available for the QAP and mQAP [7, 15, 18]. Evolutionary algorithms have also been applied [17, 9]. Additionally, several researchers have compared the performance of different search methods [20, 16].

3 Algorithm Domain

While many different algorithms have been used to solve the QAP [1, 13], only a few have been applied to mQAP [10]. This investigation compares results found by all attempts at solving the mQAP test instances developed in [12]. Table 1 list all multi-objective problem (MOP) instances solved in this investigation. Unfortunately, some researchers do not have access to solutions found by their algorithm, so a direct pareto front comparison cannot be made for a more accurate differentiation between solution quality.

Knowles and Corne [11] collected results by running 1000 local searches from each of 100 (for 2-objective instances) or 105 (3-objective instances) different λ vectors, thus giving them $\approx$ 200000 records. This technique is an interesting one; however, they do not include the actual data points found on their pareto front. Also, we previously used a multi-objective evolutionary algorithm (MOEA), MOMGA-II, to solve the problem [4]. The next section describes both the MOMGA-II and MOMGA-IIa.

---

[2] See `http://dbk.ch.umist.ac.uk/knowles/mQAP/` for more info about the mQAP.

Table 1. Multi-objective problems numbered and listed according to size and number of objectives. There are *real like* (#rl) and *uniform* (#uni) instances. The size of each problem is indicated by the two digit number following the *KC* (KC##). The number of objectives for each problem is indicated by the number preceeding the *fl* (#fl). Each column list the sizes of MOPs used: 10, 20, and 30. The shaded area of the table is identifying the MOPs with 3 objectives - others have 2 objectives

(MOP #)	Name (size 10)	(#)	Name (size 20)	(#)	Name (size 30)
1	KC10-2fl-1rl	9	KC20-2fl-1rl	17	KC30-2fl-1r1
2	KC10-2fl-1uni	10	KC20-2fl-1uni	18	KC30-3fl-1rl
3	KC10-2fl-2rl	11	KC20-2fl-2rl	19	KC30-3fl-1uni
4	KC10-2fl-2uni	12	KC20-2fl-2uni	20	KC30-3fl-2rl
5	KC10-2fl-3rl	13	KC20-2fl-3rl	21	KC30-3fl-2uni
6	KC10-2fl-3uni	14	KC20-2fl-3uni	22	KC30-3fl-3rl
7	KC10-2fl-4rl	15	KC20-2fl-4rl	23	KC30-3fl-3uni
8	KC10-2fl-5rl	16	KC20-2fl-5rl		

3.1 Extended Multi-objective fmGA (MOMGA-IIa)

The MOMGA-IIa is a multi-objective version of the fmGA that has the ability to achieve a semi-partitioned search in both the genotype and phenotype domains during execution. It is an algorithm that exploits “good” building blocks (BBs) in solving optimization problems. These explicit BBs represent “good” information in the form of partial strings that can be combined to obtain even better solutions. The MOMGA-IIa algorithm executes in three phases: Initialization, Building Block Filtering, and Juxtapositional Phase. See Figure 1 for diagram of the program flow.

The algorithm begins with the Probabilistically Complete Initialization (PCI) Phase where it randomly generates a user specified number of population members. These population members are a specified chromosome length and each is evaluated to determine its respective fitness values. Our implementation utilizes a binary scheme in which each bit is represented with either a 0 or 1.

The Building Block Filtering (BBF) Phase follows by randomly deleting loci and their corresponding allele values in each of the population member’s chromosomes. This process completes once the length of the population member’s chromosomes have been reduced to a predetermined BB size. These reduced chromosomes are referred to as underspecified[3] population members. In order to evaluate population members that have become underspecified, competitive templates (CTs) are utilized to fill in the missing allele values. Evaluation consists of the partial string being overlayed onto a CT just prior to evaluation. CTs are fully specified chromosomes that evolve as the algorithm executes. CT replacement is done after each BB generation. In the MOMGA-II, future CTs are updated with the best individuals found with respect to each objective function. However, the MOMGA-IIa selects a *competent* CTs that partitions both the phenotype and genotype. This innovative balance is achieved through two mechanisms: Orthogonal CT generation and Target Vector (TV) guidance. Orthogonal CT generation is used to partition the genotype space, while keeping a

---

[3] An underspecified chromosome is chromosome where some, but not all locus positions have an associated allele value.

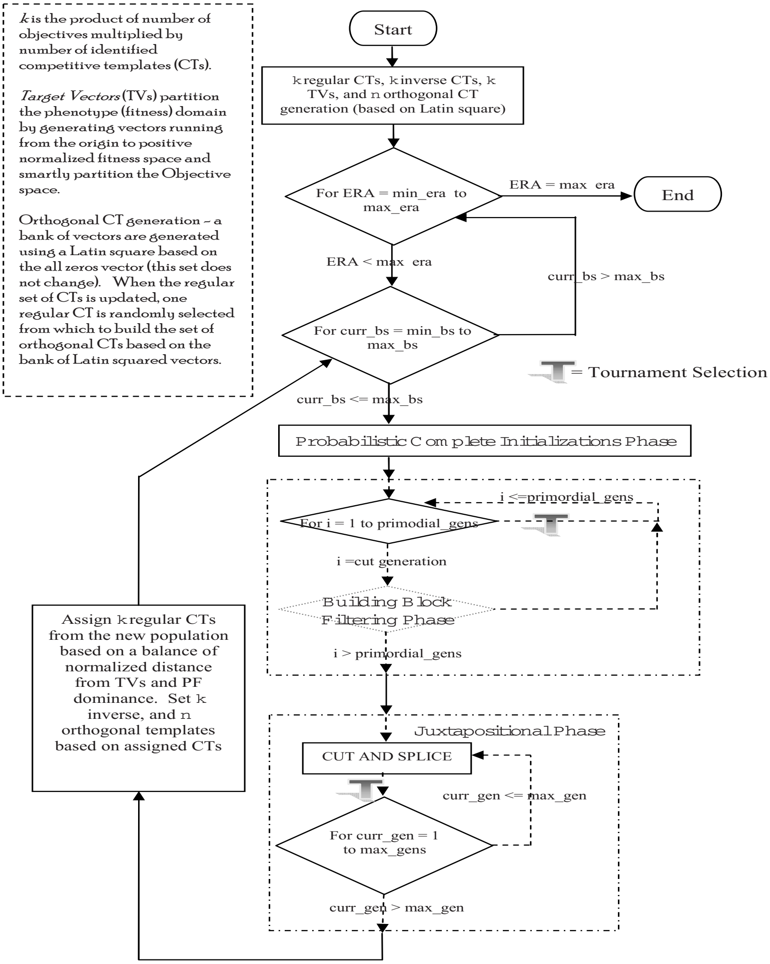

Fig. 1. This figure illustrates the MOMGA-IIa program flow. Note the placement of each phase and where tournament selection is performed. Additionally, the MOMGA-IIa exploits and partitions in both the phenotype and genotype domains by updating and generating regular, inverse, and orthogonal competitive templates. See Section 3.1 for a detailed description of the algorithm

good partition in the phenotype space requires TV guidance. TVs are normalized fitness markers that capture one solution per vector for future CT replacement. In the MOMGA-IIa, target vectors are used in a manner to divide the normalized fitness space of pareto-front members and select a distribution of CTs that fall nearest to each TV. Also, an orthogonal bank of chromosomes is used to filter a randomly selected CT through for creation of a set of orthogonal CTs.

The BBF process is alternated with a selection mechanism to keep only the strings with the "best" BBs found, or those with the best number of fitness values. In the case of a tie, where two strings each have an equal number of better fitness values (i.e. each have $\frac{m}{2}$ "best" fitness values), the string is randomly selected between the two. It should be noted that the MOMGA-IIa has a more complex selection mechanism than MOMGA-II because it maintains more fitness values per solution. In the MOMGA-II each string has m fitness values, while in MOMGA-IIa each string has $f = (c * m + i + o) * m$ fitness values associated with it – corresponding to the m objective functions to optimize, c competitive templates, i inverse templates (equal to $c * m$), and o orthogonal templates.

Finally, the juxtapositional phase uses the BBs found through the BBF phase and recombination operators to create chromosomes that are fully specified. A chromosome is referred to as fully specified if it is not missing any locus positions, or in other words, does not need to use the CT for evaluation.

The MOMGA-IIa has an outer and inner loop that must be completely iterated through using each BB size and epoch before terminating.

3.2 Non-stochastic Approaches

Two different approaches are discussed in this section. The first is the type of approach used by Knowles and Corne in [12]. The second approach is simply our exhaustive search algorithm.

Local Search Approach: The local search (LS) method employed for the mQAP problem is where positions of facilities (or objects) are switched [11]. The new positioning is kept if the new configuration yields a lower fitness value. This search method works for solving the QAP [19]; however, the mQAP makes employing a strict LS approach difficult for the deceptive hyperplanes that accompany multi-objective problems. Specifically, a researcher is faced with how to initialize the LS method. Knowles and Corne concluded that the starting points would be randomly selected out of a basin of attraction [12, 19]. After a new point is picked, the LS method is applied for a specified number of generations. This is the technique used by [12] finding most, if not all, pareto front (PF) solutions. To our knowledge, the solutions for the larger problems have not been published, making comparisons difficult.

Complete Iterative Approach: The complete iterative approach is an exhaustively deterministic approach that can be accomplished on MOPs 1∼8 (See Table 1). The number of solutions that must be evaluated is calculated by Equation 3 where n is the number of facilities and k is the number of locations. Consequently, for the mQAP, $k = n$.

$$x \approx \frac{n!}{(n-k)!}, \; \textit{where } k = n \textit{ this reduces to } \mathbf{n!} \tag{3}$$

Function calculations for each MOP are $m * 10!$, $m * 20!$, and $m * 30!$ or $m * 3628800$, $\approx m * 2.43e18$ and $\approx m * 2.65e32$. These numbers are not to be confused with the search space size. For each search space solution, m calculations must occur.

Table 2. System Configuration

Cluster 1 (TAHOE)	Cluster 2 (ASPEN)	Cluster 3 (Polywells)
Fedora Core 2/Raid 5	Redhat Linux 9.0/Raid 5	Redhat Linux 7.3/Raid 5
Dual Opteron 2.2 ghz	Ath XP 3000+ 2.1ghz	Ath XP 2800+ 2.0ghz
RAM 4 GByte/Cache(L1 I 64,D 64/L2 1024)KB	1 GByte/(64,64/512)KB	1 GByte/(64,64/512)KB
Crossbar Switch/Gb Ethernet	Crossbar Switch/Fast Ethernet	Crossbar Switch/Gb Ethernet
65 node,2 CPUS/node	48 node,2 CPUS/node	16 node,1 CPU/node

Table 3. Summary of Results for all experiments. Included in this table are the number of optimal pareto front points (when known), and the number of PF points found by each algorithm {MOMGA-IIa (M-IIa), MOMGA-II (M-II), and LS}. *u* indicates that it is unknown how many dominated solutions this particular algorithm found when compared to the best PF solutions set found by all the algorithms considered. In addition, diameter (dia) and entropy (ent) is calculated for M-II's and M-IIa's solutions

	mQAP Number, Size 10, (Deterministic PF True Points)							
Algorithm	True PF Pts Found/Total PF pts Found							
	1(58)	**2**(13)	**3**(15)	**4**(1)	**5**(55)	**6**(130)	**7**(53)	**8**(49)
LS	58/58	13/13	15/15	1/1	55/55	130/130	53/53	49/49
M-II	57/59	13/13	11/17	0/3	50/53	122/122	25/34	36/45
M-IIa†	58/58	11/12	15/15	1/1	55/55	130/130	53/53	49/49
†Time(mins)	21.5	62.3	29.8	10.9	45.5	68.1	45.4	15.8
	mQAP Number, Size 20							
	9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
LS	u/541	u/80	u/842	u/19	u/1587	u/178	u/1217	u/966
M-II	0/17	0/24	0/12	0/5	0/29	0/51	0/28	0/17
(dia/ent)	11.6/0.43	11.4/0.48	11.01/0.45	7.2/0.25	12.1/0.54	12.3/0.55	10.39/0.43	11.76/0.46
M-IIa†	36/36	33/33	31/31	7/7	63/63	139/139	48/48	44/44
(dia/ent)	13.0/0.58	13.7/0.69	11.17/0.47	3.67/0.16	14.1/0.73	15.5/0.88	12.76/0.60	11.37/0.50
†Time(days)	9.8	8.3	8.3	8.3	8.8	8.3	8.3	1.7
	mQAP Number, Size 30							
	17	**18**	**19**	**20**	**21**	**22**	**23**	
LS	-	u/1329	u/705	u/1924	u/168	u/1909	u/1257	
M-II	n/a	0/507	0/552	10/552	0/104	0/795	0/755	
(dia/ent)	-	24.1/0.79	20.1/0.50	24.3/0.78	22.3/0.64	21.2/0.57	20.4/0.56	
M-IIa†	40/40	507/507	552/552	542/552	104/104	795/795	755/755	
(dia/ent)	17.2/0.42	23.9/0.80	23.2/0.76	23.1/0.74	21.9/0.59	24.0/8.11	25.1/0.90	
†Time(days)	8	∼8	8	8	8	8	8	

4 Design of Experiments

Experiments for the MOMGA-II were conducted on Clusters 2 and 3 listed in Table 2. Experiments for the MOMGA-IIa were done on Cluster 1 in the same table. The MOMGA-II was run 10 times in parallel and the data was then processed incrementally so as to show solutions gradually being found. The MOMGA-IIa ran 10 experiments in serial and kept one pool of PF solutions at all times. The MOMGA-II was run using BB sizes 1 through (10, 10, and 10) while the MOMGA-IIa was run using BB sizes 1 through (10, 15, and 20) for each MOP sized (10, 20, and 30). These experiments are run to determine how well each algorithm can solve the MOPs in this study. This research group's hypothesis is that the MOMGA-II's CT generation and evolution mechanism limited the exploration and building block finding ability of the algorithm, while the MOMGA-IIa now has the enhancement required to overcome this limitation.

5 Results and Analysis

Overall we are pleased with results of the MOMGA-IIa. In the MOP of size less than 20, the MOMGA-IIa found all true PF points available in a short amount of time (under 16 minutes in some cases) - the MOMGA-II did not. Additionally, in MOPs of size 20, the MOMGA-IIa solutions dominated the MOMGA-II's in every case (Illustrated by Figure 2). Finally, in MOPs of size 30 the MOMGA-IIa found more solutions than the MOMGA-II and were found to be dominate in all except for the MOP 20 case where 10 solutions found by the MOMGA-II are non-dominated. The reader should note also that MOPs of size 20 and 30 all took several days to solve. As far as the results for the LS method, these results are good in quantity, but without the actual data to compare we cannot claim that either algorithm LS or MOMGA-IIa is better at solving these MOPs. PF points found for each MOP will soon be posted on our web site,

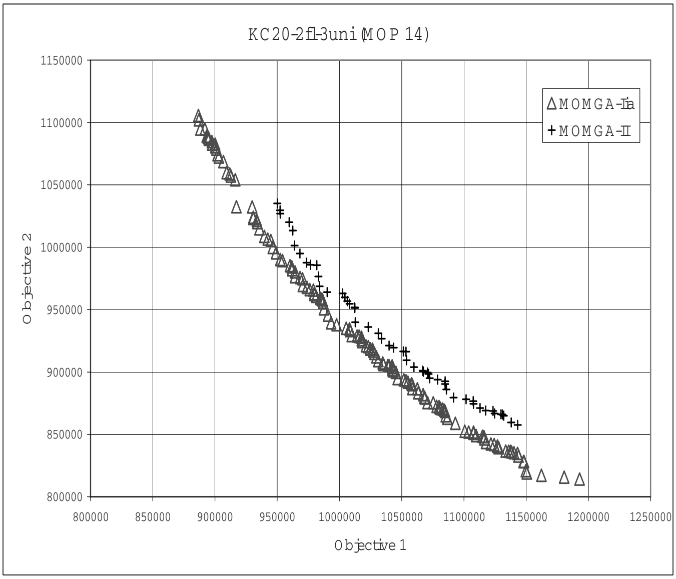

Fig. 2. Results for MOP 14 illustrating that the MOMGA-IIa's CT generator is superior at fining good BBs

`http://en.afit.edu/agct`. We conclude that the reason for the dominance of the MOMGA-IIa over the MOMGA-II is due to the CT generation and selection mechanism. In addition, MOMGA-II's limited number of CTs might be causing it to destroy some good building blocks. Lastly, the CT selection mechanism for the MOMGA-IIa allows for better multi-objective building blocks to be found - thus the MOMGA-IIa is a better building block builder. This phenomena is reflected in the data for each MOP.

Future Analysis: Further analysis of the MOMGA-IIa in solving the mQAP is required including a comparison to recent algorithm designs to solve the Biobjective QAP using Ant Colony Optimization (ACO) by Luis Paquete [14]. In addition, a memetic adjustment to MOMGA-IIa by adding a local search onto the end of the Juxtapositional Phase should be evaluated.

References

1. R.E. Burkard, S.E. Karisch, and F. Rendl. A quadratic assignment problem library. *Journal of Global Optimization*, pages 391–403, 1997.
2. Eranda Çela. *The Quadratic Assignment Problem - Theory and Algorithms*. Kluwer Academic Publishers, Boston, MA, 1998.
3. Richard O. Day. A multiobjective approach applied to the protein structure prediction problem. Ms thesis, Air Force Institute of Technology, March 2002. Sponsor: AFRL/Material Directorate.
4. Richard O. Day, Mark P. Kleeman, and Gary B. Lamont. Solving the Multiobjective Quadratic Assignment Problem Using a fast messy Genetic Algorithm. In *Congress on Evolutionary Computation (CEC'2003)*, volume 1, pages 2277–2283, Piscataway, New Jersey, December 2003. IEEE Service Center.
5. Richard O. Day and Gary B. Lamont. Multi-objective fast messy genetic algorithm solving deception problems. *Congress on Evolutionary Computation; Portland, Oregon*, 4:1502–1509, June 19 - 23 2004.
6. Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
7. L. M. Gambardella, E. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problems. *Journal of the Operational Research Society*, 50:167–176, 1999.
8. Peter Hahn, Nat Hall, and Thomas Grant. A branch-and bound algorithm for the quadratic assignment problem based on the hungarian method. *European Journal of Operational Research*, August 1998.
9. Jorng-Tzong Horng, Chien-Chin Chen, Baw-Jhiune Liu, and Cheng-Yen Kao. Resolution of quadratic assignment problems using an evolutionary algorithm. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 2, pages 902–909. IEEE, IEEE, 2000.
10. Mark P. Kleeman. Optimization of heterogeneous uav communications using the multiobjective quadratic assignment problem. Ms thesis, Air Force Institute of Technology, March 2004. Sponsor AFRL.

11. Joshua Knowles and David Corne. Towards Landscape Analyses to Inform the Design of Hybrid Local Search for the Multiobjective Quadratic Assignment Problem. In A. Abraham, J. Ruiz del Solar, and M. Koppen, editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279, Amsterdam, 2002. IOS Press. ISBN 1-58603-297-6.
12. Joshua Knowles and David Corne. Instance generators and test suites for the multiobjective quadratic assignment problem. In Carlos Fonseca, Peter Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003, Proceedings*, number 2632 in LNCS, pages 295–310. Springer, 2003.
13. Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. An analytical survey for the quadratic assignment problem. Technical report, Council for the Scientific and Technological Development, of the Brazilian gov, 2004.
14. M. López-Ibáñez, L. Paquete, and T. Stützle. On the design of ACO for the biobjective quadratic assignment problem. In M. Dorigo, M. Birattari, C. Blum, L. Gambardella, F. Montada, and T. Stützle, editors, *Proceedings of the Fourth International Workshop on Ant Colony Optimization (ANTS 2004)*, volume 3172 of *Lecture Notes in Computer Sience*. Springer Verlag, 2004. (©Springer Verlag).
15. Vittorio Maniezzo and Alberto Colorni. The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering*, 11:769–778, 1999.
16. Peter Merz and Bernd Freisleben. A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99*, volume 3, pages 1999–2070. IEEE, IEEE, 1999.
17. Peter Merz and Bernd Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4:337–352, 2000.
18. L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, 2004. (©Springer Verlag).
19. Thomas Stntzle. Iterated local search for the quadratic assignment problem. *Technical Report AIDA-99-03*, 1999.
20. Eric D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location science*, 3:87–105, 1995.
21. Jesse B. Zydallis. *Explicit Building-Block Multiobjective Genetic Algorithms: Theory, Analysis, and Development*. Dissertation, Air Force Institute of Technology, AFIT/ENG, BLDG 642, 2950 HOBSON WAY, WPAFB (Dayton) OH 45433-7765, Feb 2002.
22. Jesse B. Zydallis, David A. Van Veldhuizen, and Gary B. Lamont. A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA–II. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 226–240. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.