# An Empirical Comparison of Memetic Algorithm Strategies on the Multiobjective Quadratic Assignment Problem

Deon Garrett
Department of Computer Science
University of Memphis
Memphis, TN, 38122
Email: deong@acm.org

***Abstract*—Evolutionary algorithm based metaheuristics have gained prominence in recent years for solving multiobjective optimization problems. These algorithms have a number of attractive features, but the primary motivation for many in the community is rooted in the use of a population inherent to evolutionary algorithms, which allows a single optimization run to provide a diverse set of nondominated solutions. However, for many combinatorial problems, evolutionary algorithms on their own do not perform satisfactorily. For these problems, the addition of a local search heuristic can dramatically improve the performance of the algorithms. Often called memetic algorithms, these techniques introduce a number of additional parameters which can require careful tuning. In this work, we provide an empirical comparison of a number of strategies for the construction of multiobjective memetic algorithms for the multiobjective quadratic assignment problem (mQAP), and provide a more principled analysis of those results using insights gained from analysis of the fitness landscape properties of the different problem instances.**

## I. Introduction

In recent years, a number of metaheuristic techniques have been developed for the purpose of efficiently solving multiobjective optimization problems. Among these techniques, evolutionary algorithms have become particularly popular, due to their promise of using a population to move toward finding an entire set of good solutions in a single run. However, on many combinatorial problems, evolutionary algorithms alone are not able to successfully find good solutions. By including elements of other heuristics such as local search algorithms, as well as more sophisticated metaheuristics, into the search process, researchers have found that these hybrid or memetic algorithms can often provide very good performance on many problems.

Unfortunately, this gain is not without cost. Most forms of memetic algorithms introduce several new parameters which must be correctly determined by the user, and the overall performance of these algorithms can be very sensitive to these parameters. Thus, an ongoing focus of researchers is to determine ways in which to avoid the need for this type of parameter tuning. Of particular value is a fundamental set of tools with which one may obtain information concerning the structural properties of a given search space, along with a theory of search algorithm behaviour rooted in the relationship between algorithm performance and these search space properties.

It should also be pointed out that concurrent research on automatic tuning algorithms has produced some successes, such as F-race [3], [2], [1]. However, while tools such as F-race provide a very valuable service to the practitioner seeking to maximize performance on a given problem, they are typically very computationally demanding. In addition, one is generally somewhat limited in the space of possible parameter settings that can be efficiently searched, leading one to perhaps prematurely finalize on some details.

An alternative approach to this type of parameter tuning is to try to use insights gained from careful analysis of the problem at hand. In this type of approach, one need not prune the decision space down to a degree than can be efficiently searched. Instead, the entire space of optimization algorithms may be available, and those with the features seemingly best suited to the given problem may be selected.

Unfortunately, the type of analysis necessary to make informed decisions is still largely unknown, particularly in the case of multiobjective optimization. As a result, researchers sometimes turn to favorite metaheuristics when faced with an unknown problem, with little reason to believe that their choice is particularly good. Another contributing factor is that mulitobjective optimization through metaheuristics is still a relatively new phenomenon. Although there are now multiobjective equivalents to most conventional metaheuristics, such as ant colony systems [], tabu search [10], [11], simulated annealing [], differential evolution [], and many others, the field is still somewhat open to novel algorithm variants. With maturity, this pace tends to slow allowing more theoretical developments to catch up with the practical application of the techniques.

Regardless, some researchers have made slow progress in this direction in recent years. In [14], some preliminarily analysis of the properties of the newly proposed multiobjective quadratic assignment problem (mQAP) was performed, and used to suggest potentially fruitful approaches to solving that problem. Their work was extended in [8] to consider the role of evolutionary algorithms in the navigation of the multiobjective search space. That work also replicated many of the results of [16], while serving to provide a more fundamental understanding of those results. It should also be noted that careful tuning of low-level parameters as performed by automated tools such as F-race is generally still necessary. However, by considering the properties of a given search space, the hope is that one can limit the search to the process of fine tuning a small set of parameters.

In this work, we examine several different approaches to embedding local search algorithms into a popular multiobjective memetic algorithm. In addition, we show how the properties of the resulting search spaces can help to explain the differences in performance of the various algorithms.

## II. Previous Work

Multiobjective evolutionary algorithms have risen to prominence in recent years, and in the last decade or so, have begun to mature into well-known and widely used techniques. However, within the realm of combinatorial optimization problems, these algorithms often fail to find competitive solutions. By augmenting the evolutionary search with one or more local search operators (or by including an evolutionary aspect in an existing local search framework, depending on your perspective), high performance algorithms can often be created, even on these difficult combinatorial optimization problems.

Lopez et. al. [16] produced an extensive comparison of several population-based hybrid algorithms on the biobjective QAP. In that work, it was shown that an evolutionary algorithm (SPEA2 [21]) hybridized with a variant of Tabu Search [10], [11] produced the highest performance of the algorithms tested. This combination of evolutionary algorithms with local search is sometimes termed a memetic algorithm, and these algorithms often provide some of the best performing algorithms on several distinct optimization tasks. Local search algorithms alone were considered in [18].

To this point, relatively little has been done concerning multiobjective fitness landscape analysis. In their initial work on the mQAP [14], [12], Knowles and Corne provided some preliminary discussion of the type of fitness landscape which might be presented by their newly proposed problem and what impact that might have on algorithm performance. Extending their work, Garrett [8] developed a framework for analyzing the effects of recombination in evolutionary algorithms on the mQAP, and used it to explain the results of the performance comparison detailed by Lopez. Similar techniques were applied to a multiobjective formulation of the generalized assignment problem (mGAP) in [9], where they were used to examine the relative performance of different local search strategies.

One of the more popular themes in the recent literature from the multiobjective evolutionary algorithms community concerns what is sometimes called “Many objective optimization,” where “many” is generally taken to mean greater than or equal to about four [6]. As noted by Corne [4], there are several reasons why evolutionary algorithms begin to fail in high-dimensional objective spaces. The primary reason lies with their reliance on Pareto dominance to provide differential fitness values to guide selection. A simple consequence of the curse of dimensionality is that the likelihood of any element of a random sample dominating another point decreases exponentially with the size of the objective space. Thus, evolutionary algorithms are quite quickly reduced to random sampling when the number of objectives reaches even quite conservative values.

However, there are other reasons why these algorithms can struggle as the number of objectives is increased. Some algorithms recursively subdivide the objective space into hypercubes as a mechanism for promoting diversity. Again, the number of hypercubes grows exponentially with the dimensionality, and thus the algorithms can become unusably slow or memory hungry.

There is currently quite a bit of work being done to address these issues [13], [4]. However, given that evolutionary algorithms alone are very often insufficient to yield high performance on difficult combinatorial optimization problems, it seems reasonable to attempt to attack these problems, at least partially, through means that do not suffer from the same stiff penalties handed out to evolutionary algorithms by high-dimensional objective spaces.

One simple possibility is to employ scalarization-driven local search to both improve the quality of solutions found by a evolutionary based algorithm, but also to escape the jail introduced by the lack of dominance relations in high dimensions. Local search algorithms have their own difficulties with many objectives, principally the number of independent local searches required to construct a dense sample of a high-dimensional Pareto front. However, these difficulties are of a different character than those faced by MOEA based algorithms, and so the combination of the two could be a fruitful approach for at least some types of multiobjective combinatorial optimization problems.

## III. Memetic Algorithms for Multiobjective Optimization

Memetic algorithms are formed by adding one or more local search based metaheuristics to an existing evolutionary algorithm. In the context of multiobjective optimization, this introduces a number of questions regarding how best to take advantage of both aspects of the resulting algorithm. With regard to local search, there is a particular issue related to the lack of a clear direction of improvement, but there are also issues of which individuals will undergo local improvement and to what degree.

The issue of directionality is often solved via the introduction of a weighting vector to impose a clear direction of improvement externally. For objective spaces of small dimensionality, these weight vectors can be generated systematically to ensure that the entire Pareto front may be targeted by the local search, yielding a well-spread variety of nondominated solutions. In the studies of population based algorithms for mQAP cited above, the best performing algorithms did exactly this.

However, as the number of objectives is increased, two detrimental effects can arise, hindering the performance of such algorithms. The first is that most modern MOEAs rely on Pareto dominance in some form to apply selection pressure. As the number of objectives increases, the liklihood that there will be dominance relationships between the individual points in any sample are lessened. As a result, the MOEA is less able to apply differential selection pressures, and the evolutionary aspects of the search devolve into random mutation hill climbers.

Local search operators guided by a scalarized fitness value are not affected by this phenomenon, as they do not rely on dominance for their ability to make improving moves. However, as the number of objectives gets larger, the number of weight vectors necessary for any particular sample density along the Pareto front increases exponentially. Because this type of algorithm builds the Pareto front one point at a time, with each generated by a separate local search, the run time required to find a good Pareto front approximation also increases exponentially.

## IV. The Multiobjective QAP

The quadratic assignment problem (QAP) is one of the oldest and most widely studied combinatorial optimization problems. First formulated by Koopmans and Beckmann in 1957 [15], the QAP can be described as follows: Given two $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, find a permutation $\vec{\pi}$ that minimizes

$$\min_{\pi} \mathcal{F}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} B_{\pi_i, \pi_j}. \quad (1)$$

Conventionally, the matrices $\mathbf{A}$ and $\mathbf{B}$ are called the *distance* and *flow* matrices, the terminology arising from the original formulation of QAP as a facilities layout problem. Despite the terminology, QAP is useful in model several disparate application areas, including backboard wiring, hospital layout, and keyboard design. Not only $\mathcal{NP}$-hard [19], QAP is generally considered to be among the hardest optimization problems, with even relatively small instances posing a significant challenge to state-of-the-art branch and bound solvers. As shown by Sahni [19], there also exists no polynomial algorithm with a guaranteed error lower than some constant for every instance of QAP unless $\mathcal{P}$=$\mathcal{NP}$. As a result, stochastic local search (SLS) algorithms are the methods of choice for solving most large scale QAP instances.

In a pair of papers, Knowles and Corne proposed and provided detailed static analysis of the multiobjective QAP (mQAP) [14], [12]. The mQAP models any sort of facilities layout problem in which the minimization of multiple simultaneous flows is required. The mQAP consists of a single $n \times n$ distance matrix, and $k$ distinct $n \times n$ flow matrices. There exist then $k$ different pairings of the distance matrix with one flow matrix, yielding $k$ distinct single objective QAP problems. The objective function value of a permutation $\vec{\pi}$ is thus a $k$-dimensional vector with

$$\mathcal{F}^m(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} B^m_{\pi_i, \pi_j} \quad \forall m : 1 \leq m \leq k. \quad (2)$$

One of the key properties described in this initial work on the mQAP was the correlation between flow matrices, and thus between objective function values. The generators provided allow for the creation of either uniform or real-like instances, with specified correlations between the flow matrices. In their initial studies, the correlation played a strong role in determining which types of algorithms might perform best on a given instance. This has since been confirmed by several studies [8], [16], [18].

The mQAP has been studied by a number of researchers since its introduction. However, little has been done to examine the properties of the landscape and their effect on algorithm performance. Knowles and Corne provide a wealth of knowledge concerning a set of benchmark instances. However, their approach considers only static properties of the landscape, omitting the effects caused by the detailed behavior of the search algorithm.

In addition, most experimental studies of the mQAP have focused on the biobjective version. Since the performance of many multiobjective optimization algorithms can be strongly dependent on the number of objectives, one must be careful when generalizing results from studies that contain only a single variety of problem. Note that this is not a criticm of that type of work, however. The sheer scope of the problem of understanding multiobjective search algorithm behaviour is daunting. One of the best ways to make progress is by studying small subsets of the full problem and deriving insights from that.

As such, the focus of this work is to consider the performance of a certain class of memetic algorithms on mQAP instances with three and four objectives, in addition to the more widely studied biobjective variants. The reasons for this are that the mQAP has received enough attention that there is some understanding of the problem structure. This allows us to focus only on the differences between the algorithms we consider in the context of a search space that is at least somewhat understood.

## V. Experiments and Results

The goal of the experiments described in this work is to examine the performance of different types of memetic algorithms on different variants of the multiobjective quadratic assignment problem. In addition, the structural properties of the multiobjective search spaces imposed by the different problem instances should provide useful information concerning which algorithms would be more likely to perform better. To facilitate these goals, a set of random 60 facility mQAP instances was generated. Table I shows the parameters used for each problem instance.

This set of instances covers the more widely studied biobjective problem, as well as problems with three and four objectives. In addition, instances range from very strongly

TABLE I
MULTIOBJECTIVE QAP PROBLEM INSTANCES CONSIDERED IN THIS STUDY.

| Instance | Size | Type | Obj | Corr |
|---|---|---|---|---|
| Gar60-2fl-1uni | 60 | Uniform | 2 | -0.3 |
| Gar60-2fl-2uni | 60 | Uniform | 2 | 0.0 |
| Gar60-2fl-3uni | 60 | Uniform | 2 | 0.3 |
| Gar60-2fl-4uni | 60 | Uniform | 2 | -0.8 |
| Gar60-2fl-5uni | 60 | Uniform | 2 | 0.8 |
| Gar60-2fl-1rl | 60 | Real-Like | 2 | -0.3 |
| Gar60-2fl-2rl | 60 | Real-Like | 2 | 0.0 |
| Gar60-2fl-3rl | 60 | Real-Like | 2 | 0.3 |
| Gar60-2fl-4rl | 60 | Real-Like | 2 | -0.8 |
| Gar60-2fl-5rl | 60 | Real-Like | 2 | 0.8 |
| Gar60-3fl-1uni | 60 | Uniform | 3 | 0.0 |
| Gar60-3fl-2uni | 60 | Uniform | 3 | -0.5 |
| Gar60-3fl-3uni | 60 | Uniform | 3 | 0.5 |
| Gar60-3fl-1rl | 60 | Real-Like | 3 | 0.0 |
| Gar60-3fl-2rl | 60 | Real-Like | 3 | -0.5 |
| Gar60-3fl-3rl | 60 | Real-Like | 3 | 0.5 |
| Gar60-4fl-1uni | 60 | Uniform | 4 | 0.0 |
| Gar60-4fl-2uni | 60 | Uniform | 4 | -0.5 |
| Gar60-4fl-3uni | 60 | Uniform | 4 | 0.5 |
| Gar60-4fl-1rl | 60 | Real-Like | 4 | 0.0 |
| Gar60-4fl-2rl | 60 | Real-Like | 4 | -0.5 |
| Gar60-4fl-3rl | 60 | Real-Like | 4 | 0.5 |

negatively correlated flow matrices to very strongly positively correlated flows. As Knowles and Corne found that this correlation had a strong impact on search space structure and algorithm performance, this variety can help to ensure that the algorithms are not being selected because of good performance on a small set of very similar benchmark problems.

One of the seemingly most critical decisions to make when designing a memetic algorithm lies in how much of the search effort should be dedicated to the local improvement operator versus the more global search provided by the evolutionary operators. In order to focus on this question, we chose to implement four different strategies. All are based on the same underlying search operators, NSGA-II [5] hybridized with Robust Tabu Search [20]. In addition, each algorithm was allowed to sample the search space for 100,000,000 evaluations of the objective function.

The four strategies correspond roughly to "short bursts of local search on all individuals," "longer local search runs on all individuals," "short bursts of local search on randomly chosen individuals," and "longer local search runs on randomly chosen individuals." It has been shown in earlier work [8], [16] that very long runs of local search can perform quite well on biobjective problems, but this typically greatly limits the contribution of the evolutionary operators, and often requires somewhat lengthy computation times. Because in this study, we chose to hold the evaluation limits below the limits in the earlier work in order to focus on the results attainable in reasonably short run times, the very long runs used previously were not permitted. Therefore the results presented here may be viewed as considering the attainable performance through mulitobjective memetic algorithms in more limited run times. For reference, the algorithms considered here tended to take around 20 minutes of CPU time for a single run on a single core of an Intel Quad-Core 2.4 GHz processor.

TABLE II
PARAMETERS FOR EACH VARIANT OF MEMETIC ALGORITHM. NOTE THAT THE VALUE IN PARENTHESES FOLLOWING THE RANDOM STRATEGY DENOTES THE PROBABILITY OF APPLYING THE LOCAL SEARCH TO ANY GIVEN INDIVIDUAL. LIMIT REFERS TO THE NUMBER OF EVALUATIONS EACH RUN OF THE LOCAL SEARCH METHOD WAS ALLOTTED.

| Algorithm | Evaluations | Strategy | Limit |
|---|---|---|---|
| nsga2_ts1 | 100,000,000 | ALL | 5,000 |
| nsga2_ts2 | 100,000,000 | ALL | 50,000 |
| nsga2_ts3 | 100,000,000 | RANDOM(0.1) | 5,000 |
| nsga2_ts4 | 100,000,000 | RANDOM(0.01) | 50,000 |

It is important to note that QAP, and by extension, mQAP supports the incremental calculation of fitness values in the case of a single change from a solution with known fitness. In general, this delta calculation can be done in constant time. A linear time procedure must also be invoked when a move is accepted to compute the delta values for the next set of possible moves. On the other hand, because a GA can make arbitrarily large jumps in the space due to a single application of a genetic operator, the full quadratic time fitness evaluation procedure must generally be invoked by the evolutionary algorithm for each evaluation. This, combined with the nondominated sorting and crowding computations performed each generation by NSGA-II, means that algorithms that are heavier in terms of the proportion of their run times consumed by the tabu search tend to run faster in wall clock time.

For this comparison, the differences were not generally enough to significantly color the results, because there was enough local search in all cases to exhaust a reasonable fraction of the total allotted evaluations in reasonably short order. However, if one were to compare to a pure multiobjective evolutionary algorithm, one would almost certainly see that the MOEA would take far longer than a simpler local search based algorithm to perform the same amount of search. Restated, a pure local search algorithm would be able to search a larger percentage of the space than an evolutionary algorithm, if given the same computational resources.

Table II shows the local search parameters used for each algorithm. In all cases, the cycle crossover (CX) operator was employed with probability 0.95, and swap mutation with probability 0.01. Each evolutionary algorithm used a population size of 100. The "short" local search runs were allowed to consume only 5,000 evaluations, while the longer runs were allotted ten times that, for a total of 50,000 evaluations.

The performance of the algorithms is shown in Tables III, IV, and V in the form of mean hypervolume over ten trials. We used the dimension-sweep algorithm of Fonseca et. al. [7], [17] for the calculation of hypervolume in this work. Table III shows the results of the four strategies on the biobjective cases. In eight of the ten problem instances, the `nsga2_ts4` strategy (longer tabu search on each individual with probability 0.01) outperformed all other strategies. In the remaining two cases, longer tabu search still outperformed shorter runs, but applying the local search to all individuals proved to be better than applying it to only a few of the individuals.

TABLE III
MEAN HYPERVOLUME FOR EACH ALGORITHM ON ALL BIOBJECTIVE PROBLEM INSTANCES.

| Instance | nsga2_ts1 | nsga2_ts2 | nsga2_ts3 | nsga2_ts4 |
|---|---|---|---|---|
| Gar60-2fl-1uni | 1.2146e12 | 1.6437e12 | 1.5455e12 | 1.9511e12 |
| Gar60-2fl-2uni | 7.5126e11 | 1.0592e12 | 7.8916e11 | 1.0878e12 |
| Gar60-2fl-3uni | 2.8964e11 | 3.7761e11 | 3.4688e11 | 4.6977e11 |
| Gar60-2fl-4uni | 2.4735e12 | 2.9529e12 | 2.9353e12 | 3.1231e12 |
| Gar60-2fl-5uni | 1.9033e09 | 1.8565e09 | 1.4299e09 | 6.1882e10 |
| Gar60-2fl-1rl | 2.2740e17 | 2.4837e17 | 2.5259e17 | 2.8041e17 |
| Gar60-2fl-2rl | 1.5118e17 | 1.8552e17 | 1.5394e17 | 2.0741e17 |
| Gar60-2fl-3rl | 1.6061e17 | 1.9443e17 | 1.6161e17 | 1.7012e17 |
| Gar60-2fl-4rl | 1.4983e17 | 1.7603e17 | 1.7942e17 | 2.0168e17 |
| Gar60-2fl-5rl | 2.7382e16 | 3.2782e16 | 2.7152e16 | 3.1285e16 |

The two cases in which it was better to run many tabu searches were the two real-like instances in which the flow matrices were positively correlated. When there is a positive correlation between the objectives in a biobjective QAP, it implies that the Pareto optimal solutions are clustered somewhat closer together in the space, as solutions that are good with respect to one objective are likely to be good with respect to another objective positively correlated.

In the extreme case, such as instance `Gar60-2fl-5{uni,rl}` with a positive correlation of 0.8, all Pareto optimal solutions are quite close along the front. Fig.1 shows a scatter plot of some of the points sampled during the search by various types of search algorithms in the instances `Gar60-2fl-4rl` and `Gar60-2fl-5rl` respectively. Clearly visible in the plots, the algorithm sampled a much smaller region around the Pareto front on the positively correlated instance than it did on the negatively correlated instance.

Contrast this picture with that in which the objectives are anticorrelated. In that case, any single run of a local search can provide only a limited benefit, as there are solutions scattered across a wider range, so by definition, finding a single good point can help only so much. In order to find a diverse set of nondominated solutions, the algorithm must give itself more opportunities to explore different regions of the space. Given a limited budget of evaluations, the only ways to do this are to run shorter phases of local improvement (`nsga2_ts1`, `nsga2_ts3`), to apply the local search more infrequently (`nsga2_ts3`, `nsga2_ts4`), or to give up the evolutionary aspects of the search and just run as much tabu search as possible before exhausting the computational resources allotted (`nsga2_ts2`). However, here, as in [8], the evolutionary operators are quite effective in finding better starting locations for the tabu search. Thus, only when the correlation imposes locality constraints on the set of nondominated solutions does this third option become an effective choice.

In the three objective instances, shown in Table IV, we begin to see the picture change. The best strategy in most instances switches from `nsga2_ts4` (longer, more infrequent runs) to `nsga2_ts3` (shorter, more frequent runs). The former remains the best option for the anticorrelated uniform instance, but only just. Note that both strategies apply the local search only sporatically, thus relying more on the evolutionary search operators than the other two strategies which apply tabu search to every individual.

One possible explanation for this behaviour is that finding a diverse approximation to the entire Pareto front in three dimensions requires more points than in two dimensions. Thus, if the algorithm spends too much time in a single run of tabu search, it risks finding a single very good solution at the expense of adequate coverage in other regions of the front. In essence, the algorithm must rely on the evolutionary operators to pull their weight because they are so cheap compared to the cost of a full blown run of tabu search.

In `nsga2_ts3`, the average number of generations completed during a single run is about 1,945. In the `nsga2_ts4` variant, this number is only about 200, reflecting the order of magnitude difference in the expected number of tabu search runs during each generation. Similarly, the variants `nsga2_ts1` and `nsga2_ts2`, which perform tabu search on every individual, complete only 200 and 20 generations each run respectively.

Practically speaking, 20 generations is simply not enough for the evolutionary aspects of the search to perform their role in finding good starting locations for the search. With only 50,000 evaluations in a given tabu search run, the local search cannot effectively overcome the handicap it was handed in the form of an essentially random starting position, and the overall efficacy of the search suffers. It seems that one either must give the MOEA component enough generations to fulfill its function in aiding the tabu search via a good initial point, or the tabu search itself must be allowed to run long enough so that it can escape from a bad seed value. With the full local search performed by the `nsga2_ts1` and `nsga2_ts2` variants, we provide neither opportunity, and the performance suffers.

If this explanation is correct, we should expect to see even more obvious patterns in the four objective instances, as the number of points required to densely cover the front grows exponentially with the dimensionality of the objective space. Looking at Table V, we see this is indeed the case. The memetic strategy of sporatically running shorter bursts of tabu search is now the dominant strategy in every instance, sometimes by a much larger margin than previously seen, providing over a 100-fold increase in hypervolume for the uncorrelated uniform instance. By way of comparison, a sim-

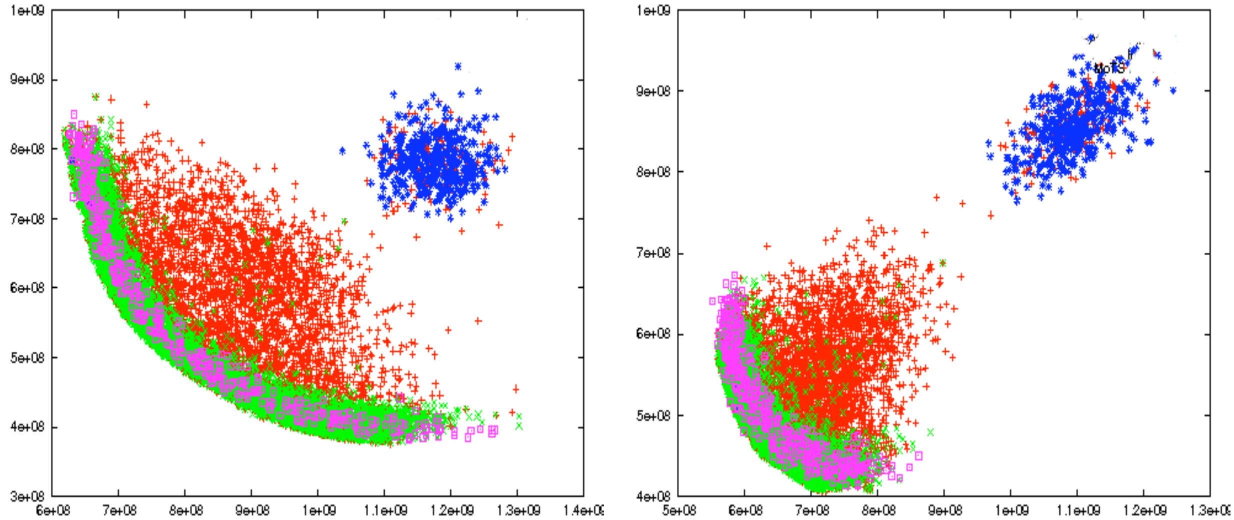


Fig. 1. Points sampled during the memetic search on the `Gar60-2fl-4rl` (left) and `Gar60-2fl-5rl` (right) instances. Note the restricted range in the positively correlated instance.

TABLE IV
MEAN HYPERVOLUME FOR EACH ALGORITHM ON ALL THREE OBJECTIVE PROBLEM INSTANCES.

| Instance | nsga2_ts1 | nsga2_ts2 | nsga2_ts3 | nsga2_ts4 |
|---|---|---|---|---|
| Gar60-3fl-1uni | 4.0688e17 | 3.5170e17 | 9.6030e17 | 6.3641e17 |
| Gar60-3fl-2uni | 1.7858e18 | 2.0373e18 | 2.0262e18 | 2.1104e18 |
| Gar60-3fl-3uni | 2.7206e16 | 6.2699e16 | 1.1713e17 | 6.5970e16 |
| Gar60-3fl-1rl | 5.1341e25 | 3.6926e25 | 1.0168e26 | 7.1308e25 |
| Gar60-3fl-2rl | 8.0894e25 | 6.9708e25 | 1.4180e26 | 8.8676e25 |
| Gar60-3fl-3rl | 1.3199e25 | 1.5642e25 | 3.3432e25 | 2.0691e25 |

ple multiobjective extension of Robust Tabu Search itself; an algorithm which does nothing more than apportion its available computational effort equally among 100 independent runs using different weight vectors, i.e., no evolutionary aspects at all, yields a mean hypervolume on the `Gar60-4fl-1uni` instance of 1.5515e24. This is better than three of the memetic strategies on this instance, but still nearly two orders of magnitude worse than the best strategy, `nsga2_ts3`.

It is somewhat expected that as the number of objectives increases, accompanied by a corresponding increase in the number of points required for a dense sample of the Pareto front, the multiobjective search becomes harder, requiring more search effort to find a good approximation. However, what these results tell us is that not just any increase in search time will yield the desired results. One must carefully match the *modus operandi* of the algorithm to the problem at hand to achieve the best results.

The ultimate goal is to be able to make informed decisions of this sort before committing the time and effort required to build and run different optimization algorithms. We would like to be able to use only the description of the problem along with readily obtainable information to determine what type of algorithm might be most likely to perform well. The above discussion provides one simple heuristic that we may state thusly. As the range over which desirable solutions are found increases in size, adjust the search to spend more time doing exploration. In the context of multiobjective memetic algorithms, this may mean decreasing the effort allotted to fine-tuning solutions via local search in favor of finding more areas to explore via evolutionary operators combined with shorter, less aggressive forms of local search.

It might be tempting to read this as a rule of thumb concerning the relationship between memetic algorithm performance and the dimensionality of the objective space. However, one must remember that there are other factors that affect the distribution of Pareto optima along the nondominated front. As shown clearly in the biobjective QAP instances, substantial differences can arise, even between different instances of the same optimization problem, and those differences can have noticeable impact on the relative performance of different search algorithms.

## VI. CONCLUSIONS

To find high quality solutions to most multiobjective combinatorial optimization problems, it is generally accepted that one needs to augment a multiobjective evolutionary algorithm with some form of local improvement operator. However, the picture is considerably more complex than that, as there are numerous choices to be made in designing a memetic algorithm that properly balances its constituent components. In this work, we took the view that the two components were competing for the limited usage of search time, and compared

TABLE V

MEAN HYPERVOLUME FOR EACH ALGORITHM ON ALL FOUR OBJECTIVE PROBLEM INSTANCES.

| Instance | nsga2_ts1 | nsga2_ts2 | nsga2_ts3 | nsga2_ts4 |
|---|---|---|---|---|
| Gar60-4fl-1uni | 3.1561e23 | 1.8578e23 | 8.6196e25 | 4.5066e23 |
| Gar60-4fl-2uni | 1.6395e24 | 1.9953e24 | 2.1983e24 | 1.9716e24 |
| Gar60-4fl-3uni | 1.9952e22 | 2.3616e22 | 4.4505e22 | 2.7999e22 |
| Gar60-4fl-1rl | 9.5741e33 | 5.1663e33 | 4.3686e34 | 2.0683e34 |
| Gar60-4fl-2rl | 1.3613e34 | 8.9712e33 | 3.2585e34 | 1.9453e34 |
| Gar60-4fl-3rl | 8.7276e33 | 4.1614e33 | 5.5425e34 | 2.4125e34 |

several different strateges on a wide range of instances of the multiobjective quadratic assignment problem.

Using widely used and well known components (NSGA-II and Robust Tabu Search), it was shown that both the number of objectives and the specific structure of the search space could have a profound impact on the performance of the different techniques, even when considering only different instances of but a single combinatorial optimization problem.

We demonstrated the need for a multiobjective memetic algorithm to ensure that a proper balance between evolutionary and local search operators is maintained. We showed that as the number of objectives is increased, the performance of these memetic algorithms shifted, with algorithms that promote exploration tending to perform significantly better, even with only three objectives. With four objectives, the performance difference between the algorithms increased even further.

Another potential approach is to replace the tabu search component of the algorithms described in this work with a faster, if somewhat less effective local search heuristic. While tabu search is known to perform quite well on QAP and mQAP, as a steepest descent search algorithm, it necessarily exhausts more evaluations early on than a next descent method. By replacing the tabu search with an algorithm that climbs quickly downhill, one could potentially combine some of the best of all worlds. Such an algorithm might form part of a highly effective memetic algorithm.

While there are still many questions that are far from settled, the fact that local search and evolutionary algorithms face different types of obstacles in highly multiobjective spaces would seem to suggest that it may be possible to combine these types of algorithms in novel ways to help alleviate some of the difficulties that plague current approaches to the optimization of problems with more than three or four objectives.

## REFERENCES


[1] P. Balaprakash, M. Birattari, and T. Stützle. Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. In *Hybrid Metaheuristics: Fourth International Workshop (MH'07)*, 2007.

[2] M. Birattari. *The Problem of Tuning Metaheuristics as Seen From a Machine Learning Perspective*. PhD thesis, 2004.

[3] M. Birattari, T. Stützle, , L. Paquette, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'02)*, 2002.

[4] D. Corne and J. Knowles. Techniques for highly multiobjective optimization: Some nondominated points are better than others. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'07)*, 2007.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[6] P. Fleming, R. Purshouse, and R. Lygoe. Many-objective optimization: An engineering design perspective. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO'05)*, 2005.

[7] C. Fonseca, L. Paquette, and M. López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06)*, 2006.

[8] D. Garrett and D. Dasgupta. Analyzing the performance of hybrid evolutionary algorithms on the multiobjective quadratic assignment problem. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2006.

[9] D. Garrett and D. Dasgupta. Multiobjective landscape analysis and the generalized assignment problem. In *Proceedings of the 2007 Learning in Intelligent Optimization Conference (LION'07)*, 2007.

[10] F. Glover. Tabu search – part I. *OSRA Journal on Computing*, 1(3):190–206, 1989.

[11] F. Glover. Tabu search – part II. *OSRA Journal on Computing*, 2(1):4–32, 1990.

[12] J. Knowles and D. Corne. Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization (EMO 2003), Second International Conference*, pages 295–310, Faro, Portugal, April 2003.

[13] J. Knowles and D. Corne. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In *Evolutionary Multicriteria Optimization*, 2007.

[14] J. D. Knowles and D. W. Corne. Towards landscape analyses to inform the design of a hybrid local search for the multiobjective quadratic assignment problem. In A. Abraham, J. R. del Solar, and M. Koppen, editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279. IOS Press, 2002.

[15] T. Koopmans and M. Beckmann. Assignment problems and the location of economics activities. *Econometrica*, 25:53–76, 1957.

[16] M. López-Ibáñez, L. Paquete, and T. Stützle. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. Technical Report AIDA–04–11, FG Intellektik, FB Informatik, TU Darmstadt, DEC 2004. Accepted by Journal of Mathematical Modelling and Algorithms.

[17] L. Paquette, C. Fonseca, and M. López-Ibáñez. An optimal algorithm for a special case of klee's measure problem in three dimensions. Technical Report CSI-RT-I-01/2006, Universidade do Algarve, 2006.

[18] L. Paquette and T. Stützle. A study of local search algorithms for the biobjective qap with correlated flow matrices. *European Journal of Operational Research*, 2006.

[19] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[20] E. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 2001.