Analyzing the Performance of Hybrid Evolutionary Algorithms for the Multiobjective Quadratic Assignment Problem

Deon Garrett Dipankar Dasgupta

***Abstract*— It is now generally accepted that the performance of evolutionary algorithms can nearly always be significantly improved through the inclusion of some form of local search. Most often, practitioners have developed hybrid algorithms in which all individuals created during the evolutionary process are subjected to a local improvement operator. This form of algorithm can be viewed as an evolutionary search for good starting points from which to apply the local search procedure and has proven very successful over a wide range of combinatorial optimization problems. However, a large number of possible implementation strategies exist for how best to incorporate the local search into the evolutionary process. In this work, we extend some commonly used static (fitness landscape) and dynamic (incorporating information concerning the run-time behavior of a particular search algorithm) analysis techniques into the multiobjective realm, and analyze the structure of the widely-studied multiobjective quadratic assignment problem. In particular, we show that the advantages of a state-of-the-art hybrid evolutionary algorithm over a simpler iterated local search algorithm can be explained reasonably well through a random walk analysis of the effects of recombination.**

I. Introduction

For a number of years, researchers have known that the performance of evolutionary algorithms can generally be improved through the addition of a local search operator. These hybrid evolutionary algorithms have been shown to outperform conventional evolutionary algorithms over a wide range of combinatorial and numerical optimization problems, often yielding state of the art performance for a particular problem. While there are no restrictions on how the global and local search aspects of these algorithms interact, by far the most popular approach is to perform some form of local search on every offspring produced by the evolutionary operators, inserting the improved individual into the offspring pool after the local search terminates.

For many combinatorial optimization problems, very efficient local search procedures exists which allow practitioners to apply a complete local search to every member of the population in a hybrid evolutionary algorithm. When such procedures are not available, these algorithms are unsatisfactory due to the extremely long run times required. In those cases, a better understanding of the manner in which the algorithm's components interact with the fitness landscape is needed to determine how best to integrate the local search operator into the evolutionary process. That is to say, if the computational requirements dictate that local search be applied sporatically or incompletely, we must address the question of how best to allocate the available resources toward solving the problem at hand. Currently, much of what is known in this area comes from studies that are experimental in nature. However, little is known concerning the detailed interactions between the structure of the fitness landscape and the performance of search algorithms. In the case of multiobjective optimization, one can argue that even less is known.

In multiobjective optimization, the lack of a single direction of improvement and the necessity of some form of diversity maintenance complicate the design of effective hybrid algorithms. However, aside from the additional degrees of freedom which render effective algorithm design more difficult, the nature of multiobjective evolutionary algorithms may provide some advantages as well. For example, in a typical simple genetic algorithm, convergence all but assures that crossover will become less effective as the search progresses. In MOEAs, however, the requirement of diversity maintainence provides some assurance that the explorative power of crossover can be maintained throughout the search. How best to utilize such information remains an open question.

Despite these additional avenues of research, most approaches to designing hybrid multiobjective evolutionary algorithms (MOEAs) are quite similar to their single-objective counterparts, often specifying a weight vector for each offspring and performing the local search on the resulting scalar objective function. Despite these differences, similar concerns to those described above need to be addressed. In this work, we present a fitness landscape analysis of the multiobjective quadratic assignment problem focused on the interaction between algorithm and landscape to attempt to provide additional insight into the design of effective hybrid MOEAs.

The remainder of the paper is organized as follows. In Section II, we briefly review the quadratic assignment problem and multiobjective quadratic assignment problem. Section III details the particular means of hybridization of local and evolutionary search used in this paper. The landscape analysis tools and results are presented in Section IV, while conclusions and further areas of research are described in Section V.

II. The Quadratic Assignment Problem

The quadratic assignment problem (QAP) is one of the oldest and most widely studied combinatorial optimization problems [1], [2]. First formulated by Koopmans and Beckmann in 1957[3], the QAP can be described as follows: Given

Deon Garrett is with the Department of Computer Science and the Institute for Intelligent Systems, University of Memphis, Memphis, TN, 38157, USA (email: jdgarrtt@memphis.edu)

Dipankar Dasgupta is with the Department of Computer Science and the Institute for Intelligent Systems, University of Memphis, Memphis, TN, 38157, USA (email: dasgupta@memphis.edu)

two $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, find a permutation π that minimizes

$$f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{\pi_i,\pi_j}. \quad (1)$$

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 [4], 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 [4], 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 [5], [6], [7].

In a pair of papers, Knowles and Corne proposed and provided detailed static analysis of the multiobjective QAP (mQAP) [8], [9]. 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 independent[1] single objective QAP problems. The objective function value of a permutation π is thus a k-dimensional vector with

$$f^m(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b^m_{\pi_i,\pi_j} \quad \forall m : 1 \leq m \leq k. \quad (2)$$

The mQAP models any sort of facilities layout problem in which the minimization of multiple simultaneous flows is required.

The mQAP has been studied by a number of researchers since its introduction [8], [9], [7], [10]. 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.

III. Hybrid Multiobjective Evolutionary Algorithms

Over the last decade, evolutionary algorithms have become quite prominent in the field of multiobjective optimization [11], [12]. It is often argued that the population of solutions inherent in most EAs can be exploited to produce a diverse set of nondominated solutions in a single run of the algorithm. In contrast, algorithms such as tabu search, simulated annealing, or classical mathematical programming techniques likely require multiple runs with differing parameter settings to provide a diverse set of solutions. A number of multiobjective evolutionary algorithms (MOEAs) have been proposed. Of these, the Nondominated Sorting Genetic Algorithm (NSGA-II) [13] and Strength Pareto Evolutionary Algorithm (SPEA2) [14] are commonly used, but several other effective MOEAs exist. As a hybridization of SPEA2 has previously been shown to be a state-of-the-art algorithm for the mQAP [7], we shall briefly describe both the SPEA2 algorithm and the aforementioned hybridization.

Essentially all modern MOEAs are elitist, in that they maintain and continuously update an external archive of nondominated solutions located during the run [15]. In SPEA2, each candidate solution is assigned a value called the *strength* equal to the number of other solutions it dominates. The fitness of each solution is computed as a function of the total strengths of all solutions that dominate the solution in question, as well as the distance to the nearest neighbors along the Pareto front. In order to maintain a fixed size archive, dominated individuals are allowed to enter the archive if too few nondominated solutions have been located, and nondominated solutions are removed from the archive if too many have been found. To reduce the archive size to the required level, the individual nearest to its nearest neighbor is removed. To break ties, distance to additional neighbors are considered until a clear choice emerges. This archive truncation measure has been shown to be very effective at maintaining diversity, but has a fairly large computational overhead.

It is widely accepted that augmenting evolutionary algorithms with some form of local search often greatly improves the performance of the algorithm. While other techniques are certainly possible, most published hybrid evolutionary algorithms apply the local search operator to all solutions generated by the evolutionary algorithm. The large number of successes in the single-objective optimization community arising from the use of these hybrid algorithms has led many researchers to extend this idea into the multiobjective realm. However, in multiobjective optimization, there is no simple analogue to the idea of climbing "downhill" as a local search in a single objective problem might do.

For combinatorial optimization, there are two basic means by which a local search may operate. First, we may induce a structure upon the neighborhood via the use of a scalarization of the objective functions. The alternative is to directly utilize some form of dominance to guide the local search. Most hybrid algorithms tend to use weight vectors to guide the local search, with M-PAES [16] the most notable counterexample.

In [7], the performance of hybrid SPEA2 and ant colony optimization (ACO) algorithms is analyzed with both a next-ascent hill climber and several different lengths of tabu search (TS) on the mQAP. In their comprehensive comparison, they concluded that the most promising approach was SPEA2 hybridized with very short runs of TS, noting that this algorithm currently provides the state of the art for mQAP. As such, we shall ignore the ACO algorithms in this work, while reiterating the authors' view that the preliminary nature of much of the multiobjective ACO research implies that much more needs to be done in this area before the utility of

[1]Technically, the problems need not be independent in that the flow matrices may be correlated, but we can treat them as independent for the purposes of formulating the problem.

ACO for multiobjective optimization is settled one way or the other. Because the Robust Tabu Search [5], provides a simple, efficient, and very effective tabu search strategy for the QAP, most hybrid algorithms for the mQAP are quite similar. The efficacy of RoTS means that most researchers choose to incorporate it into their algorithms, and the efficiency of each iteration allows all individuals to be improved without unacceptably long run times. As a result, the run time of most such algorithms is completely dominated by the time required to run large numbers of iterations of tabu search.

In hybridizing the SPEA2 algorithm, the authors chose to construct a set of μ distinct weight vectors, where μ is the population size in SPEA2. As each offspring is generated, it is assigned an index i according to the number of offspring generated before it in the current generation. The weight vector used to optimize the newly created offspring is given by

$$\left(\frac{\mu - i}{\mu}, \frac{i}{\mu}\right) \quad 1 \leq i \leq \mu. \tag{3}$$

These weights thus subdivide the interval between weight vectors $[0,1]$ and $[1,0]$ into equally sized chunks in an attempt to provide an even coverage of the Pareto front. The offspring is then improved according to this weight vector using $\ell \cdot n$ iterations of RoTS. Upon completion of the RoTS run, the improved offspring is inserted into the offspring population where it waits to compete with the other offspring to join the archive. In their experiments, SPEA2 with short runs of tabu search ($\ell = 1$) was found to be the most promising variant, producing state-of-the-art performance on several instances of the biobjective QAP.

Note that the scheme for generating the different scalarizations proposed in [7] and described above results exhibits a slight but systematic bias in favor of the second objective. For example, with $\mu = 4$, the weights for each of the four offspring will be $(0.75, 0.25)$, $(0.5, 0.5)$, $(0.25, 0.75)$, and $(0.0, 1.0)$. We have therefore modified their algorithm slightly so that the weights used for offspring number i are produced according to

$$\left(\frac{\mu - i - 1}{\mu - 1}, \frac{i}{\mu - 1}\right) \quad 0 \leq i < \mu. \tag{4}$$

With this modification, the weights for the four offspring are a more appropriate $(1.0, 0.0)$, $(0.67, 0.33)$, $(0.33, 0.67)$, and $(0.0, 1.0)$.

We were able to replicate the results reported by Lopez et.al. in [7]. In addition, we tested the ϵ-MOEA algorithm of Deb et.al. [17] hybridized with tabu search in a similar fashion. The results of these experiments are not reported here, but were competitive with those produced by SPEA2. While the ϵ-MOEA is much more efficient than SPEA2 in terms of computational effort per generation (more precisely, per fixed number of fitness evaluations), when every offspring is subjected to a run of tabu search, any efficiency gains are swamped by the local search computations. However, our results showed that the ϵ-MOEA could be a worthwhile alternative in the case of hybridizations utilizing infrequent or incomplete local search.

IV. Fitness Landscape Analysis for the mQAP

It is quite clear that no single algorithmic strategy is appropriate for all problems, multiobjective or otherwise. Differences in the structure of the fitness landscape between different problems or even between different instances of a single problem class can result in marked differences in performance of any metaheuristic. However, the relationship between the properties of the fitness landscape and the performance of a given algorithm are generally not well understood.

Several techniques have been proposed for categorizing problems based on properties of their fitness landscapes. For many of these techniques, there exist straightforward generalizations to the multiobjective realm. In the following paragraphs, we briefly describe some mechanisms for multiobjective fitness landscape analysis and detail the results obtained by applying each to the multiobjective quadratic assignment problem.

We wish to allow the hybrid SPEA2 algorithm to exhaust some predetermined amount of time, perhaps 10 minutes. On our experimental platform, we can perform roughly 250 million fitness evaluations in 10 minutes. For all results reported in this paper, we use a set of ten 60-facility mQAP problems generated using the generator developed and made available by Knowles and Corne [9]. The ten benchmark instances consist of five uniform and five real-like problems, one each with the correlation between flow matrices set to 0, 0.3, -0.3, 0.8, and -0.8 respectively. All algorithms are allowed to exhaust 250 million evaluations.

Note that in this paper, we adopt the terminology that a fitness evaluation may refer to either the full $O(n^2)$ evaluation as required by a randomly created solution or the $O(1)$ incremental update utilized by the local search procedure. While somewhat cumbersome, it avoids the need to report actual times, and given sufficient information such as the population size and number of generations of the surrounding evolutionary algorithm, still permits the reader to make direct comparisons to other methods. Given these computation limits, only 48 generations of SPEA2 are performed before reaching the evaluation limit. Stated another way, approximately $(50 \times 48)/250,000,000 = 0.00096\%$ of the computation time is allotted to searching via the evolutionary operators.

Despite the overwhelming majority of run time devoted to tabu search, hybrid algorithms have consistently been shown to outperform pure tabu search algorithms for the QAP and mQAP [7], [18], [19]. In [18], advanced landscape analysis techniques were developed to examine the performance of hybrid evolutionary algorithms for single objective optimization. The goal of this work is to adapt and interpret those and other similar techniques applied the unique challenges of multiobjective optimization.

All runs of SPEA2 utilized a population size of 50 and an archive of size 200, as recommended in [7]. The CX crossover operator [19] was used with probability 1.0 and a swap mutation was applied with probability 0.01. Every offspring was subjected to improvement via 60 iterations

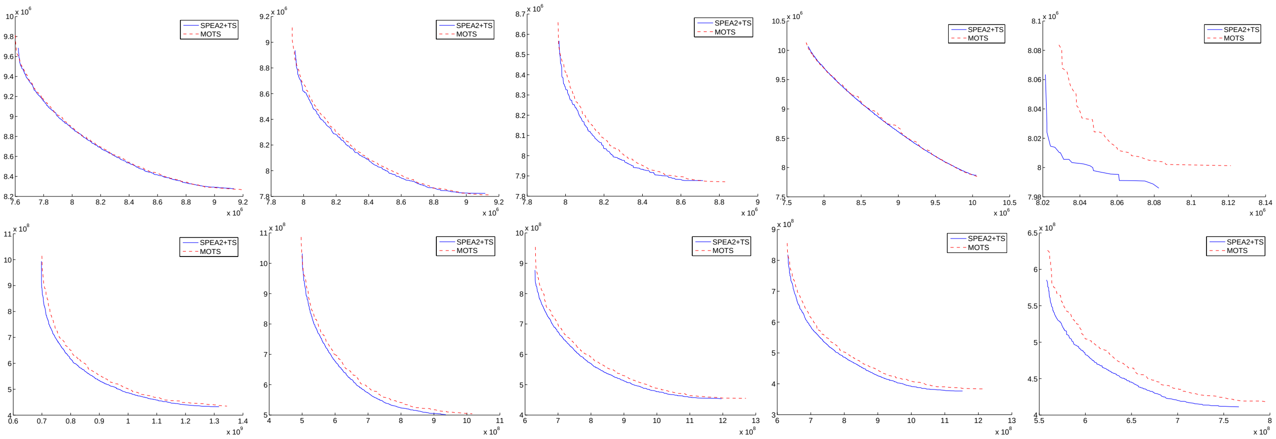

Fig. 1. Median empirical attainment surfaces (EAF) for SPEA2+TS and moTS for the 60-facility mQAP instances. Top row: uniform instances, bottom row: real-like instances. From left to right, c =-0.3, 0.0, 0.3, -0.8, and 0.8.

of RoTS using a particular weight vector assigned by the scheme described in the previous section. For comparison, a simple multiobjective tabu search (moTS) was implemented in which the same set of weight vectors as used by the hybridization of SPEA2 was used to optimize a randomly generated set of solutions. In the moTS implementation, a randomly generated solution was improved by 1000 iterations of tabu search using one of the weight vectors. The process continues randomly generating solutions and optimizing them with TS until 250 million evaluations have been exhausted. This design is meant to simulate the results one might get by simply running independent tabu searches. The limit of 1000 iterations for each solution helps to compensate for a few poor initial locations yielding strongly suboptimal final solutions by ensure that each weight vector is used a number of times. Figure 1 shows the median empirical attainment surfaces (EAF) for SPEA2+TS and moTS on each problem instance. As the goal here is not to demonstrate the superiority of one algorithm, we do not perform the sort of EAF analysis done in [7]. However, our results showed reasonably consistent performance, so the median EAF is representative of algorithm performance.

Thirty trials of each algorithm were executed on each problem instance. For all portions of the analysis below that require an estimate of the true Pareto front, the Pareto set approximations produced by all trials of all algorithms for each particular instance were concatenated, and the resulting set of nondominated individuals was considered as the Pareto front.

A. Fitness Distance Correlation

One of the more popular metrics by which problem difficulty may be measured, the fitness distance correlation coefficient (FDC) [20] measures the correlation between the fitness of a solution and the distance from the solution to the nearest globally optimal solution. For minimization problems, an FDC coefficient of 1.0 indicates that these quantities are perfectly correlated and that search should be easy. Often, FDC analysis is presented using scatter plots to augment or replace the actual coefficient of correlation.

In most hybrid evolutionary algorithms, the local search operator is applied to every newly generated offspring. In the case of combinatorial optimization problems, performance has generally been found to be improved if the local search is allowed to reach a local optimum. Thus, this type of algorithm is often viewed as performing a search in the space of local optima, or alternately of searching for good points from which to initialize a local search. In either viewpoint, the distribution of local optima is one of the most important factors in the difficulty of a particular problem or class of problems.

In order to perform FDC analysis, one requires knowledge of all global optima, a requirement which is clearly impractical for realistic problems. Thus, typically a high-performance metaheuristic is employed to generate a large set of solutions, the best of which is taken to represent the optimum.

In addition, several authors [21], [22] report results in which the FDC is misleading as a measure of problem difficulty. In spite of these results, however, FDC is indicative of problem difficulty for evolutionary algorithms on many problems, and does provide useful insight into the structure of the fitness landscape.

In the case of multiobjective optimization, we can consider the set of all globally optimal solutions to be the set of all Pareto-optimal solutions. Under such a characterization, the FDC in a multiobjective problem measures the correlation between a sample of local optima and their nearest points on the Pareto front. However, in using this FDC to estimate problem difficulty for most conventional hybrid MOEAs, we may underestimate the true difficulty. In most algorithms of this type, each newly generated candidate solution is assigned a particular weight vector which is used by the embedded local search procedure to improve the offspring. However, there is no reason to believe that the chosen weight vector will lead the search in the direction of the nearest Pareto optimal solution. As illustrated in Figure 2, a solution may be very near the

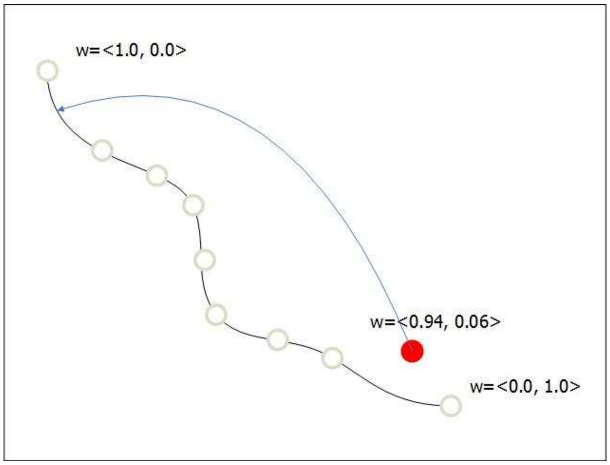

Fig. 2. In multiobjective optimization, the distance to the nearest Pareto-optimal solution may be misleading with typical hybrid algorithms due to the lack of a principled selection of weight vectors. In this example, the shaded point may be forced to move toward a distance area of the Pareto front rather than the short distance to the nearest point along the front.

Pareto front, yet require a lengthy search process to reach a local optimum very far away from the nearby Pareto optimal points. In Figure 2, the shaded point is generated according to some procedure such as crossover or mutation and assigned a randomly selected weight vector. The point is near the front, but the chosen weight vector forces the local search to move the point a much larger distance to try to reach a different region of the Pareto front.

Intuitively, the performance of such an algorithm might be improved if the search direction utilized by the local search could be chosen in such a way as to improve the likelihood of finding better solutions. However, the relationship between algorithm performance, search direction, and proximity to the Pareto front is in general unknown. Note that a similar issue could be raised in the case of single objective optimization. Multiple global optima may exist, and there is no guarantee that a search algorithm will move a given solution toward the nearest optimum. However, we assume that in practical applications, we do not know the location of all optimal solutions. In contrast, while we do not know the location of all Pareto optimal solutions in multiobjective problems, we do have information concerning the weight vectors associated with different regions of the space. Given that we have found other solutions similar to a given point, we can use the weight vectors used for these previously found solutions to guide the current solution in any direction we choose. How best to manage the selection of weights for use with the local search is likely problem dependent, and the impact of problem structure on these decisions is currently unknown.

It is of course known that a given point on the Pareto optimal surface is optimal with respect to a weight vector oriented normal to the supporting hyperplane at that point [11]. The discussion of appropriate selection of weight vectors is thus interpreted as selecting the point on the Pareto front which the search should progress toward. While such information can be valuable, many algorithms – including that of [7] which is adapted in this work – do not make use of this information. How best to incorporate the principled selection of weight vectors in local search remains an open problem.

Figure 3 shows fitness distance correlation scatter plots for the set of 60-facility uniform and real-like problem instances, each with five different values for the correlation between objectives. We see immediately that the uniform instances exhibit no correlation between fitness and distance to the Pareto front, while a small positive correlation exists for the real-like instances. We also note that the correlation between objectives plays little or no role in determining FDC. At the very least, we may conclude that any effects due to the correlation of flow matrices is dominated by the effects due to problem structure.

B. Random Walk Analysis

Merz in [18] models crossover and mutation in hybrid evolutionary algorithms as random walks initiated from local optima. A series of mutations is represented by an undirected random walk starting from a local optimum. In contrast, crossover is modeled as a random walk starting at one local optimum and ending at another. Provided that the crossover operator is respectful [23], [18], this random walk between local optima explores the space in which offspring will be produced.

Two general classes of evolutionary algorithms can be considered. On one hand, we have algorithms based primarily on mutation (evolution strategies, evolutionary programming), while on the other we have recombination centric algorithms such as genetic algorithms. Of course, a third possibility is the class of iterated local search algorithms which generally utilize neither recombination nor mutation, as in the moTS used as a baseline in this work. It should be noted that iterated local search algorithms often construct initial solutions based in part upon previous solutions found. However, as moTS simply performs random restarts, we ignore any mutation-like aspects found in other iterated local search algorithms.

According to the prevailing view of hybrid evolutionary algorithms, they operate by searching for good seed points for local search. If this view is correct, then a hybrid evolutionary algorithm should outperform an iterated local search algorithm only if the genetic operators produce offspring which, when passed to the local search, result in better local optima. With the exception of the instances exhibiting strong positive correlation, Figure 1 shows that moTS performs almost as well as SPEA2+TS on all problems. Thus, the observed differences must arise when the evolutionary algorithm is exploring near the Pareto front, and moTS is forced to restart from randomly chosen locations.

To examine how SPEA2+TS searches near the Pareto front, we look at the distribution of points reached via random walks initiated from points on the Pareto front. We generate 1000 random walks of length 60 beginning from points randomly selected from the Pareto front. In one set of trials, the random

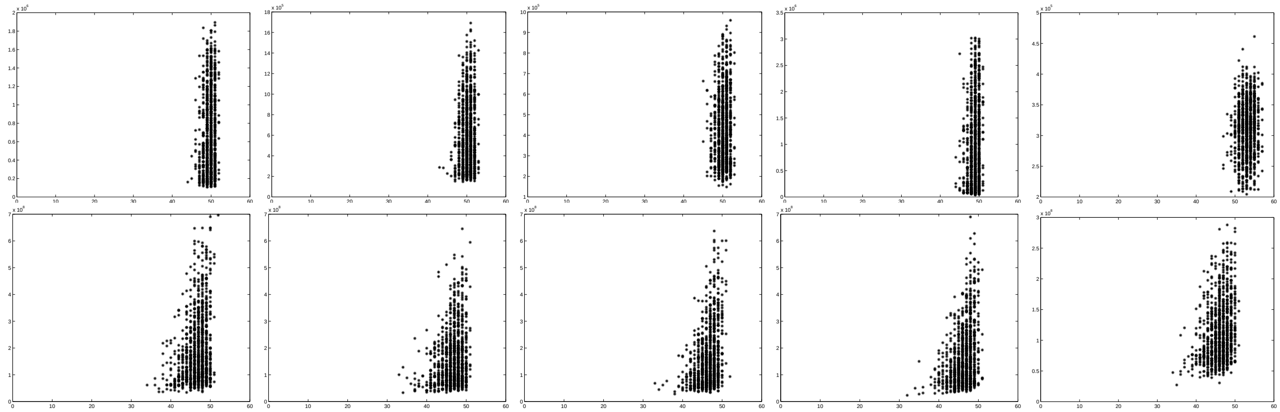

Fig. 3. Fitness distance correlation for the mQAP. Top row: uniform instances, bottom row: real-like instances. From left to right, c =-0.3, 0.0, 0.3, -0.8, and 0.8.

walks are undirected. In another, we choose not only an initial point, but also an end point, and take a random walk between the two. The former thus simulates a sequence of mutations and the latter a distribution of offspring generated by crossover.

Figures 4 and 5 show the distance in objective space and decision space between the points on the random walk and the nearest point on the Pareto front. Interpreting these graphs in the case of multiobjective optimization will benefit from a brief aside on the design space of multiobjective evolutionary algorithms with local search.

In most MOEAs, we forego scalar fitness assignment via a weight vector in favor of fitness based on Pareto dominance. However, in order to make effective use of an embedded local search procedure, we typically assign each solution a weight vector solely for use in the local optimization stage. Within this very general framework, numerous possibilities exist. For example, an algorithm may provide a means of incest prevention to force recombination of solutions from different regions of the Pareto front, or enforce mating restrictions requiring that only similar solutions be allowed to participate in recombination.

Another possible design feature lies in the selection of the weight vectors used by the local search. Because MOEAs are population-based, we have information available with which we can estimate the results of applying local search to a particular combination of solution and weight vector based upon previous experience. For example, given two solutions A and B on the Pareto front generated with weight vectors w_A and w_B, we may choose a weight vector by somehow averaging w_A and w_B to attempt to find a solution C on the Pareto front between A and B.

The success of such an algorithm will clearly depend in some way on the ability of the search operators to actually manage to produce the point C on the Pareto front, as opposed to rediscovering A or B or terminating at a strongly suboptimal point away from the front entirely. Intuitively, the odds of success are increased if the offspring produced by the genetic operators are reasonably close to the front prior to any local optimization.

Figure 4 shows the distance in objective space to the nearest Pareto optimal solution, where the distance to the front is also expressed in objective space. Note that distances in the objective space refer to the Euclidean distance between two objective function vectors throughout this paper. Thus, this figure essentially shows the fitness of solutions produced by the simulated mutation and recombination operators. Note that this graph does not show the distance in decision space, i.e., the number of swaps necessary to reach the Pareto front. In general, the entropy of Pareto optimal solutions for the mQAP is quite high, indicating that they are spread somewhat uniformly through the space, although some clustering may occur in particular regions [8]. As such, one does not necessarily expect that two points very near to one another on the Pareto front are particularly similar in the decision space, and thus the number of swaps needed to reach any particular location on the front is likely to be highly erratic as we consider different initial and final points in a local search.

Comparing Figure 4 with Figure 1 shows a noticeable correlation between the effectiveness of recombination and the advantage of SPEA2+TS over moTS on the instances considered. From the EAF plots, we see that the largest difference in performance was on the strongly positively correlated instances (the rightmost column in both figures). The high correlation between objectives in these instances results in fewer Pareto optimal solutions, and a decrease in entropy of the points in the Pareto front. Therefore, offspring tend to be nearer to the front, because they are more constrained by the proximity of both parents than would be the case with many parents with high entropy to choose from. In contrast, the strongly negatively correlated instances (column four in the figures), exhibit the opposite behavior. The entropy across the Pareto front is much higher, and as a result, recombination produces results similar to the undirected random walk. In this case, the expected offspring (equidistant from either parent) tend to resemble randomly generated solutions and thus there

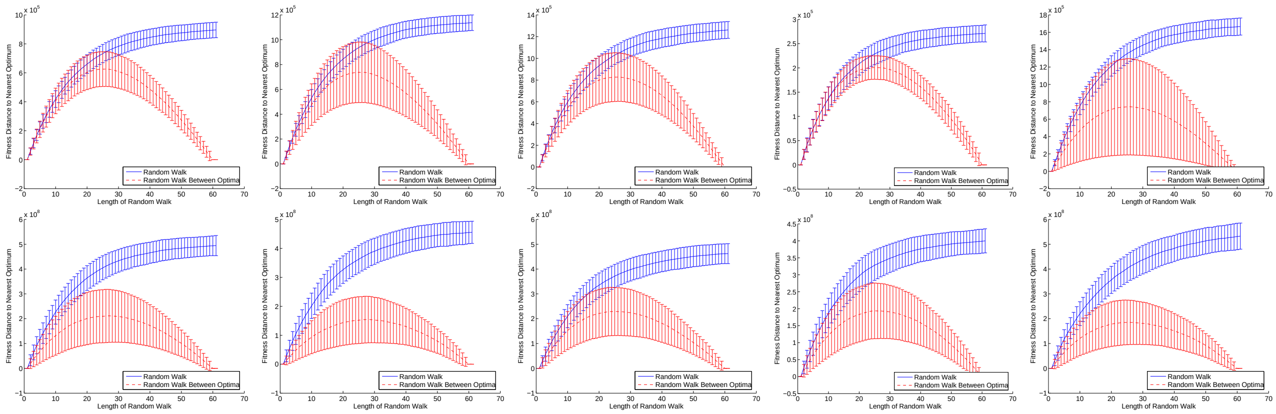

Fig. 4. Distance in objective space (Euclidean distance between objective vectors) to nearest Pareto optimal point from successive points on directed and undirected random walks. From left to right, $c =$-0.3, 0.0, 0.3, -0.8, and 0.8.

is no performance gain expected from recombination over moTS.

The correlation between the effectiveness of recombination and algorithm performance is not perfect. Recombination on the uniform instance exhibiting small positive correlation (row 1, column 3 in Figure 4) looks very similar to that of the uncorrelated uniform example. However, the difference in the EAF plots for those instances show that SPEA2+TS does outperform moTS on the former by a significantly greater margin than on the latter. Such results indicate that certain features of the landscape that are not considered in this analysis affect the overall effectiveness of the algorithms.

Figure 5 shows graphically the impact of high entropy. The same random walks as shown in Figure 4 are shown here, but the distances considered are in the decision space, i.e., the minimum number of swaps necessary to reach any point on the Pareto front. The directed random walk simulating recombination is very telling. We choose two solutions from the Pareto front at random and perform a random walk between the two. The most likely outcome for any of the tested problem instances is that the nearest Pareto optimal point for the first half of the walk is the initial point, and the nearest for the second half is the destination. In other words, it is unlikely that crossover produces an offspring that is nearer to any individual on the Pareto front than to one or both of its parents.

Again, the cases exhibiting large magnitude correlation of objectives yield different behavior than in the other cases. For the positive correlations, recombination appears to often provide solutions very near the Pareto front in terms of distance in decision space. When the correlation is strongly negative, however, the curves traced out by recombination and mutation are indistinguishable until halfway through the walk when recombination forces the solutions to move back toward the destination on the Pareto front.

V. Conclusions

While we know that the structure of a fitness landscape has a profound effect on the performance of a given search algorithm, very little is known concerning exactly how the relationship manifests itself. Most existing studies are very empirical in nature, focusing primarily on the quality of the final results. While providing useful guidelines to practitioners working on similar problems, these results often fail to provide information concerning the reasons behind the superiority of one algorithm over another. A better understanding of the ways in which different algorithms navigate the search space could provide a more solid foundation on which better algorithms and further improvements to existing algorithms can be made.

In multiobjective optimization in particular, little is known about how structural properties of the fitness landscape impact search behavior. Experimentally, it has been shown that multiobjective evolutionary algorithms for many combinatorial optimization problems can be significantly improved via the addition of some form of local search. This is no surprise, as similar statements have been made in the case of single objective optimization for years. However, even in the single objective case, little is known concerning the best way to incorporate local search into the evolutionary process. Fortunately, for many combinatorial problems, very efficient local search algorithms can be developed which allow the practitioner to sidestep some of these issues by simply running a complete local search on every solution encountered. However, in other cases, such an approach is infeasible due to the lack of an efficient fitness update procedure. In these cases, it is vital to understand the manner in which these complex hybrid algorithms search the fitness landscape in order to develop effective algorithms.

In this work, we present an analysis of the fitness landscape of the multiobjective quadratic assignment problem that differs from earlier approaches in that we attempt to directly tie the analysis to the run time behavior of hybrid evolutionary algorithms. Static landscape properties provide a great deal of useful insight. However, by measuring the interaction between algorithm and landscape, additional information is gained concerning the means by which better performing algorithms can be developed.

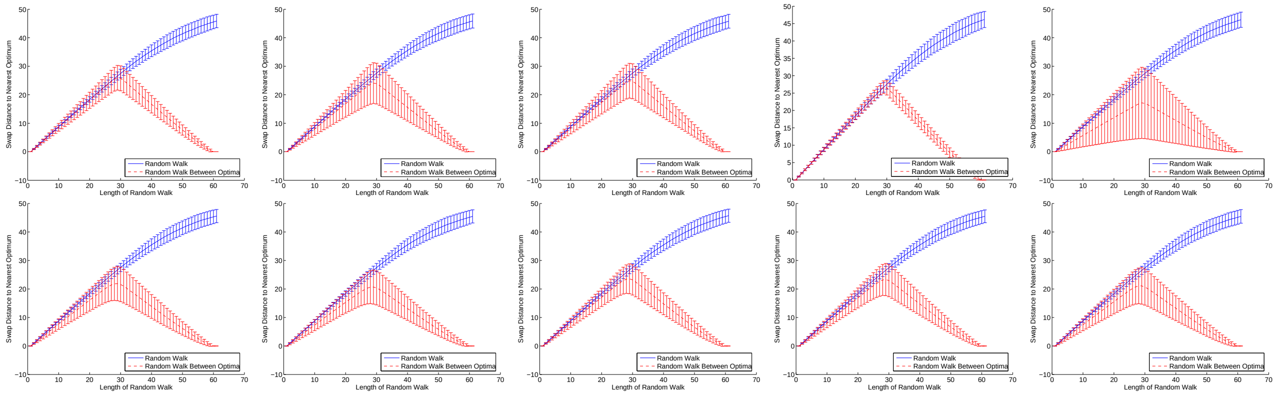

Fig. 5. Distance in decision space to nearest Pareto optimal point from success points on directed and undirected random walks. From left to right, $c =$-0.3, 0.0, 0.3, -0.8, and 0.8.

We have shown a correlation between the advantage obtained through a hybrid MOEA versus a simpler iterated local search algorithm and the distribution of offspring generated via recombination. As the correlation between objectives increases, the distance between good offspring and the Pareto front decreases, thus allowing the embedded local search procedure to exploit these good initial locations to find improved Pareto optima.

The rudimentary analysis provided in this paper is a small step toward a better understanding of the interaction between the fitness landscape and the behavior of hybrid evolutionary search algorithms. The conclusions derived for the biobjective QAP are almost certainly specific to that problem. However, the general approach described in this work may be directly applied to other combinatorial multiobjective optimization problems to better understand the structure of these other problems. While this work focused primarily on the effect of recombination, it seems clear that other factors play a vital role in determining the performance of hybrid MOEAs. Much more work is needed to determine how additional features of a landscape can affect algorithm performance.

References

[1] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishing, 1998.

[2] L. M. Gambardella, E. D. Taillard, and M. Dorigo, "Ant colonies for the quadratic assignment problem," *Journal of the Operational Research Society*, vol. 50, no. 2, pp. 167–176, 1999.

[3] T. Koopmans and M. Beckmann, "Assignment problems and the location of economics activities," *Econometrica*, vol. 25, pp. 53–76, 1957.

[4] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM*, vol. 23, no. 3, pp. 555–565, 1976.

[5] E. D. Taillard, "Robust taboo search for the quadratic assignment problem," *Parallel Computing*, vol. 17, pp. 443–455, 1991.

[6] ——, "A comparasion of iterative searches for the quadratic assignment problem," *Location Science*, vol. 3, pp. 87–105, 1995.

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

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

[9] ——, "Instance generators and test suites for the multiobjective quadratic assignment problem," in *Evolutionary Multi-Criterion Optimization (EMO 2003), Second International Conference*, Faro, Portugal, April 2003, pp. 295–310.

[10] 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.

[11] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[12] C. A. Coello-Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

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

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

[15] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[16] J. Knowles and D. Corne, "M-PAES: A memetic algorigm for multiobjective optimization," in *In Proceedings of the Congress on Evolutionary Computation (CEC00)*, 2000, pp. 325–332.

[17] K. Deb, M. Mohan, and S. Mishra, "Towards a quick computation of well-spread pareto-optimal solutions," in *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference*, 2003, pp. 222–236.

[18] P. Merz, "Advanced fitness landscape analysis and the performance of memetic algorithms," *Evolutionary Computation*, vol. 12, no. 3, pp. 303–325, 2004.

[19] P. Merz and B. Friesleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 337–352, 2000.

[20] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," in *Proceedings of the 6th International Conference on Genetic Algorithms*, L. Eshelman, Ed., 1995, pp. 184–192.

[21] L. Altenberg, "Fitness distance correlation analysis: An instructive counterexample," in *Proceedings of the 7th International Conference on Genetic Algorithms*, T. Bäck, Ed., 1997, pp. 57–64.

[22] R. Smith and J. Smith, "An examination of tuneable, random search landscapes," in *Foundations of Genetic Algorithms 5*, W. Banzhaf and C. Reeves, Eds., 1999, pp. 165–181.

[23] N. Radcliffe, "Forma analysis and random respectful recombination," in *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.