Genetic operators, the fitness landscape and the traveling salesman problem

Keith Mathias and Darrell Whitley
Computer Science Department, Colorado State University, Fort Collins, CO 80523 USA
mathiask/whitley@cs.colostate.edu

Abstract

Edge Recombination and Maximal Preservative Crossover (MPX) are two operators developed to preserve edge information for the Traveling Salesman Problem. Manderick et al. introduced the notion of a *fitness landscape* to measure the fitness correlation between parents and offspring under different recombination operators. The work on the fitness landscape is extended by studying the interrelationship between the fitness landscape, operator failure rates (in terms of non-inherited edges) and the effect of operator failure on tour length. The use of local improvement operators is also examined.

1. MOTIVATION

The application of genetic based search to Traveling Salesman Problems (TSP) of several hundred cities has produced encouraging results. Mühlenbein [5], Ulder et al. [7], Gorges-Schleuter [2], and Eshelman [1] all report near optimal results on the Padberg 532-city problem. All of these approaches combine local search in the form of 2-Opt with genetic search. We look at two recombination operators, Edge and Maximal Preservative Crossover (MPX), developed especially for the TSP. To better understand the computational behavior of these genetic operators we examine their recombination behavior using various metrics, including the ρ_{op} metric introduced by Manderick et al. [4] to look at the *fitness landscape*.

1.1. Alphabet cardinality

One thing that distinguishes the application of genetic algorithms to permutation problems such as the TSP from other optimization problems is the nature of the encoding. A great deal of effort has been expended creating crossover operators that recombine sequence permutations while maintaining feasibility and transferring as much critical adjacency information from parents to offspring as possible. However, alphabet cardinality is also a critical issue, especially as the size of the permutation problem becomes larger.

Each tour in a TSP is a Hamiltonian cycle on a *fully connected graph* where each city is a vertex. The fully connected graph for an N city TSP has $(N^2 - N)/2$ edges. For a 100 city problem, the corresponding fully connected graph has 4950 edges. If each tour sampled 100 unique edges, then at least $\lceil (N-1)/2 \rceil$, or 50 tours would be needed to cover the graph. By covering the graph, we mean that every edge in the graph is included at least once in the population of N/2 tours.

Generating a minimum sample of $\lceil (N-1)/2 \rceil$ tours seems to be a nontrivial problem since generating each tour requires generating a Hamiltonian cycle in a partial graph and finding a Hamiltonian cycle in an arbitrary partial graph is NP-complete. At the very least *N/2 is a clear lower bound on the population size needed to cover all possible edges.* In the best case, population sizes must be of at least of O(N) with respect to the number of cities in the tour to sample all edges once. In the worst case, population sizes may need to grow as a polynomial function of $O(N^2)$ with respect to the number of cities in a TSP, especially if edges are to be sampled more than once.

While this view of population size requirements may be simplistic, it does raise a critical issue. Genetic algorithms have been applied to problems with 500 and 600 cities with some success using very small population sizes. Eshelman's [1] results for the Padberg 532-city problem [6] were within 0.1% of optimal and were obtained using a population size of only 50 strings. The only way a genetic algorithm using such small population sizes can produce good results on problems where a permutation encoding is used is to introduce new edges into the population during search. This can be done either by mutation or by local hill-climbing. Thus, it is important to ask how local operators, such as 2-Opt, impact inheritance during genetic search and how they interact with recombination operators.

1.2. A view of the fitness landscape

Recently, Manderick et al. [4] introduced a way of looking at the relationship between operators, inheritance and fitness. An operator dependent view of the fitness landscape is obtained by calculating the *correlation coefficient* ρ_{op} between the fitness of the parent strings and the fitness of the offspring produced using different operators.

$$\rho_{op}(F_p, F_c) = Cov(F_p, F_c) / [\sigma(F_p)\sigma(F_c)] \qquad (1)$$

A higher correlation coefficient suggests better preservation of information during inheritance and a better ability to exploit the fitness landscape. Manderick et al. [4] applied the ρ_{op} metric to several operators for the TSP. The resulting ranking was consistent with their empirical performance results, as well as the results reported by Starkweather et al. [8]. The best of these operators proved to be *Edge recombination* [9].

The ρ_{op} metric would appear to be a very useful way of examining operator effectiveness. This work is extended in three ways. First, Edge recombination is compared to the MPX operator introduced by Mühlenbein [5]. MPX was not included in either of the previous studies. Second, in order to better understand the ρ_{op} metric, we examine various factors that appear to contribute to the fitness correlation. Finally, these same factors are examined in the context of parents and offspring improved by 2-Opt.

1.2.1. Factors contributing to fitness correlation

The work of Manderick et al. [4] shows that some recombination operators produce offspring with a fitness more correlated to parent fitness than others. One obvious reason for the differences in operators is their ability to transfer information in the form of edges passed from parents to offspring. We refer to a *foreign edge* as an edge that is introduced into an offspring which does not appear in either parent. Introduction of a foreign edge represents a *failure*, which occurs because a partial tour has been built up to some city X and neither of the two parents have a connection to another city from X that has not already been used in constructing the partial tour. The *failure rate* is a count of the number of foreign edges introduced into an offspring and represents another method of evaluating an operator. We would expect failure rate to be correlated with the ρ_{op} metric.

Foreign edges will typically degrade the fitness of the offspring and reduce the correlation between the offspring's fitness and the parents' fitness. However, instead of just counting the number of foreign edges, one can directly measure the contribution of these foreign edges to the total length of the offspring's tour. The *failure-error* is obtained by summing the combined length of the foreign edges.

Finally, if two parents are highly fit, their offspring may have a lower fitness than the parents. Consider two parents that have been locally optimized. The edges that compose the parents should all be relatively good. However, failures during recombination will mean that foreign edges are introduced into the offspring. If foreign edges are randomly chosen, they can be arbitrarily large and will have a significant impact on fitness. Even with failure rates of 5%, foreign edges may be a significant factor in offspring fitness.

To better understand the impact of failures and their interaction with local optimization, we introduce a final metric. *Insertion rate* indicates how many offspring are produced that are at least as good as the best P strings produced so far, where P is the population size. Insertion rate is a useful and general measure of how successful any genetic algorithm is at discovering improved points in the search space. Insertion rate can be calculated regardless of what kind of genetic algorithm is actually used during search.

In our experiments we use the GENITOR algorithm [9]. In this algorithm, offspring replace the worst member in a population of size P+1. If we think of the "worst member" slot as a buffer that is not actually part of the population, then the remaining P members are the best P strings encountered so far during the search. This means that the population (minus the buffer) is monotonic. Therefore, *insertion rate* is an especially important metric with respect to this particular algorithm.

2. RECOMBINATION AND LOCAL OPERATORS

A brief overview of Edge recombination and the MPX operator is provided. A new enhancement to Edge recombination is also introduced as Edge-3. A previous enhancement to the original edge recombination operator is described by Starkweather et al.[8]; which we refer to as Edge-2.

2.1. Edge-2 recombination

The edge recombination operator uses an "edge table" to perform recombination. The "edge table" is an adjacency table listing the edges into and out of a city as observed in the two parent tours. The edge table is used to construct the offspring so that a minimal number of foreign edges are introduced into the offspring, while emphasizing edges common to both parents. Consider the following tours as parents to be recombined:

Parent 1: g d m h b j f i a k e c   Parent 2: c e k a g b h i j f m d.

An edge list is constructed for each city in the tour. The edge list for some city a is composed of all of the cities in the two parents that are adjacent to city a. If some city is adjacent to a in both parents, this entry is flagged (using a negative sign). Figure 1 shows the edge table which is the collective set of edge lists for all cities.

city	edge list	city	edge list	city	edge list	city	edge list
a	-k, g, i	d	-m, g, c	g	a, b, c, d	j	-f, i, b
b	-h, g, j	e	-k, -c	h	-b, i, m	k	-e, -a
c	-e, d, g	f	-j, m, i	i	h, j, a, f	m	-d, f, h

Figure 1. Edge table.

City *a* is randomly chosen as the first city in the tour and occurrences of *a* are removed from all edge lists. The tour is then extended by choosing the next city according to the following priority scheme: 1) flagged cities have first priority; 2) cities whose own edge list has the fewest entries have second priority. Ties are resolved randomly. City *k* is chosen as the second city in the tour since the edge *[a - k]* occurs in both parent tours. All occurrences of city *k* are removed from the edge table. City *e* is chosen from the edge list of city *k* as the next city in the tour since this is the only city remaining in *k's* edge list. This procedure is repeated until the partial tour contains the sequence: [a k e c].

At this point there is no deterministic choice for the fifth city in the tour. Edges to cities *d* and *g* occur only once in the parent tours and both the edge list for city *d* and city *g* have two unused edges remaining. Therefore city *d* is randomly chosen to continue the tour. The normal deterministic construction of the tour then continues until position 7. At position 7 another random choice is made between cities *f* and *h*. City *h* is selected and the normal deterministic construction continues until we arrive at the following partial tour: [a k e c d m h b g].

In this situation, a failure occurs since there are no edges remaining in the edge list for city *g*. City *i* has been chosen randomly from the list of all remaining cities to continue the tour, thereby introducing a *foreign edge*. The tour can now be completed in the normal deterministic fashion. The final tour, **[a k e c d m h b g i f j]**, contains two foreign edges: *[g - i]* and *[j - a]*. The last edge, *[j - a]*, which completes the Hamiltonian cycle, is not directly inherited but is a side-effect of tour construction.

2.2. Edge-3 recombination

The Edge-2 recombination operator has been used to generate optimal solutions on smaller TSPs (e.g. 100 cities). This has been attributed in part to the low number of failures observed, which introduces fewer foreign edges. Edge-3 recombination was designed to use the same basic mechanism as Edge-2 with an additional failure guarding mechanism. When a potential failure occurs during Edge-3 recombination, we attempt to continue construction at a previously unexplored terminal point in the tour.

A *terminal* is a city which occurs at either end of a partial tour, where all edges in the partial tour are inherited from the parents. The terminal is said to be *live* if that city still has entries in its edge list; otherwise it is said to be a *dead* terminal. Because city *a* was randomly chosen to start the tour in the previous example, it serves as a new terminal in the event of a failure. Conceptually this is the same as inverting the partial tour to build from the other end. This situation is best described by the following sequence of events:

1) Start Tour	$[L_1]$	4) Reverse Partial Tour	$[D_1 \;....\; L_1\ ]$
2) Build Tour	$[L_1 \;....\; L_*\ ]$	5) Continue Partial Tour	$[D_1 \;....\; L_*\ ]$
3) First Failure Occurs	$[L_1 \;....\; D_1\ ]$	6) Second Failure Occurs	$[D_1 \;....\; D_2\ ]$

In this representation scheme, L_1 represents the city chosen to randomly start the tour. It is a live edge, but after step 1, it is not currently active. L_* is used to indicate a *live* terminal that is currently being extended. D_1 represents the first dead terminal and D_2 represents the second dead terminal.

After both terminals on the partial tour are dead, we now have no choice but to start a new subtour. A new city, L_3, is randomly chosen as the new live terminal. The construction of the next partial tour is illustrated as follows:

7) Start Subtour	$[D_1 \ldots D_2]\ [L_3]$	11) Reverse Subtour	$[D_1 \ldots D_2]\ [D_3 \ldots L_3\ ]$
8) Build Tour	$[D_1 \ldots D_2]\ [L_3 \ldots L_*\ ]$	12) Continue Subtour	$[D_1 \ldots D_2]\ [D_3 \ldots L_*\ ]$
9) Third Failure	$[D_1 \ldots D_2]\ [L_3 \ldots D_3\ ]$	13) Fourth Failure	$[D_1 \ldots D_2]\ [D_1 \ldots D_2\ ]$

When a failure occurs, there is at most one *live* terminal in reserve at the opposite end of the current partial tour. In this example, the unused terminal at the opposite end of the partial tour was assumed live. In fact, it is not guaranteed to be live, since the construction of the partial tour could isolate this terminal city. Once both terminals of the current partial tour are found dead, a new partial tour must be initiated. Note that no local information is employed.

The example in section 2.1 can be used to compare Edge-2 and Edge-3 recombination. The offspring is again built using city a as the first city in the tour. City a is saved as the reserve terminal. The tour is built exactly the same way as with Edge-2 recombination until the failure occurs at city g where the tour was: [a k e c d m h b g]. Since there is a reserve terminal (i.e., city a) the edge list of city a is examined for a city that has not been previously used in the tour. In this case, the only city remaining in the edge list of city a is city i. If there were others then the next city would be chosen according to the previously described priority scheme.

The partial tour inclusive of the live and dead terminals is reversed (i.e., [g b h m d c e k a]). Then city i is added to the tour after city a. The tour is then constructed in the normal fashion. In this case, there are no further failures. The final offspring tour is: [**g b h m d c e k a i f j**]. The offspring produced has a single foreign edge (i.e., *[j - g]*.)

2.3. Maximal preservative crossover (MPX)

MPX and variants of MPX have been shown to be effective in solving TSPs [1]. This operator was designed to produce offspring similar to both parents in terms of edge information [5]. MPX produces offspring by first directly copying a segment from the Donor parent into the offspring. Cities are added consecutively to the offspring from the other parent (Receiver) with a hierarchy of rules for handling failures and redundancies (cities already present in the offspring). All of these rules are based on consecutive order.

Mühlenbein defines the operator [5] such that the length of the initially copied segment is a random value between two bounds. Ulder et al. [7] appear to define the segment length to be a fixed value which is 1/3 of the number of cities in the tour. Here, MPX has been empirically found to perform better with a fixed segment length of 1/3.

2.4. Local hill-climbing operators

The 2-Opt heuristic [3] has been used to optimize TSP tours in connection with genetic search. 2-Opt removes two edges in a tour, then one of the resulting segments is reversed and the two segments are reconnected. If 2-Opt results in an improved tour, the change is kept. Otherwise the tour is returned to its original form. 2-Opt is typically applied to all $(N^2 - N)/2$ pairs of edges. If any improvement is found the process can be reapplied to the set of all edges. When no further improvements are found the tour has *converged* to a local optimum with respect to 2-Opt. A *pass* is defined to be a *single application* of 2-Opt to all pairs of edges in a tour. The number of passes required to reach convergence will not be the same for all tours. Since each pass is of $O(N^2)$, running 2-Opt to convergence is computationally expensive. The computational expense can be reduced by performing a single *pass* over the tour such that all tours require exactly $(N^2 - N)/2$ evaluations. Typically there is also more improvement to be gained on the first pass than on subsequent passes. One way to combine 2-Opt with genetic search is to use *1-Pass of 2-Opt*. This can be used to enhance the initial population before genetic search, thereby increasing the concentration of quality edges in the population.

Another adaptation of the 2-Opt heuristic is *2-Repair* [2, 1]. This operator, which is specifically designed to work with genetic recombination operators, requires significantly fewer comparisons than 2-Opt. 2-Repair is the same as 2-Opt except that the comparisons are only performed on the pairs of foreign edges introduced into the offspring during recombination. This in effect connects the partial tours in a locally optimal fashion with respect to 2-Opt without disturbing inherited edges. By applying 2-Repair until no further improvements are found, all pairwise orderings of the partial tours are considered.

3. EXPERIMENTAL RESULTS

Our experimental results are presented in two sections. First, the ρ_{op} metric and other measures that help to explain operator effectiveness are examined. Then empirical results for the Padberg 532-city problem are presented in relation to the preceding metrics.

3.1. The ρ_{op} metric results and analysis

All tests shown in Figures 2 - 5 were performed using Edge-2, Edge-3, MPX with a fixed segment length of exactly 1/3 the tour cardinality (referred to as MPX-fixed) and MPX with a random segment length of at most 1/3. Edge-3 and MPX-fixed exhibited consistently better performance than the other operators and are the primary focus of subsequent experiments. All experiments were performed using GENITOR with a linear selective bias of 1.25 and a population of 5000, unless specified otherwise.

In this section we compare the Edge-3 and MPX-fixed operators with and without the use of *1-Pass of 2-Opt*. In these comparisons 1-Pass of 2-Opt can be applied to either the parents (i.e., the initial population), or the parents and subsequent offspring. In Figures 2 to 5, **PC** refers to tests where no 2-Opt is used. **P2C** indicates that 1-Pass of 2-Opt is applied only to the parents. **P2C2** refers to tests where 1-Pass of 2-Opt is used to improve the parents in the initial population and the offspring after each recombination. We largely explore the **PC** and **P2C** situations to avoid the interference of hyperplane sampling and the extra work associated with applying 1-Pass of 2-Opt to each offspring (i.e., **P2C2**).

The ρ_{op} metric comparisons in Figure 2 indicate that Edge-3 and MPX-fixed have very similar fitness correlations in the **PC** test case. The fitness correlation is significantly reduced when 1-Pass of 2-Opt is used to improve the initial population, the offspring, or

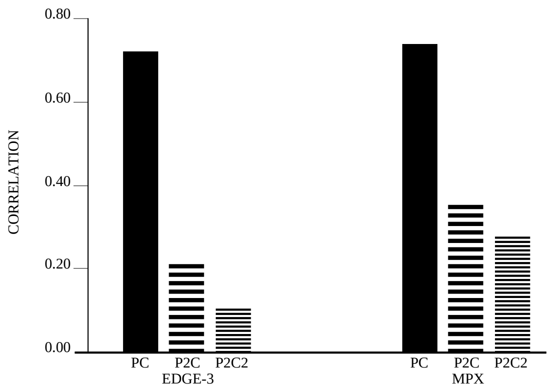

Figure 2. Parent/Offspring fitness correlation.

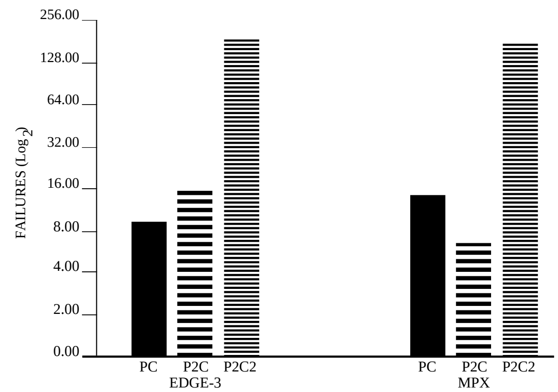

Figure 3. Average failures per recombination.

both. However, Figure 2 also indicates that MPX-fixed displays higher fitness correlations than Edge-3 when 1-Pass of 2-Opt is applied.

Figure 3 shows the average failures per recombination measured in an initial population. When no 2-Opt is applied, Edge-3 exhibits a failure rate that is approximately half that of MPX-fixed. (Note the logarithmic scale.) However, when the initial population is improved using 1-Pass of 2-Opt (i.e., **P2C**), MPX-fixed exhibits a failure rate less than half that of Edge-3 using 1-Pass of 2-Opt.

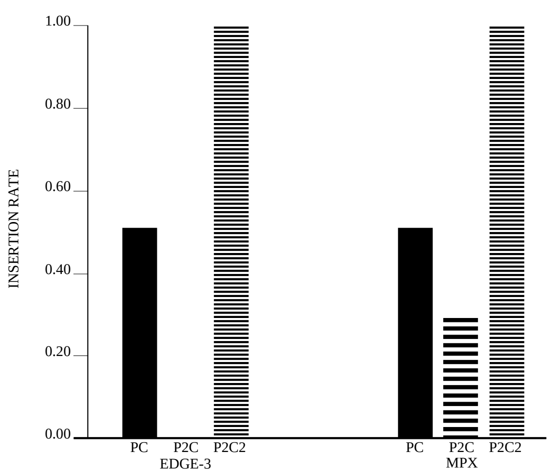

Figure 4. Insertion rate of offspring.

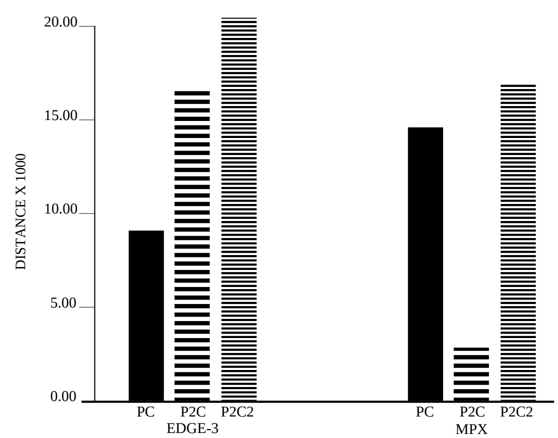

Figure 5. Avg. distance due to foreign edges.

The insertion rate of offspring into the initial GENITOR population is shown in Figure 4. The most surprising phenomenon is that, while Edge-3 and MPX-fixed operators have similar insertion rates when 1-Pass of 2-Opt was not applied, Edge-3 was almost never able to insert offspring produced when the population was initialized with 1-Pass of 2-Opt (i.e., **P2C**). It also seems to suggest that Edge-3 will outperform MPX-fixed in the GENITOR algorithm search when 1-Pass of 2-Opt is not applied and that the performance will be reversed when 1-Pass of 2-Opt is applied to the initial population.

Figure 5 shows the average distance incurred due to foreign edges introduced into the offspring during recombination. MPX-fixed offspring incur less overall distance than Edge-3 offspring when 1-Pass of 2-Opt is applied initially. This is certainly affected by the fewer number of failures observed but the average distance of *a single foreign edge* using Edge-3 recombination is 1008 while that distance using MPX-fixed is 412. This appears to be a result of the relative order method used by MPX to choose the next city in a tour after a failure. Consider the following donor tour segment where the boxes represent cities that have already been included in the offspring tour:

$$\square \quad \square \quad d \quad \square \quad \square \quad \square \quad f \quad \square \quad \square \quad h \quad g \quad \square \quad a \quad c \quad \square \quad \square$$

If a failure occurs after including city *f* in the offspring tour (i.e.; there is no city following city *f* in either the receiver or donor that has not already been chosen), then the MPX operator will search along the donor chromosome sequentially from the point of the last city inserted into the offspring tour (i.e., city *f*) looking for the next city that has not yet been used in the offspring (i.e., *h*). However, Edge-3 randomly picks the next city to continue the tour. Since the initial parent tours have been improved using 2-Opt, the distance between the cities in a sequential order is expected to be much less than the distance resulting from two cities placed in a random order. This suggests that Edge-3 could be improved.

The failure distance metric suggests that Edge-3 will outperform MPX-fixed if 2-Opt is not applied. However, it also suggests that MPX-fixed will outperform Edge-3 when 1-Pass of 2-Opt is used, due to the distance incurred due to failure.

3.2. 2-Repair and inheritance

The analysis of Edge-3 and MPX-fixed suggests that the use of some low cost operation to minimize the distances between segments of inherited edges (marked at failure points) might cause the two operators to behave similarly. 2-Repair is one such operator [2]. We have not yet studied the interaction of 2-Repair with the recombination operators. This needs to be done since 2-Repair will not interact with the operators in the same way as 2-Opt. In any practical application of these operators some "repair" operator would be advantageous since it does not affect edge inheritance or hyperplane sampling. Given the small number of foreign edges (e.g. 10 to 20) introduced for the 532-city problem during recombination, a 3-Repair operator using 3-Opt is also feasible.

3.3. Inheritance empirical performance

The following experiments were designed to compare the performance of the MPX-fixed and Edge-3 operators, as well as, to evaluate the metrics presented earlier. This work was not done to eclipse or duplicate other performance records on the Padberg 532-city problem. There were no attempts to tune or augment GENITOR for this problem. In the first experiment the Edge-2, Edge-3 and MPX-fixed operators were tested using a random initial population without any local optimization. Some interesting behavior is exhibited in Figure 6. As suggested by the failure rate metric, Edge-3 without 1-Pass of 2-Opt outperforms the other operators. The slower progress exhibited by MPX-fixed may be due to the it's higher failure rate. Failure rate appears to be the more significant of the metrics in this context, since the ρ_{op} metric is very close for both operators.

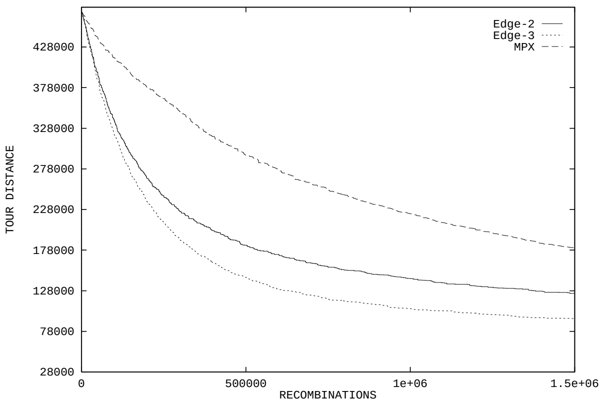

Figure 6. Operator performance comparisons.

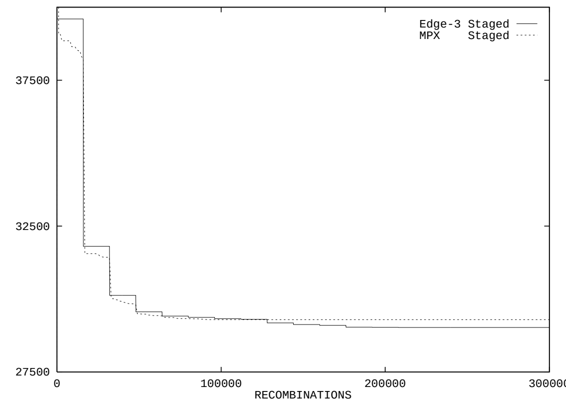

Figure 7. Performance of staged search.

The performance of the Edge-3 and MPX-fixed operators was compared when the initial population was improved using 1-Pass of 2-Opt. Edge-3, which had an *insertion rate* of zero in the earlier metrics, displayed trivial insertion rates over prolonged search. It was never able to improve on the best string contained in the initial enhanced population as predicted in Figure 4. MPX-fixed found some improvements and slightly outperformed Edge-3.

Another experiment was designed to test the effects of application of 2-Repair, as explained in section 3.2, to offspring produced from parents enhanced using 1-Pass of 2-Opt (**P2C**). Neither operator showed significant improvement over their performance when 2-Repair was not used.

An experiment was designed to use the 1-Pass of 2-Opt in such a way as to keep interference with edge inheritance and hyperplane sampling to a minimum by applying 1-Pass of 2-Opt at *discrete intervals*. A smaller population was used to reduce the amount of work. In this experiment, an initial population of 2000 strings was improved using 1-Pass of 2-Opt. Then genetic search was executed for 16,000 recombinations while applying 2-Repair to each offspring. 1-Pass of 2-Opt was then reapplied to the entire population and the cycle was repeated. We refer to this as a *Staged* search. The Staged experiments shown in Figure 7 resulted in an average solution of 28,979 with a best solution of 28,752 for the Edge-3 recombination operator. The Staged experiments resulted in an average solution of 29,294 with a best solution of 29,171 for the MPX-fixed recombination operator. The optimal tour has a distance of 27,686. Eshelman [1] and Gorges-Schleuter [2] have reported better results, but these experiments show the value of the defined metrics in comparing the recombination operators.

4. CONCLUSIONS

The goal this paper has been to evaluate issues that impact the use of genetic algorithms for solving TSPs. Alphabet cardinality, the effectiveness of recombination operators at preventing foreign edges, and the interaction of recombination operators with 2-Opt have all been explored.

The current study suggests that MPX and Edge-3 recombination have different behaviors; each has advantages and it seems reasonable that some hybrid of these two operators might yield better performance than either by itself. Each individual metric provides useful information about certain aspects of the recombination operators. However, no single metric explored here seems to predict operator performance. When considered together however, the metrics provide analytical information which may prove helpful in constructing a superior hybrid operator. It also seems that those operator characteristics that appear to be the most desirable at the beginning of the search are not necessarily the best throughout the remainder of the search.

This paper has not attempted to address issues related to improvements in the genetic algorithm itself. However, improvements in the genetic algorithm, efforts to better deal with the alphabet cardinality issue as well as improved recombination operators could all contribute to the solution of larger TSPs by genetic search.

5. ACKNOWLEDGEMENTS

This research was supported in part by NSF grant IRI-9010546. Also, special thanks to E. Pesch and L. Eshelman for providing guidance, code and useful discussion.

6. REFERENCES

1 L. Eshelman. (1991) "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination." In *Foundations of Genetic Algorithms.* Morgan Kaufmann.

2 M. Gorges-Schleuter. (1989) "ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy." *Proc. Third Int. Conf. on Genetic Algorithms.* Morgan Kaufman.

3 S. Lin, and B. Kernighan (1973) "An Efficient Heuristic Procedure for the Traveling Salesman Problem." In *Operations Research.*, 21:498-516.

4 B. Manderick, M. de Weger, and P. Spiessens. (1991) "The Genetic Algorithm and the Structure of the Fitness Landscape." In *Proc. Fourth Int. Conf. on Genetic Algorithms.* Morgan Kauffman.

5 H. Mühlenbein. (1991) "Evolution in Time and Space - The Parallel Genetic Algorithm" In *Foundations of Genetic Algorithms.* Morgan Kaufmann.

6 W. Padberg and G. Rinaldi. (1987) "Optimization of a 532-City Symmetric TSP". In *Operations Research Letters.*, 6(1):1-7

7 N. Ulder, E. Aarts, H. Bandelt, P. Laarhoven, E. Pesch. (1990) "Genetic Local Search Algorithms for the Traveling Salesman Problem." In *Parallel Problem Solving In Nature.* Springer/Verlag.

8 T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley. (1991) "A Comparison of Genetic Sequencing Operators." In *Proc. Fourth Int. Conf. on Genetic Algorithms.* Morgan Kauffman.

9 D. Whitley, T. Starkweather, and D. Fuquay. (1989) "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator." In *Proc. Third Int.. Conf. on Genetic Algorithms.* Morgan Kaufmann.

10 D. Whitley, T. Starkweather, and D. Shaner. (1990) "Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination." In *Handbook of Genetic Algorithms.* Van Nostrand.