A Comparison of Memetic Recombination Operators for the Traveling Salesman Problem

Peter Merz
University of Tübingen
Department of Computer Science - WSI-RA
Sand 1, D-72076 Tübingen, Germany
peter.merz@ieee.org

Abstract

Several memetic algorithms (MAs) – evolutionary algorithms incorporating local search – have been proposed for the traveling salesman problem (TSP). Much effort has been spent to develop recombination operators for MAs which aim to exploit problem characteristics to achieve a highly effective search.

In this paper, several recombination operators for the TSP are compared. For the purpose of identifying the important properties of a recombination operator, a new generic recombination operator (GX) is proposed which is comprised of four phases. These phases can be controlled by parameters reflecting the most important properties of recombination operators. It is shown that GX recombination is superior to MPX and DPX when all common edges are preserved in the offspring.

1 INTRODUCTION

The traveling salesman problem (TSP) is one of the best-known combinatorial optimization problems. It can be stated as follows: Given n cities and the geographical distance between all pairs of these cities, the task is to find the shortest closed tour in which each city is visited exactly once. From a graph theoretical point of view, this is equivalent to finding the shortest Hamiltonian cycle in a complete graph.

The TSP has been widely used as a problem for testing new heuristic algorithms and general purpose optimization techniques. In particular, several evolutionary algorithms have been proposed to tackle this $\mathcal{NP}$-hard problem. Simple evolutionary algorithms have been shown to be ineffective in finding near optimum solutions [9]. Therefore, several researchers incorporated local search into an evolutionary framework such that all individuals in the population are local optima, leading to highly effective algorithms, known as genetic local search or memetic algorithms (MAs)[23, 24]. Memetic algorithms have been shown to be among the best heuristics for the TSP [18, 13, 17, 12, 10, 1]. Other important aspects of MAs not covered in this paper include spatial population structures [7], MA theory [29], and self-adaptation [14].

In this paper, memetic algorithms for the TSP are studied by concentrating on the most important part of the evolutionary framework – the recombination of solutions. A new generic greedy recombination operator is introduced to the study three important aspects of TSP tour recombination: the inheritance of common edges to both parents, the insertion of new edges, and the inheritance of edges found in just one of the parents. The greedy operator can be controlled by three parameters in respect to these aspects. Several parameter settings of GX are compared with maximally preserving crossover (MPX) [25, 8] and distance preserving crossover (DPX) [6, 18]. The experiments show the importance of inheriting common edges and provide a meaningful choice of the remaining two parameters. The effectiveness of the MA with GX is demonstrated on several instances form TSPLIB [28].

The paper is organized as follows. In section 2, the memetic algorithm framework used in this paper, as well as the new greedy recombination operator is introduced. In section 3, a comparison of memetic recombination operators is performed on selected TSP instances. And results are presented of the MA using GX on 15 TSP instances. Section 4 concludes the paper and outlines areas for future research.

2 MEMETIC ALGORITHMS FOR THE TSP

Although there are other effective evolutionary algorithms incorporating sophisticated problem dependent procedures such as Nagata and Kobayashi's [26] evolutionary algorithm with *edge assembly crossover*, MAs provide a general framework for hybrid algorithms that can be applied to other combinatorial problems such as the quadratic assignment problem [20], the binary quadratic programming problem [19], and graph bipartitioning [21].

```
procedure MA;
  begin
    initialize population P;
    foreach i ∈ P do i := Local-Search(i);
    repeat
      for i := 1 to #recombinations do
        select two parents i_a, i_b ∈ P randomly;
        i_c := Recombine(i_a, i_b);
        i_c := Local-Search(i_c);
        add individual i_c to P;
      endfor;
      for i := 1 to #mutations do
        select parent i_a ∈ P randomly;
        i_c := Mutate(i_a);
        i_c := Local-Search(i_c);
        add individual i_c to P;
      endfor;
      P := select(P);
      if P converged then
        foreach i ∈ P\{best} do
          i := Local-Search(Mutate(i));
      endif
    until terminate=true;
  end;
```

Figure 1: The Memetic Algorithm

MAs for the TSP are similar to other evolutionary algorithms: a population of locally optimal solutions is evolved over time by applying evolutionary variation operators (mutation and recombination operators), and by selection of the best individuals from the pool of parents and offspring. The pseudo code for the MAs used in this contribution is shown in Fig. 1. To ensure that the individuals in the population are local optima, after each application an evolutionary variation operator, local search is applied. This includes the initialization phase of the population in which solutions are constructed from scratch: A local search procedure is applied to these solutions so that even the first generation consists exclusively of local optima.

The problem-specific parts of the algorithm comprise initialization, local search, and the evolutionary variation operators: recombination and mutation. In comparison to other evolutionary algorithms, the role of mutation and recombination is different. Firstly, mutation and recombination are performed independently from each other. Secondly, the phenotypic changes caused by the variation operators must be large enough to reach the basin of attraction of new local optima, since local search is always applied after mutation or recombination.

2.1 Initialization and Local Search

To initialize the population of the MA, TSP tours have to be generated either randomly or by a randomized tour construction heuristic such as nearest neighbor or the greedy heuristic [11]. After the generation of feasible tours, a local search is applied.

The most effective local search procedures for the TSP are *2-opt*, *3-opt*, and the Lin-Kernighan (LK) heuristic [15, 11]. These heuristics exchange 2, 3, or a variable number of edges in each iteration, respectively. Generally, the stronger the local search used the better the performance of the MA. Therefore, the Lin-Kernighan heuristic has been used in [6, 5, 18].

2.2 Recombination Operators

During recombination, a new offspring is generated by copying edges from the parents. However, the TSP tour constraints have to be obeyed: each node (city) is connected with exactly two other nodes via two edges and the tour is required to have only one cycle. These constraints are hard to obey, hence many proposed recombination operators introduce foreign edges which are not contained in one of the parents to meet the constraints. These foreign edges can be considered as *implicit mutations*, and have a high impact on the performance of EAs for the TSP [16], since they can be very long, destroying the benefit of combining the short edges from the parents.

2.2.1 Properties of Recombination Operators

The use of local search after the application of a recombination operator – as is the case in memetic algorithms – can compensate for the disruptive effects of implicit mutations. In some cases, implicit mutations have a positive effect on the performance of the local search, and in some situations they have not. Thus, it is important that implicit mutations can be controlled in some way. Besides the number of foreign edges introduced during recombination, another aspect appears to be important: which edges are inherited from the parents and which are not. More formally, recombination operators can be classified according to Radcliffe and Surry [27] as

Respectful: The alleles that are identical in both parents are preserved in the offspring, i.e. all edges found in both parent tours (common edges) are found in the offspring tour

Assorting: The offspring contain only alleles from either one of the parents, i.e. all edges in the child tour are found in at least one of the parent tours, thus no implicit mutation occurs

While respectful recombination can be easily achieved by a recombination operator for the TSP, assorting recombination is hardly accomplished. Note, that for binary representations a respectful recombination is also assorting.

2.2.2 MPX and DPX

Although there are many recombination operators proposed for the TSP, we concentrate on those especially useful in combination with local search und thus in a memetic framework. Other recombination operators such as the *edge recombination operator family* [16, 3] or the *edge assembly crossover* [26] are aimed at preserving edges without additional local search. These operators are inferior to other more disruptive operators if local search is used [3, 17].

In the MPX proposed in [8] a sub-path between two randomly chosen crossover points is copied from the first parent to the offspring. The first crossover point is chosen to be at an edge not contained in the second parent. The partial tour is extended by copying edges from the second or first parent afterwards. If no parental edge can be included a foreign edge is introduced to maintain feasibility. To a high extent, edges from the parents are retained. This operator does not guarantee to be respectful.

The DPX proposed in [6, 5] is an operator that is only useful in combination with local search. In contrast to MPX or other recombination operators such as the edge recombination operators [30], it forces the inclusion of foreign edges in the offspring instead of preventing it.

DPX tries to generate an offspring that has equal distance to both of its parents, i.e., its aim is to achieve that the three distances between offspring and parent 1, offspring and parent 2, and parent 1 and parent 2 are identical. It works in two phases: (1) all common edges are copied to the offspring, and (2) the tour fragments present in the offspring are reconnected based on a nearest neighbor algorithm where edges contained in one of the parents are not considered.

2.2.3 The Generic Greedy Recombination Operator

A new recombination operator is proposed in the following that utilizes the greedy construction scheme of the greedy heuristic [11]. The *generic greedy recombination operator (GX)* consists of four phases:

Phase I: (*common edges*)
In the first phase, some or all edges contained in both parents are copied to the offspring tour.

Phase II: (*new edges*)
In the second phase, new short edges are added to the offspring that are not contained in one of the parents. These edges are selected randomly among the shortest edges emanating from each node. These edges are with high probability contained in (near) optimum solutions and are thus good candidates for edges in improved tours.

Phase III: (*non-common edges*)
In a third phase, edges are copied from the parents by making greedy choices. Edges are inserted in order of increasing length, and only candidate edges are considered, i.e., edges that violate the TSP constraints.

Phase IV: (*remaining edges*)
In the fourth and last phase, further edges are included in order of increasing length until the child consists of n edges and is thus a feasible TSP tour.

All greedy choices in the fourth step are randomized by selected the shortest remaining edge with a probability of 0.66 and the second shortest edge with a probability of 0.33.

The GX operator has three parameters: the common edges inheritance rate (cRate) that determines the probability that a common edge is added to the child and is thus a control parameter for the first phase. With a rate of 1.0, respectful recombination is achieved, all other rates lead to disrespectful recombination. The second phase is controlled by the new edges insertion rate (nRate) that determines the number of new edges to include. A rate of 0.5, for example, determines that half of the remaining edges to insert after phase one are new edges that are short but not contained in one of the parent solutions. The maximum number of edges to inherit from the parents is determined by the inheritance rate (iRate). In the last phase, allowed edges in increasing length are chosen that may or may not be found in the parents. For am more detailed explanation see [17].

2.3 The Mutation Operator

Simple mutation operators are not suited for use in MAs, since subsequently applied local search procedures will usually revert the changes made. For example, the inversion operator randomly exchanging two edges is ineffective when *2-opt*, *3-opt* or LK local search is used. Therefore, in MAs alternative mutation operators are required.

The non-sequential four change (NS4) is an edge exchange involving four edges [15]. It is especially useful in connection with the LK heuristic. Since LK only performs sequential exchanges, it cannot reverse a non-sequential four change in one iteration. The NS4 is used in the iterated Lin-Kernighan heuristic [11], which is known to be very effective.

2.4 Selection and Restarts

In this work, a single panmictic population structure is used. Thus selection utilized in the memetic algorithms is a global selection strategy and similar to the selection in the $(\mu + \lambda)$-ES (*Evolution Strategy*): The new population is derived by selecting the best individuals out of the pool of parents and children. Duplicates are eliminated such that a solution is contained no more than once in the population.

Due to small population sizes and the use of local search in memetic algorithms, the problem of premature convergence arises. Therefore, the restart technique proposed by Eshelman [4] is employed. During the run, it is checked whether the search has converged. If so, the whole population is mutated except for the best individual. The mutation used here exchanges k edges with k being high compared to the mutation operator described above.

3 EXPERIMENTAL RESULTS

Several experiments have been conducted to evaluate the performance of MAs for the TSP. All experiments described in the following were conducted on a PC with Pentium III Processor (500 MHz) under Linux 2.2. All algorithms were implemented in *C++*. For details of the algorithms see [17].

3.1 Comparison of Recombination Operators

In a first set of experiments, several recombination operators for the TSP were tested under the same conditions on three selected TSP instances contained in TSPLIB: att532, pr1002, and fl1577. To get a clear picture of the operator effectiveness, no additional mutation was performed and the restart mechanism was disabled during the runs. Furthermore, a fast *2-opt* local search was used in the MAs that is not as effective as *3-opt* local search or the Lin-Kernighan heuristic to reduce the strong influence of the (sophisticated) local search. The recombination operators MPX, DPX, and the generic greedy recombination operator were studied with various parameter settings. The population was set to $P = 100$ in all runs, and the variation operator application rate was set to 0.5, i.e., 50 offspring were generated per generation. The results of the experiments are summarized in Table 1. For each instance/operator, the average number of generations, the shortest tour length found, and the percentage excess over the optimum solution value is provided. For the GX operator, the values for *cRate*, *nRate* and *iRate* are provided in the form *cRate/nRate/iRate*. For example, a parameter setting of 1/0.25/0.75 means that the common inheritance rate *cRate* was set to 1.0, the new edges insertion rate *nRate* was set to 0.25, and the inheritance rate *iRate* was set to 0.75. The dot in each column block indicates the best result within this block.

For all three instances, MPX and DPX are outperformed by GX for some of the parameter settings: all GX variants with a common inheritance rate of 1.0 and a new edge introduction rate of 0.25 perform better than MPX and DPX. However, the best parameter setting for GX is for each of the instances a different one implying that there is no "golden rule" leading to the best recombination strategy for all TSP instances! For example, the best setting for fl1577 is 1/0/0.75 but all other combinations with *nRate* set to 0.0 do not perform as well as the GX variants with *nRate* set to 0.25. Furthermore, it becomes apparent that respectfulness is a very important property of recombination operators since all GX versions with a common inheritance rate less than 1 perform significantly worse than the respectful greedy recombination operators. However, choosing a high inheritance rate can compensate the phenomenon to an extent since the common edges of the parents have a chance to be included in the offspring in the third phase of the generic recombination. Additionally, iterated *2-opt* local search (ILS) and a MA with the non-sequential four-change mutation (NS4) and no recombination has been applied to the three instances. The mutation based al-

Table 1: Comparison of MA Recombination Strategies for the TSP (2-opt)

Operator	att532		pr1002		fl1577	
DPX	1565	27793.0 - 0.386%	664	266240.5 - 2.778%	653	22314.0 - 0.292%
MPX	2691	27772.0 - 0.311%	3404	261695.5 - 1.023%	1240	22347.8 - 0.444%
GX-Params						
1/1/1	650	27738.7 - 0.190%	307	268183.5 - 3.528%	554	22295.6 - 0.210%
1/1/0.75	708	27744.7 - 0.212%	354	268072.9 - 3.485%	592	22306.7 - 0.259%
1/1/0.5	725	27740.0 - 0.195%	415	267033.1 - 3.084%	585	22304.0 - 0.247%
1/1/0.25	669	27772.0 - 0.311%	304	268487.4 - 3.645%	580	22296.5 - 0.213%
1/0.5/1	868	27729.8 - 0.158%	759	260907.8 - 0.719%	624	22294.8 - 0.206%
1/0.5/0.75	929	27727.0 - 0.148%	733	261981.0 - 1.133%	713	22294.6 - 0.205%
1/0.5/0.5	923	27725.2 - 0.142%	808	261121.2 - 0.801%	682	22296.7 - 0.214%
1/0.5/0.25	892	27723.9 - 0.137%	832	260723.4 - 0.648%	641	22303.5 - 0.245%
1/0.25/0	928	27724.5 - 0.139%	1223	260671.2 - 0.628%	690	22304.5 - 0.250%
1/0.25/0.75	1091	• 27719.2 - 0.120%	1430	260683.9 - 0.633%	769	22294.8 - 0.206%
1/0.25/0.5	1065	27722.4 - 0.131%	1422	260585.9 - 0.595%	684	22311.7 - 0.282%
1/0.25/0.25	998	27723.3 - 0.135%	1334	• 260508.6 - 0.565%	696	22307.0 - 0.261%
1/0/1	956	27763.5 - 0.280%	1321	261379.9 - 0.901%	736	22323.4 - 0.335%
1/0/0.75	1071	27728.0 - 0.152%	1481	260894.8 - 0.714%	735	• 22287.8 - 0.174%
1/0/0.5	1035	27725.4 - 0.142%	1434	260949.5 - 0.735%	744	22312.0 - 0.283%
1/0/0.25	1006	27737.7 - 0.186%	1412	260984.0 - 0.749%	719	22326.2 - 0.347%
0.75/0.5/1	201	28429.8 - 2.686%	226	269423.5 - 4.007%	212	22725.8 - 2.143%
0.75/0.5/0.75	224	28435.5 - 2.707%	254	269423.5 - 4.007%	230	22725.8 - 2.143%
0.75/0.5/0.5	215	28435.5 - 2.707%	243	269423.5 - 4.007%	225	22725.8 - 2.143%
0.75/0.5/0.25	206	28434.8 - 2.705%	232	269423.5 - 4.007%	219	22725.8 - 2.143%
0.75/0.25/0	233	27986.0 - 1.084%	229	269271.2 - 3.948%	227	22679.0 - 1.932%
0.75/0.25/0.75	269	28230.8 - 1.968%	288	269423.5 - 4.007%	269	22671.2 - 1.897%
0.75/0.25/0.5	254	28063.3 - 1.363%	258	269335.2 - 3.972%	254	22657.9 - 1.838%
0.75/0.25/0.25	243	27976.5 - 1.049%	240	269384.7 - 3.991%	239	22649.5 - 1.800%
0.75/0/1	407	27869.0 - 0.661%	422	263536.0 - 1.734%	270	22583.3 - 1.503%
0.75/0/0.75	517	27771.5 - 0.309%	705	• 261696.8 - 1.024%	620	• 22319.3 - 0.316%
0.75/0/0.5	457	• 27747.2 - 0.221%	558	262236.0 - 1.232%	398	22415.2 - 0.747%
0.75/0/0.25	415	27750.5 - 0.233%	435	262634.5 - 1.386%	298	22492.2 - 1.093%
0.5/0.25/0	156	28394.2 - 2.558%	179	269400.0 - 3.998%	161	22725.8 - 2.143%
0.5/0.25/0.75	191	28433.2 - 2.699%	224	269423.5 - 4.007%	187	22725.8 - 2.143%
0.5/0.25/0.5	172	28414.0 - 2.630%	201	269423.5 - 4.007%	178	22724.8 - 2.139%
0.5/0.25/0.25	162	28373.5 - 2.483%	187	269423.5 - 4.007%	170	22725.8 - 2.143%
0.5/0/1	195	28041.8 - 1.285%	216	266696.7 - 2.954%	174	22693.8 - 1.999%
0.5/0/0.75	403	27870.7 - 0.667%	455	• 263020.8 - 1.535%	363	• 22416.0 - 0.751%
0.5/0/0.5	293	• 27838.5 - 0.551%	316	263258.8 - 1.627%	242	22530.1 - 1.263%
0.5/0/0.25	220	27894.7 - 0.754%	227	265673.8 - 2.559%	192	22628.6 - 1.706%
ILS	61365	27777.7 - 0.331%	126457	260683.6 - 0.633%	150797	22369.2 - 0.540%
NS4	744	27860.2 - 0.629%	1438	261922.0 - 1.111%	1633	22304.0 - 0.247%
Time:		60 sec.		120 sec.		200 sec.

gorithms perform relatively well but can not compete with the greedy recombination MAs. For the instance fl1577, the MA with NS4 performs much better than ILS indicating that for this type of landscape search from multiple points (population-based search) is more promising.

In the second experiment, we replaced the fast *2-opt* local search with the Lin-Kernighan heuristic. The population size was set to 40, the variation operator application rate was set to 0.5, i.e., 20 offspring were generated per generation, and restarts were enabled with a diversification rate of 0.3 ($0.3 \times n$ edges were randomly exchanged with n denoting the number of cities). The results obtained from experiments with MAs using DPX, MPX, respectful GX, non-sequential-four-change mutation (denoted NS4) in comparison to the iterated Lin-Kernighan heuristic (ILK) are displayed in Table 2. For each instance/operator pair, the average number of generations, and the percentage excess over the optimum solution value is provided. For the GX operator, the values for *nRate* and *iRate* are provided in the form *nRate/iRate*. *cRate* was set to 1.0 in all experiments. The dot in each row indicates the best result for an instance.

Table 2: Comparison of MA Recombination Strategies for the TSP (LK)

Operator	att532	rat783	pr1002	fl1577	pr2392	pcb3038
DPX	0.030 %	0.004 %	0.023 %	• 0.028 %	0.068 %	0.113 %
MPX	• 0.021 %	• 0.001 %	0.169 %	0.142 %	0.054 %	0.128 %
GX 1.0/1.0	0.030 %	0.007 %	0.036 %	0.055 %	0.042 %	0.132 %
GX 1.0/0.75	0.035 %	0.026 %	0.022 %	0.058 %	0.053 %	0.211 %
GX 1.0/0.5	0.040 %	0.008 %	0.011 %	0.045 %	0.050 %	0.171 %
GX 1.0/0.25	0.043 %	0.006 %	0.013 %	0.051 %	0.047 %	0.146 %
GX 0.5/0.5	0.033 %	0.006 %	0.009 %	0.042 %	0.037 %	0.112 %
GX 0.5/0.75	0.031 %	0.007 %	0.031 %	0.048 %	0.055 %	0.175 %
GX 0.5/0.5	0.035 %	0.008 %	0.005 %	0.046 %	0.051 %	0.143 %
GX 0.5/0.25	0.037 %	0.009 %	0.011 %	0.037 %	0.044 %	0.136 %
GX 0.25/0	0.026 %	0.002 %	0.017 %	0.044 %	0.022 %	0.125 %
GX 0.25/0.75	0.038 %	0.012 %	0.003 %	0.041 %	0.031 %	0.151 %
GX 0.25/0.5	0.035 %	0.006 %	0.002 %	0.036 %	0.025 %	0.111 %
GX 0.25/0.25	0.041 %	0.005 %	0.002 %	0.040 %	0.023 %	• 0.111 %
GX 0.0/1.0	0.045 %	0.008 %	0.006 %	0.052 %	• 0.020 %	0.123 %
GX 0.0/0.75	0.036 %	0.003 %	• 0.000 %	0.043 %	0.027 %	0.115 %
GX 0.0/0.5	0.034 %	0.011 %	0.008 %	0.052 %	0.029 %	0.122 %
GX 0.0/0.25	0.037 %	0.004 %	0.002 %	0.050 %	0.035 %	0.123 %
ILK	0.046 %	0.018 %	0.065 %	0.158 %	0.215 %	0.135 %
NS4	0.055 %	0.010 %	0.020 %	0.181 %	0.119 %	0.171 %
Time:	60 sec.	80 sec.	200 sec.	300 sec.	400 sec.	800 sec.

Here, the performance differences of the MAs are in most cases not significant. For the problems att532, rat783, and pr1002 all algorithms perform well with only small differences, except for the MA with MPX recombination in case of pr1002. Surprisingly, this MA performs significantly worse than the other algorithms. For fl1577, the MAs with DPX and GX outperform all other competitors, with the MA using DPX being the best. For pr2392, all recombination based algorithms perform similarly, but the MAs with mutation and ILK perform significantly worse. In case of pcb3038, the largest instance considered, all results lie close together. The MAs with DPX and MPX outperform ILK and the MA with NS4. In the greedy recombination MAs, high differences can be observed. The best results are obtained with a new edge insertion rate of 0.25. The results show no clear tendency, and often the values lie too close together to be significantly different. However, in none of the cases, ILK or the MA with mutation is able to outperform the MA using DPX or the best greedy recombination. The performance differences between mutation and recombination operators have become more apparent using *2-opt* local search. For larger instances, this may be also observed for MAs with the LK heuristic.

3.2 DPX vs. GX Recombination

Using a NS4 mutation application rate of $m = 0.1$, the MAs have been run on a variety of problem instances contained in TSPLIB, to show the robustness and scalability of the memetic approach. In Table 3, the results are shown for five instances up to a problem size of 1002. The population size was set to $P = 40$ in all runs, the recombination application rate was set to 0.5, and the diversification rate to 0.1. Two MAs were run on each instance, the first one with DPX recombination and the second one with GX recombination. In the latter, *cRate* was set to 1.0, *nRate* was set to 0.1 which appears to be a good compromise between 0.25 and 0.0, and *iRate* was set to 0.5. The programs were

Table 3: Average Running Times of two MAs to find the Optimum

Instance	**Op**	**gen**	**quality**	N_{opt}	**t in s**
lin318	DPX	19	42029	30/30	8
	GX	13	0.00%	30/30	8
pcb442	DPX	824	50778	30/30	147
	GX	286	0.00%	30/30	68
att532	DPX	560	27686	30/30	127
	GX	289	0.00%	30/30	106
rat783	DPX	122	8806	30/30	26
	GX	136	0.00%	30/30	35
pr1002	DPX	333	259045	30/30	112
	GX	182	0.00%	30/30	98

terminated as soon as they reached an optimum solution. In the table, the average number of generations (gen) and the average running time of the algorithms (t in s) in seconds is provided. In 30 out of 30 runs, the optimum could be found for all instances in less than two minutes. The average running time for rat783 is much lower than for `att532` due to the structure of the fitness landscapes (see [22] for details): In most cases, the MA with greedy recombination appears to be slightly superior to the MA with DPX.

Additional experiments have been performed on TSPLIB instances up to a problem size of 85900. Due to the limited number of pages in this contribution, the results are not displayed here. They can be found in [22].

4 Conclusions

In an extensive study, several recombination operators including a newly proposed generic greedy recombination operator (GX) are compared in a MA framework. The MAs show significant performance differences if a simple *fast 2-opt* local search is employed. For MAs with the sophisticated Lin-Kernighan local search, the results lie much closer together. The study has shown that respectfulness is the most important property of a recombination operator. Furthermore, we have shown that the MA with the newly proposed greedy recombination operator outperforms all its competitors: MAs with DPX or MPX recombination, MAs with non-sequential four change mutation, and iterated local search.

MAs with DPX and GX recombination and mutation have been applied to various instances contained in TSPLIB to show robustness and scalability of the approach. For problems with up to 1000 cities the optimum could be found in all runs in an average time of less than two minutes on a personal computer with 500 MHz.

References

[1] D. Bonachea, E. Ingerman, J. Levy, and S. McPeak, "An Improved Adaptive Multi-Start Approach to Finding Near-Optimal Solutions to the Euclidean TSP," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, (D. Whitley *et al.*, eds.), (Las Vegas, Nevada, USA), pp. 143–150, Morgan Kaufmann, 2000.

[2] D. Corne, M. Dorigo, and F. Glover, eds., *New Ideas in Optimization*. McGraw-Hill, London, 1999.

[3] J. Dzubera and D. Whitley, "Advanced Correlation Analysis of Operators for the Traveling Salesman Problem," in *Parallel Problem Solving from Nature - Proceedings of the third Workshop, PPSN III*, (H.-P. Schwefel and R. Männer, eds.), (Dortmund, Germany), pp. 68–77, Springer-Verlag, Berlin, Germany, 1994.

[4] L. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," in *Foundations of Genetic Algorithms*, (G. J. E. Rawlings, ed.), pp. 265–283, Morgan Kaufmann, 1991.

[5] B. Freisleben and P. Merz, "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, (T. Bäck, H. Kitano, and Z. Michalewicz, eds.), (Piscataway, NJ), pp. 616–621, IEEE Press, 1996.

[6] B. Freisleben and P. Merz, "New Genetic Local Search Operators for the Traveling Salesman Problem," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature - PPSN IV*, (H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, eds.), pp. 890–900, Springer, 1996.

[7] M. Gorges-Schleuter, "Explicit Parallelism of Genetic Algorithms through Population Structures," in *Parallel Problem Solving from Nature*, (H. Schwefel and R. Manner, eds.), pp. 150–159, Springer-Verlag, 1991.

[8] M. Gorges-Schleuter, "Asparagos96 and the Traveling Salesman Problem," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 171–174, IEEE Press, 1997.

[9] J. J. Grefenstette, "Incooperating Problem Specific Knowledge into Genetic Algorithms," in *Genetic Algorithms and Simulated Annealing*, (L. Davis, ed.), pp. 42–60, Morgan Kaufmann Publishers, 1987.

[10] W. W. Hsu and C.-C. Hsu, "The Spontaneous Evolution Genetic Algorithm for Solving the Traveling Salesman Problem," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, (L. Spector *et al.*, eds.), pp. 359–366, Morgan Kaufmann, 2001.

[11] D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem: A Case Study," in *Local Search in Combinatorial Optimization*, (E. H. L. Aarts and J. K. Lenstra, eds.), pp. 215–310, Wiley and Sons, New York, 1997.

[12] S. Jung and B.-R. Moon, "The Natural Crossover for the 2D Euclidean TSP," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, (D. Whitley *et al.*, eds.), pp. 1003–1010, Morgan Kaufmann, 10-12 July 2000.

[13] K. Katayama and H. Narihisa, "Iterated Local Search Approach using Genetic Transformation to the Traveling Salesman Problem," in *GECCO-1999: Proceedings of the Genetic and Evolutionary Computation Conference*, (W. Banzhaf *et al.*, ed.), pp. 321–328, Morgan Kauffman, 1999.

[14] N. Krasnogor and J. Smith, "A Memetic Algorithm With Self-Adaptive Local Search: TSP as a Case Study," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, (D. Whitley *et al.*, eds.), pp. 987–994, Morgan Kaufmann, 2000.

[15] S. Lin and B. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, vol. 21, pp. 498–516, 1973.

[16] K. Mathias and D. Whitley, "Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem," in *Parallel Problem Solving from Nature - Proceedings of 2nd Workshop, PPSN 2*, (R. Männer and B. Manderick, eds.), pp. 219–228, Elsevier Science Publishers, 1992.

[17] P. Merz, *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.

[18] P. Merz and B. Freisleben, "Genetic Local Search for the TSP: New Results," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, (T. Bäck, Z. Michalewicz, and X. Yao, eds.), (Piscataway, NJ), pp. 159–164, IEEE Press, 1997.

[19] P. Merz and B. Freisleben, "Genetic Algorithms for Binary Quadratic Programming," in *GECCO-1999: Proceedings of the Genetic and Evolutionary Computation Conference*, (W. Banzhaf *et al.*, eds.), pp. 417–424, Morgan Kauffman, 1999.

[20] P. Merz and B. Freisleben, "Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 337–352, 2000.

[21] P. Merz and B. Freisleben, "Fitness Landscapes, Memetic Algorithms and Greedy Operators for Graph Bi-Partitioning," *Evolutionary Computation*, vol. 8, no. 1, pp. 61–91, 2000.

[22] P. Merz and B. Freisleben, "Memetic Algorithms for the Traveling Salesman Problem," Tech. Rep., Department of Computer Science, University of Siegen, Germany, 2001. Accepted for publication in *Complex Systems*.

[23] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," Tech. Rep. C3P Report 826, Caltech Concurrent Computation Program, California Institue of Technology, 1989.

[24] P. Moscato, "Memetic Algorithms: A Short Introduction," in *New Ideas in Optimization*, (D. Corne, M. Dorigo, and F. Glover, eds.), ch. 14, pp. 219–234, McGraw-Hill, London, 1999.

[25] H. Mühlenbein, "Evolution in Time and Space – The Parallel Genetic Algorithm," in *Foundations of Genetic Algorithms*, (G. J. E. Rawlins, ed.), Morgan Kaufmann Publishers, 1991.

[26] Y. Nagata and S. Kobayashi, "Edge Assembly Crossover: A High–power Genetic Algorithm for the Traveling Salesman Problem," in *Proceedings of the 7th International Conference on Genetic Algorithms*, (T. Bäck, ed.), pp. 450–457, Morgan Kaufmann, 1997.

[27] N. Radcliffe and P. Surry, "Fitness Variance of Formae and Performance Prediction," in *Proceedings of the Third Workshop on Foundations of Genetic Algorithms*, (L. Whitley and M. Vose, eds.), (San Francisco), pp. 51–72, Morgan Kaufmann, 1994.

[28] G. Reinelt, "TSPLIB— A Traveling Salesman Problem Library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

[29] A. Sinha and D. E. Goldberg, "Verificiation and Extension of the Theory of Global-Local Hybrids," in *Proceedings of the Genetic and Evolutionary Computation Conference*, (L. Spector *et al.*, ed.), pp. 592–598, Morgan Kaufmann, 2001.

[30] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley, "A Comparison of Genetic Sequencing Operators," in *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 69–76, Morgan Kaufmann, 1991.