Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 151 (2003) 352–364

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Genetic local search with distance preserving recombination operator for a vehicle routing problem

Andrzej Jaszkiewicz *, Paweł Kominek

Institute of Computing Science, Poznań University of Technology, ul. Piotrowo 3a, 60-965 Poznań, Poland

Abstract

The paper describes a systematic adaptation of the genetic local search algorithm to a real life vehicle routing problem. The proposition is motivated by successful implementations of genetic local search-based heuristics for a number of combinatorial optimization problems. The key element of the proposed approach is the use of global convexity tests. The tests allow finding the types of solution features that are essential for solution quality. The results of the tests are used to construct an appropriate distance preserving recombination operator. Results of computational experiments demonstrating the efficiency of the proposed approach are reported.

Keywords: Local search; Combinatorial optimization; Population-based metaheuristics; Vehicle routing problem

1. Introduction

Metaheuristic algorithms, e.g., genetic algorithms, simulated annealing or tabu search, have been successfully applied to many difficult optimization problems. The reader of many publications on metaheuristics may get an impression that the methods can solve any optimization problem efficiently.

However, the "No Free Lunch" (NFL) theorem (Wolpert and Macready, 1997) clearly states that there is no algorithm with a performance, averaged over all possible objective functions, that outperforms systematic enumeration. NFL is valid for any performance measure based on the objective function values of all solutions generated by the algorithm in a specified number of iterations. In particular, NFL is valid if the performance is evaluated with the objective function value of the best solution found by the algorithm. Thus, all optimization algorithms, including metaheuristics, are based on some, possibly implicit, assumptions; i.e. they are appropriate for some classes of problems. The NFL theorem is based on the assumption of a uniform probability distribution over the set of all possible objective functions. The probability distribution corresponding to the "real life" problems can be much different.

The success of metaheuristics in many applications proves that their assumptions are often met in practice.

Metaheuristics define general schemes for the optimization procedures that have to be adapted for specific problems. One direct conclusion of the

* Corresponding author.

E-mail addresses: jaszkiewicz@cs.put.poznan.pl (A. Jaszkiewicz), kominek@cs.put.poznan.pl (P. Kominek).

0377-2217/$ - see front matter
doi:10.1016/S0377-2217(02)00830-5

NFL theorem is that no such general scheme guarantees efficient optimization without appropriate adaptation. Thus, the way a given metaheuristic is adapted to a particular problem may have a crucial influence on its performance.

Genetic local search (GLS) is a metaheuristic algorithm that combines genetic (evolutionary) algorithms with local optimization. Other frequently used names are Memetic algorithms or Hybrid Genetic algorithms. Heuristics based on the GLS scheme often prove to be extremely efficient in combinatorial optimization (see e.g., Freisleben and Merz, 1996; Merz and Freisleben, 1997, 2000; Gorges-Schleuter, 1997; Galinier and Hao, 1999). It is quite difficult to track the single origin of GLS. To our knowledge, the first description of GLS was published by Ackley (1987), but similar algorithms were developed probably completely independently by several authors. Furthermore, there are significant similarities between GLS and scatter search (Glover, 1977; see also Glover, 1995, for discussion of the similarities).

Freisleben and Merz (1996) and Merz and Freisleben (1997) proposed a very efficient GLS-based heuristic for the traveling salesperson problem (TSP). The algorithm uses results of many years of studies on the TSP, e.g., it uses the nearest-neighbor heuristic (Lawler et al., 1985) to generate initial solutions, an efficient local search algorithm proposed by Lin and Kernighan (1973) and results of studies on global convexity of the TSP by Mühlenbein (1991) and Boese et al. (1994). An important component of the heuristic is the distance preserving recombination operator (Freisleben and Merz, 1996) motivated by global convexity of the TSP. An optimization problem has the property of global convexity if its good solutions have some significant similarities. Mühlenbein (1991) and Boese et al. (1994) noticed that good solutions (local optima) of the TSP instances have many common arcs. The distance preserving recombination operator places in the offspring solutions all the arcs common to both parents and completes the offspring with randomly selected arcs. In other words, this operator constructs an offspring with has the same features in common with its parents as the two parents have with each other.

Global convexity is not a unique feature of the TSP. According to the results of Jones and Forrest (1995), some numerical optimization problems also exhibit the property of global convexity. Taillard (1995) studied various classes of quadratic assignment problems using entropy. His results also indicate that some of the classes are globally convex. Merz and Freisleben (2000) also noticed global convexity of the quadratic assignment problems.

Proper definition of the recombination operator has a crucial influence on the performance of genetic (evolutionary) algorithms and genetic local search. The traditional approach to the development of recombination operators relies on intuition and extensive experiments with different operators. For example, Sevaux and Dauzère-Pérès (2003) test a number of different recombination operators for a scheduling problem and Michalewicz (1992) describes a number of recombination operators for the TSP.

In this paper, we propose a systematic approach for the construction of recombination operators appropriate for a given optimization problem and apply it to a real life vehicle routing problem. The approach follows the line of research that resulted in the efficient GLS-based heuristic by Freisleben and Merz (1996) and Merz and Freisleben (1997). We construct a distance preserving recombination operator based on global convexity tests. We propose to use the correlation between solution quality and distance in the decision space to find significant features of good solutions. Such features should be preserved by the recombination operator.

Jones and Forrest (1995) introduced fitness distance correlation analysis similar to the global convexity tests. Their goal, however, is to a study problem difficulty for genetic algorithms rather than finding appropriate adaptation. Furthermore, they study the difficulty of finding global optima while we are interested in the development of efficient heuristic algorithms.

Of course, the proposed systematic approach cannot overcome the limits imposed by the NFL theorem. Our goal is to propose an approach allowing the development of efficient heuristics for problems that:

- are globally convex, i.e., their good solutions are similar,
- allow efficient local optimization.

In Section 2, we describe the genetic local search algorithm. The real life vehicle routing problem to which we apply the approach is described in Section 3. In Section 4, we describe in detail the adaptation of the GLS algorithm to the vehicle routing problem. Computational experiments are described in Section 5. In Section 6, conclusions and directions for further research are presented.

2. Genetic local search algorithm

Local search can be combined in many ways with recombination operators. In GLS, each offspring resulting from recombination is a starting point for local optimization. Furthermore, the starting population is composed of solutions obtained by local optimization. Of course, local optimization can also be performed in many ways. For example, Radcliffe and Surry (1994) consider an algorithm in which a single iteration of local search is applied to each offspring. On the other hand, Taillard (1995) applies tabu search to each offspring. In the rest of the paper, we will assume, however, that the recombination operator is combined with a standard greedy or steepest local search, i.e., that local optimization always stops at a local optimum.

From the genetic (evolutionary) algorithm's perspective, GLS may be interpreted as a standard GA working on a set of local optima only. From this point of view, local search is just a part of the recombination operator. The efficiency of GLS may be explained by the fact that it works on a smaller search space than the standard GA. So, the best results should be achieved on problems where local optima constitute a relatively small part of the search space and the local optima can be generated in an efficient way.

GLS may be also interpreted as a modification of multiple start local search with random starting solutions. However, starting solutions are constructed in an intelligent way by combining properties of other good solutions. If the recombination operator is well designed, starting solutions obtained in this way should be better starting points for local search than random solutions. The efficiency of GLS in comparison to multiple start local search can be explained by the fact that local search, when started from a good starting solution, usually yields a better local optimum and requires less iterations to reach it.

In the rest of the paper, we will use the following version of the GLS algorithm:

Parameters: size of the current population—N, stopping criterion
Initialization:
Current population $P := \emptyset$
repeat N times
  Construct a new feasible solution $\mathbf{x}$ by a randomized algorithm
  Apply local search to $\mathbf{x}$ to obtain $\mathbf{x}'$
  Add $\mathbf{x}'$ to P
Main loop:
repeat
  Draw at random with uniform probability two solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ from P.
  Recombine $\mathbf{x}_1$ and $\mathbf{x}_2$ obtaining $\mathbf{x}_3$
  Apply local search to $\mathbf{x}_3$ to obtain $\mathbf{x}_3'$
  if $\mathbf{x}_3'$ is better than the worst solution in P and is different from all solutions in P **then**
    Add $\mathbf{x}_3'$ to P and delete the worst solution from P
until the stopping criterion is met

The above algorithm is not an orthodox version of a genetic algorithm. It assumes acceptance of every improvement, i.e., the algorithm has a reduced diversification factor with respect to more traditional algorithms using roulette wheel or tournament selection. Furthermore, the above algorithm does not explicitly introduce genetic generations; instead, it implements any new solution in the current population. However, the ideas proposed in the paper may be easily combined with other genetic schemes.

The above algorithm is just a general scheme for an optimization heuristic. In order to adapt it to a given problem the following issues should be addressed:

- the way the solutions are encoded,
- the algorithm for finding initial solutions,
- the local search algorithm,
- the recombination operator.

3. The vehicle routing problem

We apply the proposed approach to a real life vehicle routing problem. A characteristic feature of this problem, and of many other real life problems, is that although it is similar to classical problems known from the literature it has some special features making the existing algorithms inappropriate.

The problem concerns the operations of a waste management company in a city of about 600 000 inhabitants. The company removes about 60% of waste in the city. The waste is transported to two dumping sites with different costs. The company has about 30 000 waste containers in the city. They are emptied at almost 20 different frequencies. The highest frequency is five disposals per week; the lowest is equal to one disposal over 12 weeks. In addition individual, one-time orders should be considered. The containers are grouped in, so-called, sectors. Each sector contains a number of closely located containers. It is assumed that all containers in one sector are served by a single vehicle. About 100–200 sectors have to be visited during one day. In each sector, the vehicle spends time related to the capacity of the waste, type of containers and local conditions. The time does not depend on the type of vehicle. We do not consider the issue of sequencing containers within a sector, because local conditions usually precisely define the proper sequence of containers. We assume that the distances and travel times between each pair of sectors are known. The distances and travel times do not need to be symmetric. The company uses a non-homogeneous fleet of garbage trucks. Each of them can handle all types of containers. The vehicles differ by their capacity and cost per kilometer and hour. Usually about 30 vehicles are available each day. Each vehicle that operates in a given day starts its route at the company's base. Then it collects the waste in the city and goes to one of the dumping sites. It is assumed that a vehicle may serve a given sector only if it can collect all waste in the sector, i.e., each sector is visited by exactly one vehicle. The vehicle may visit a dumping site several times (usually twice) during the day. At the end of the day, it returns to the base. Vehicle return to the base must be empty.

There are some limitations on the working time but in practice, they are treated as soft constraints. The suggested working time is 8 hours but it is often slightly extended.

The decision consists in assigning a route to each vehicle through the sectors and dumping sites for the given day or in deciding that it does not operate on this day. Each vehicle route starts and ends in the company's base. The objective is to minimize the total operating cost of the waste removal, i.e., sum of costs related to the distance and working time of each vehicle that operates on the given day.

Potvin and Bengio (1996) proposed a genetic local search algorithm for a vehicle routing problem. As their problem definition differs from that considered in the paper, their results are not directly applicable in our case.

4. Adaptation of the genetic local search algorithm to the vehicle routing problem

4.1. Solution encoding

Traditional genetic algorithms use binary coding of solutions (Goldberg, 1988). For many problems, this kind of coding is not natural. Recently, solutions are often encoded with some specialized data structures. Algorithms using this form of encoding are sometimes called evolutionary algorithms (Michalewicz, 1992). We use the latter approach for the vehicle routing problem.

The solution encoding should not only be natural but it should also be efficient from the point of view of local search and recombination operator.

The solution to the vehicle routing problem is defined by a set of vehicle routes. The route of each vehicle is stored as a sequence of sector indices. Dumping sites are also treated as sectors of a special type. A part of the route that starts from the base or a dumping site and ends in a dumping

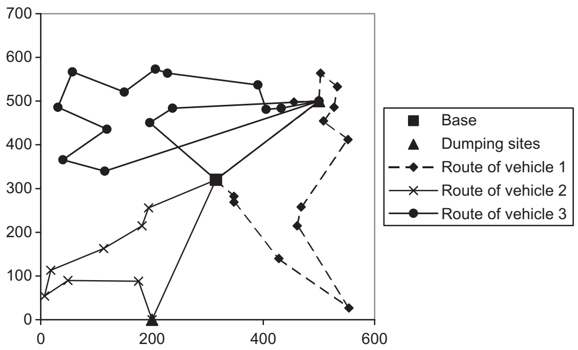

Fig. 1. Solution example.

site is called a *route segment*. The vehicle route may be empty. An example is presented in Fig. 1.

4.2. Finding initial solutions

Local search in the initialization phase may be started from randomly created solutions. The efficiency of this phase may be increased, however, if the quality of initial solutions is better than average. A simple specialized heuristic may be used to this end. GLS could be biased, however, if the method used to find initial solutions generate solutions that are too similar. This may result in premature convergence to a specific region of the decision space.

Obviously, the method used to find initial solutions should be randomized. Let assume that the method generates a solution of average quality f_0 higher than average quality over all feasible solutions. The method, when run a number of times should generate a representative sample of solutions of such quality. In practice, it may be difficult to guarantee this, but an effort should be made to avoid biases even at the cost of average quality.

In the case of the vehicle routing problem, we use the following algorithm for finding good initial solutions:

For each vehicle
    Create a vehicle route with a single default segment
For each sector s in random order
    For each vehicle route VR
        For each route segment RS in the route VR
            Evaluate insertion of sector s in the route segment RS
        Evaluate insertion of sector s in a newly created route segment
    Add sector s to the best possible position, i.e., position that results in the lowest increase to the total operating costs
Remove all vehicle routes composed of default segment only

The default segment starts at the base and ends at one of the dumping sites. The dumping site that results in the lowest cost for the vehicle is selected. Because the hour and kilometer costs are different for different vehicles and, in general, are not proportional, different dumping sites may be selected for different vehicles.

Whenever the insertion of a sector in a route segment is evaluated, the cost of the segment is optimized. This requires solving of a simple asymmetric TSP with usually up to 10 nodes. We use a greedy local search algorithm with the 2-opt neighborhood to this end. In addition, when the insertion of a sector in a route segment is evaluated, we test the possibility of changing the dumping sites at the beginning and at the end of the route segment. If the route segment starts at the base and ends at one of the dumping sites, the possibility of changing the ending dumping site is tested. If the route segment both starts and ends at a dumping site three additional possible configurations of starting and ending nodes are tested. The cost of the segment is optimized for each possible configuration of starting and ending nodes. If the route segment is not the first one, changing its starting dumping site requires that the ending dumping site of the previous segment is changed. In such case, the cost of the previous segment is also optimized. The same applies to the situation when the route segment is not the last one and its ending dumping site is changed.

The soft constraint related to the suggested working time corresponds to a penalty term added to the objective function. If the working time of a vehicle exceeds the suggested time, the cost of its route is increased.

4.3. Local search

We use local search with an operator that shifts a sector from a vehicle route to another route. The sector is inserted into the vehicle route in the best possible position found, in the same way as in the case of the initial solution.

Two versions of local search are considered. Steepest local search browses the whole neighborhood and accepts the best improving move in the neighborhood. In greedy local search, the neighborhood is browsed in random order and the first improving move is accepted. Greedy local search is applied to the initial solutions. Steepest local search is applied to the solutions obtained by recombination. This approach was found to give the best results. The same observation was previously made in the case of the TSP (Jaszkiewicz, 1998).

Note that the cost change corresponding to a sector shift is a sum of cost changes related to removal and insertion of the sector. Furthermore, each move changes two vehicle routes only, while the other routes remain unchanged. The two facts allow significant improvements in the efficiency of local search. At the beginning of local search two temporary cost tables are associated with each vehicle route. The cost tables store removal and insertion cost changes for each sector removed or inserted in the routes. Initially, all tables contain empty values. Whenever removal or insertion of a sector from a vehicle route is evaluated, the appropriate element in one of the tables is tested. If the cost is already known, i.e., if the appropriate element in the table is not empty, the value from the table is used. Otherwise the removal/insertion cost is evaluated and stored in the table. Furthermore, whenever a vehicle route is changed because of an accepted move all elements in the two tables associated with the route are filled with empty values. In the case of considered instances, this approach reduces the computation time of local search by a factor of 5.

4.4. Recombination operator

4.4.1. Global convexity tests

We formulate the hypothesis that the vehicle routing problem is globally convex, i.e., its good solutions are similar. The goal of the tests described in this section is to test this hypothesis and to find similarity measures concordant with this hypothesis.

Boese et al. (1994) in their experiments for the TSP use the number of common arcs as a similarity measure of two solutions. They report two indicators of the global convexity: the average number of common arcs between two local optima and the correlation between the quality of a local optimum and its average similarity to other local optima. The correlation is not computed numerically but presented in graphical form similar to that used in Fig. 2.

Assume that a set C of solutions is known. We propose to calculate the correlation between the value of the objective function f for solution $\mathbf{x}$ and the average similarity of $\mathbf{x}$ to other solutions not worse than $\mathbf{x}$ denoted by $\hat{s}(\mathbf{x})$. The average similarity is calculated as

$$\hat{s}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in C|f(\mathbf{y})\leqslant f(\mathbf{x})} s(\mathbf{x},\mathbf{y})}{|C|},$$

where $s(\mathbf{x},\mathbf{y})$ is similarity of $\mathbf{x}$ and $\mathbf{y}$ and objective function f is assumed to be minimized.

Four possible similarity measures were formulated and used in the global convexity tests. Each of them has some intuitive explanation.

- *The percentage of common arcs*. This measure is analogous to the one used in the case of the TSP. One could expect that short arcs and arcs directed towards dumping sites should often appear in good solutions.
- *The percentage of common assignments of sectors to vehicles*. One could expect that sectors lying far from both the base and the dumping sites should be served by vehicles with relatively low kilometer and hour costs. In addition, sectors with high loading times should be served by vehicles with low hour costs.
- *The percentage of common assignment of arcs to vehicles*. This measure is a combination of the previous two measures. It assumes that some arcs not only tend to appear often in good solutions but they also tend to be assigned to the same vehicles. One could expect that long arcs, e.g., arcs leading to remote sectors, should be

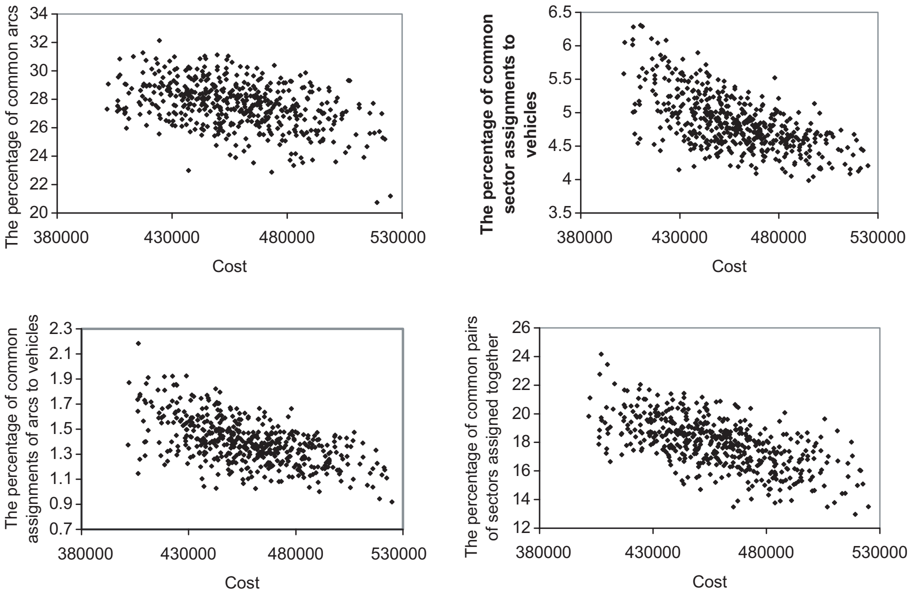

Fig. 2. Graphical presentation of correlations between solution quality (cost) and the similarity measures.

assigned to vehicles with relatively low kilometer and hour costs.

- *The percentage of common pairs of sectors assigned together to a single route segment*. The idea of the measure is very close to the approach used currently in the company. Consider a city neighborhood in which a number of sectors are to be visited. If the sectors are placed close to each other, it should be a good approach to serve them during a single trip of a vehicle, while the order in which the sectors are visited is of lower importance.

Note that our goal is to develop an efficient heuristic for the problem at hand. Usefulness of the above measures may depend on the values of some parameters, e.g., on tightness of capacity constraints.

In the experiments, we used 10 different instances of the problem, each corresponding to a different day of the company operations. In each of these instances, the same set of 31 vehicles was available. The instances differed by the sets of sectors to be served and the amounts of waste in the sectors. For each instance, 500 local optima were generated. The results presented in Table 1 are averages for the 10 instances. Fig. 2 presents a correlation example for one instance.

All the proposed similarity measures are correlated with solution quality. The highest correlations are obtained for the percentage of common assignment of sectors to vehicles and the percentage of common pairs of sectors assigned together to a single route segment. An interesting observation is that the percentage of common assignments of arcs to vehicles is significantly correlated with solution quality, however, the average number of common arcs assigned to the same vehicle is very low.

4.4.2. Distance preserving recombination operator

As all the similarity measures tested in Section 4.4.1 are correlated with solution quality, the distance preserving recombination operator should preserve features of all types. Below we propose a recombination operator that guarantees preservation of the features of two types and gives high probability for preserving the other features:

Recombination operator 1:

Parameters: two parent solutions $\mathbf{x}_1$ and $\mathbf{x}_2$

Phase 1:

For each vehicle v that has an assigned route in $\mathbf{x}_1$

Table 1
Results of the global convexity tests

	Similarity measure			
	The percentage of common arcs	The percentage of common assignments of sectors to vehicles	The percentage of common assignment of arcs to vehicles	The percentage of common pairs of sectors assigned together to a single route segment
Average value over all pairs of local optima [%]	27.68	4.87	1.40	18.13
Correlation with solution quality	−0.452	−0.581	−0.561	−0.611

With probability 0.5 place the whole route of vehicle v in $\mathbf{x}_1$ in the offspring solution
Phase 2:
For each vehicle v that has an assigned route in $\mathbf{x}_2$
  For each route segment *RS2* in the route of vehicle v in $\mathbf{x}_2$
    Remove from route segment *RS2* all sectors already assigned in phase 1
    If route segment *RS2* remains non-empty
      For each route segment *RS* in the route of vehicle v in the offspring solution
        Evaluate insertion of route segment *RS2* in route segment *RS*
      Evaluate insertion of route segment *RS2* in a new route segment
      Insert route segment *RS2* in the best possible position, i.e., the position that results in the smallest increase to the total operating costs

A route segment *RS2* considered in phase 2 may be empty if all its sectors are already assigned.

In phase 2, insertions of whole route segments *RS2* into existing route segments, are considered. The chain of regular sectors from *RS2* might be inserted at the beginning of the existing segment, or at the end of the existing segment, or between any two sectors from the existing segment. In neither case, the chain of sectors of *RS2* is broken nor their sequence is changed. Special sectors corresponding to dumping sites are not considered in this step. Node, however, that while inserting *RS2* an arc common to both parents may be broken. Thus, the operator does not guarantees preservation of common arcs and preservation of common assignments of arcs to vehicles. This may happen, however, only if breakage of a common arc results in the smallest increase to the total operating costs.

Theorem 1. *Each sector a assigned in both parents* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *to the same vehicle v will be assigned to the same vehicle.*

Proof. In phase 1, each route is assigned to the vehicle of parent 1, thus all its sectors are assigned to the same vehicle. In phase 2, each route segment is assigned to the vehicle of parent 2, thus all its, sectors not yet assigned in phase 1, are assigned to the same vehicle. □

Theorem 2. *Each pair of sectors assigned together in both parents* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *to a single route segment will be assigned together to a single route segment in offspring* $\mathbf{x}_3$.

Proof. Consider a pair of sectors a and b assigned together in both parents $\mathbf{x}_1$ and $\mathbf{x}_2$ to a single route segment. As complete vehicle routes are assigned in phase 1, either both of the sectors are assigned together in phase 1 or both of them are assigned together in phase 2. If the pair of segments is assigned in phase 1, then the route it belongs to in parent 1 is not changed. Thus, the pair of sectors remains in the same route segment.

If the pair of segments is assigned in phase 2, then the only change to the route segment it belongs to in parent 2 consists in removing sectors assigned in phase 1. Then the remaining route segment is inserted in the offspring as a whole. Thus, the pair of sectors remains in a single route segment. □

In order to test the importance of preservation of the common features, we have developed two other recombination operators that do not preserve some of the features. Both operators differ from operator 1 in phase 2 only. Recombination operator 2 does not preserve common pairs of sectors assigned together to a single route segment. In phase 2, sectors not yet assigned from each vehicle route of parent $\mathbf{x}_2$ are considered in random order and each of them is inserted separately in the offspring solution.

Recombination operator 2:

Parameters: two parent solutions $\mathbf{x}_1$ and $\mathbf{x}_2$
Phase 1:
For each vehicle v that has an assigned route in $\mathbf{x}_1$
  With probability 0.5 place the whole route of vehicle v in $\mathbf{x}_1$ in the offspring solution
Phase 2:
For each vehicle v that has an assigned route in $\mathbf{x}_2$
  For each sector s in the route of vehicle v in $\mathbf{x}_2$ (in random order)
    If the sector s has not already been assigned in phase 1
      For each route segment RS in the route of vehicle v in the offspring solution
    Evaluate insertion of sector s in route segment RS
      Evaluate insertion of sector s in a newly created segment
      Insert sector s in the best possible position, i.e., the position that results in the smallest increase to the total operating costs

Recombination operator 3 does not preserve common assignments of sectors to vehicles. In phase 2, the assignment of each route segment to each vehicle route is evaluated and the best position is selected.

Recombination operator 3:

Parameters: two parent solutions $\mathbf{x}_1$ and $\mathbf{x}_2$
Phase 1:
For each vehicle v that has an assigned route in $\mathbf{x}_1$
  With probability 0.5 place the whole route of vehicle v in $\mathbf{x}_1$ in the offspring solution
Phase 2:
For each vehicle v that has an assigned route in $\mathbf{x}_2$
  For each route segment *RS2* in the route of vehicle v in $\mathbf{x}_2$
    Remove from route segment *RS2* all sectors already assigned in phase 1
    If route segment *RS2* remains non-empty
      For each vehicle route *VR* in the offspring
        For each segment *RS* in *VR*
      Evaluate insertion of route segment *RS2* in *RS*
      Evaluate insertion of route segment *RS2* in a new route segment
      Insert route segment *RS2* in the best possible position, i.e., the position that results in the smallest increase to the total operating costs

5. Computational experiments

In the experiments, we used 10 different instances of the problem, each corresponding to a different day of the company operations. In each of these instances, the same set of 31 vehicles was available. The instances differed by the sets of sectors to be served and the amounts of waste in the sectors. The waste were transported to two dumping sites with different costs. Each instance included 100 sectors.

Fig. 3 contains experimental results with a number of different methods described in Section 4.4.2. GLS denotes the main version of the proposed algorithm with recombination operator 1; GLS2 denotes the version of the genetic local search algorithm with recombination operator 2 and GLS3 denotes the version of algorithm based on recombination operator 3. In all cases, populations of size 40 were used.

The proposed method is compared also to an evolutionary algorithm EA which uses standard roulette wheel selection. The evolutionary algorithm uses recombination operator 1. The initial population is obtained using the algorithm for

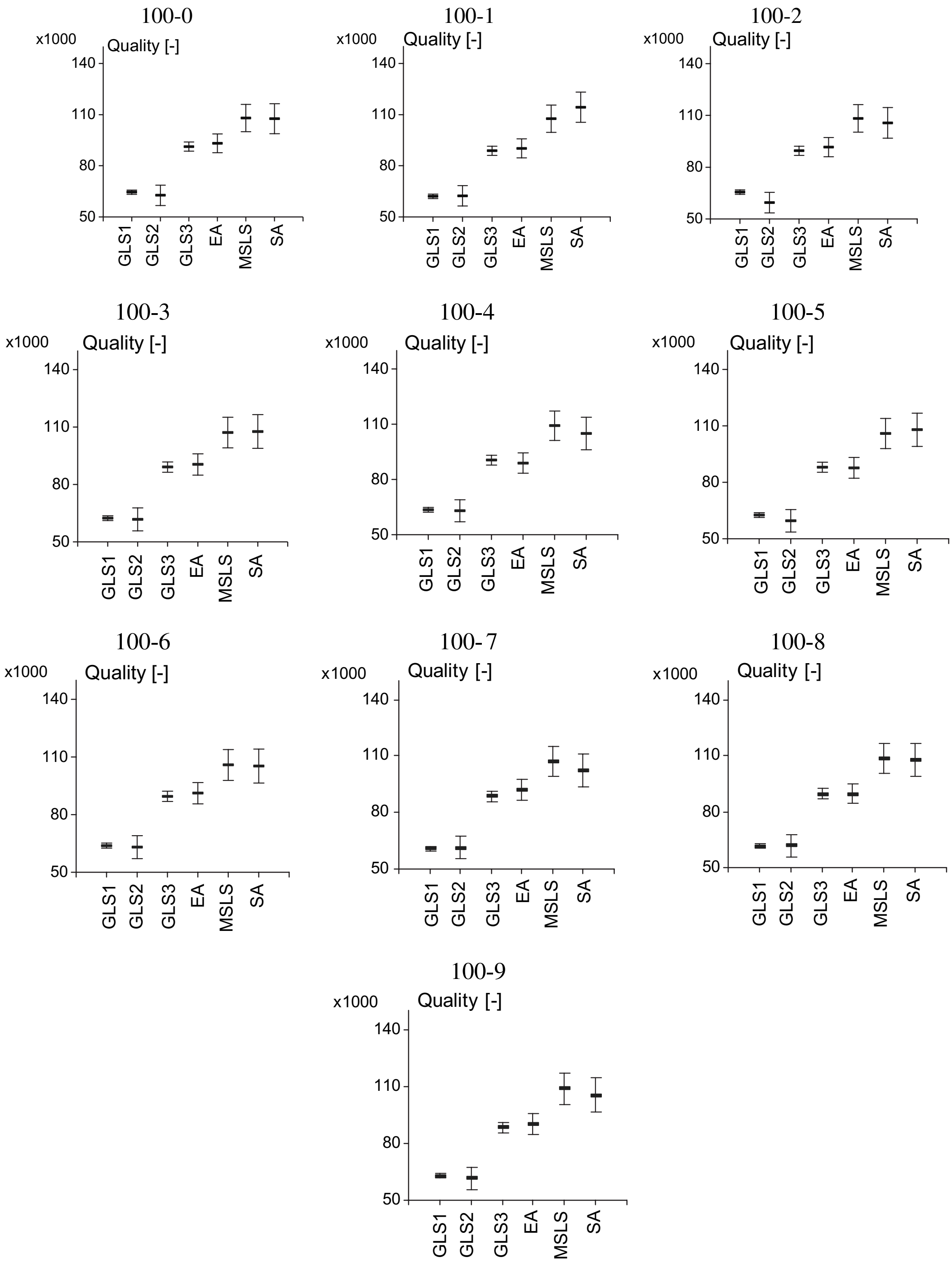

Fig. 3. Graphical comparison of the results. Each chart corresponds to a different instance. Each chart contains six box plots representing the distribution of cost for (from left to right) GSL1, GLS2, GLS3, EA, MSLS, SA. Note that the box plots are in many cases practically invisible because of the very low dispersion of the results.

finding initial solutions. The size of the population was 50. We have used a larger population than in the case of GLS because in EA smaller populations resulted in a very fast convergence to relatively poor solutions. We use the following version of the EA algorithm:

Parameters: size of the current population—N, stopping criterion
Initialization:
Current population $P := \emptyset$
repeat N times
  Construct a new feasible solution $\mathbf{x}$ by a randomized algorithm
Main loop:
repeat
  Draw at random with uniform probability two solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ from P.
  Recombine $\mathbf{x}_1$ and $\mathbf{x}_2$ obtaining $\mathbf{x}_3$
  if $\mathbf{x}_3$ is better than the worst solution in P and is different to all solutions in P **then**
    Add $\mathbf{x}_3$ to P and delete from P the worst solution
until the stopping criterion is met

The parameters of the methods were set as follows:

- size of the current population 50
- crossover probability 0.95

In addition, we compare results of GLS to multiple start local search MSLS and a simulated annealing SA algorithm. In the case of MSLS, local search is started from solutions constructed by the algorithm for finding initial solutions. The result of MSLS is the best local optimum found in the given number of iterations. SA also starts from solutions constructed by the algorithm for finding initial solutions. An intensive experiment was performed in order to find good settings for the starting and final temperature for SA. The number of moves at a temperature plateau was set in order to assure running times comparable to the other algorithms.

Greedy version of local search is used. It tests the neighborhood moves in random order and performs the first improving move. The local search algorithm is stopped when no improving move is found after testing all the possible neighborhood moves, which means that a local optimum is achieved.

We use neighborhood denoted by $V(x)$ with an operator that shifts a randomly selected sector between two randomly selected vehicle routes in the same way as in the case of the local search described in Section 4.3.

We used the following version of the SA algorithm:

Parameters: starting temperature T_0, the rules of temperature decreasing, stopping criterion, temperature update factor, number of moves at a temperature plateau L
  Generate a starting solution $x \in S$ using the algorithm for finding initial solutions $T := T_0$

Repeat
  For $i := 1$ **to** L **do**
    Construct $y \in V(x)$;
    If $f(y) < f(x)$ **then**
      $x := y$
    Else
      $x := y$ (accept y) with probability $\exp(-(f(y) - f(x))/T)$
  If the conditions for changing the temperature are fulfilled decrease T;
Until the stopping criterion is met

The parameters of the methods were set as follows:

- starting temperature 1500
- final temperature 50
- temperature update factor 0.99
- number of moves at a temperature plateau 12 000

We use two criteria to evaluate the performance of the tested methods: CPU time and quality (cost) of the best solution. Each algorithm was allowed to run for 120 seconds. The experiments were performed on a PC with Pentium 733 MHz processor. We have noticed an increase of the running time did not influence quality of results significantly. All the algorithms shared a common code, for the most part.

The reported results are averages over 10 runs of each algorithm on 10 different instances which correspond to different days of the company operations. Note that we used different instances from those used in the global convexity tests.

The results indicate that the genetic local search with the proposed recombination operators significantly outperforms the other methods. Good results were also obtained with recombination operator 2. Note that this recombination operator does not guarantee preservation of common pairs of sectors assigned together to a single route segment, however, it gives a good chance for preserving this feature. This is because each sector is inserted at the best possible position. This approach may restore many common arcs and pairs of sectors. An interesting observation is that in all cases the genetic local search with recombination operator 1 generates very good solutions with very low dispersion.

The use of recombination operator 3 in genetic local search gives results only slightly better than the evolutionary algorithm. Recombination operator 1 may been seen as a combination of the two other operators, because it preserves features preserved by these two operators. Thus, the use of recombination operators that preserves different types of common features seems to be a promising approach.

The evolutionary algorithm gives better results than multiple start local search and simulated annealing on the long run. The two latter methods, based on local search, are in general the worst performers.

The results of the experiments prove that the synergy of local search and recombination operators leads to methods that outperform algorithms based on local search only or recombination only.

Fig. 4 illustrates the changes in quality of new local optima and the CPU time needed to achieve these new local optima during an example run of the GLS algorithm with recombination operator 1. The first chart presents the CPU time needed to reach a new local optimum from the starting solution obtained by recombination. The second chart presents the quality of the local optima. Note that this chart presents the quality of all newly generated solutions even if they are not included in the current population. During the optimization process, the quality of solutions contained in the current population and thus the quality of recombined solutions improves. One can see that the use of the recombination operator indeed improves both the quality of the new local optima and reduces the CPU time needed to reach them.

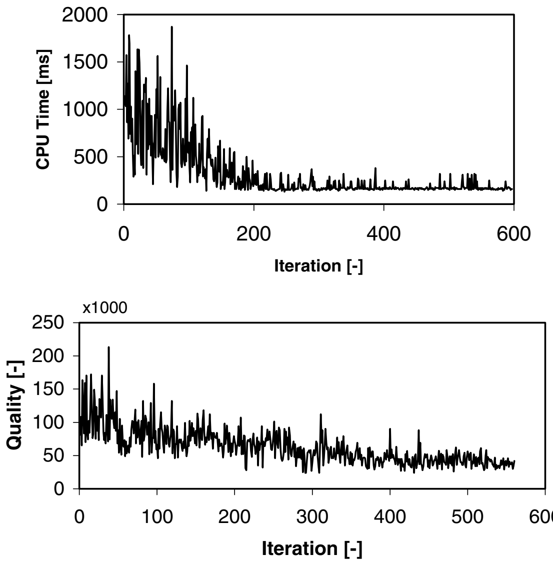

Fig. 4. Changes in quality of new local optima and CPU time needed to achieve the new local optima during a run of GLS algorithm.

6. Conclusions and directions for further research

A genetic local search algorithm has been applied to a real life vehicle routing problem. We used a systematic approach based on global convexity tests to develop appropriate recombination operators. Computational experiments demonstrate that genetic local search using the developed recombination operators gives high quality solutions in a relatively short time in comparison with other classic methods. It is also demonstrated that the use of the recombination operators improves both the quality of new local optima and reduces the CPU time needed to generate the local optima.

The best results were obtained with the recombination operator that preserves two kinds of features: common assignment of sectors to vehicles and common pairs of sectors assigned together to a single route segment. Thus, the use of recombination operators preserving different types of common features seems to be a promising approach.

The systematic approach presented in the paper was applied to a particular real life problem. The data are available from the authors upon request. We believe, however, that the idea is general and may be applied in many other cases. At present, we are applying this approach in the case of queries optimization in data warehouses. This systematic approach could be especially useful for practitioners facing real life problems that often differ from standard problems described in the literature. We believe that a systematic approach leads to efficient algorithms faster than traditional approaches based on intuition and extensive experiments with different recombination operators.

An interesting direction for further research is also taking into account similarities between some vehicles. The company may use several vehicles of the same type characterized by the same or very similar parameters. In this case similarity measures should not distinguish vehicles of the same type.

Acknowledgements

This work was supported by grant no. 8 T11F 006 19 from the State Committee for Scientific Research and by subsidy no. 4/2001 from the Foundation for Polish Science.

References

Ackley, D.H., 1987. A Connectionist Machine for Genetic Hillclimbing. Kluwer Academic Press, Boston.

Boese, K., Kahng, A., Muddu, S., 1994. A new adaptive multistart technique for combinatorial global optimization. Operations Research Letters 16, 101–113.

Freisleben, B., Merz, P., 1996. A genetic local search algorithm for travelling salesman problem. In: Voigt, H.-M., Ebeling, W., Rechenberg, I., Schwefel, H.-P. (Eds.), Proceedings of the 4th Conference on Parallel Problem Solving fram Nature-PPSN IV, pp. 890–900.

Galinier, P., Hao, J.-K., 1999. Hybrid evolutionary algorithms for graph coloring. Technical Report, Parc Scientifique Georges Besse, Nimes.

Glover, F., 1977. Heuristic for integer programming using surrogate constraints. Decision Science 8, 156–166.

Glover, F., 1995. Scatter search and star-paths: Beyond the genetic metaphor. OR Spektrum 17, 125–137.

Goldberg, D.E., 1988. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA.

Gorges-Schleuter, M., 1997. On the power of evolutionary optimization at the example of ATSP and large TSP Problems. In: European Conference on Artificial Life '97, Brighton, UK.

Jaszkiewicz, A., 1998. Genetic local search for multiple objective combinatorial optimization. Research report, Institute of Computing Science, Poznań University of Technology, RA-014/98, pp. 23.

Jones, T., Forrest, S., 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Santa Fe Institute Working Paper 95-02-022.

Lawler, E.L., Lenstra, J.K., Kan, A.H.G., Shmoys, D.B., 1985. The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley and Sons, New York.

Lin, S., Kernighan, B., 1973. An effective heuristic algorithm for the travelling salesman problem. Operations Research 21, 498–516.

Michalewicz, Z., 1992. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, Heidelberg.

Merz, P., Freisleben, B., 1997. Genetic local search for the TSP New results. In: Proceedings of the 1997 IEEE International Conference on Evolutionary Computation. IEEE Press, New York, pp. 159–164.

Merz, P., Freisleben, B., 2000. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Transactions on Evolutionary Computation 4 (4), 337–352.

Mühlenbein, H., 1991. Evolution in time and space—the parallel genetic algorithm. In: Rawlins, G.J.E. (Ed.), Foundations of Genetic Algorithms. Morgan Kaufmann Publishers, Los Altos, CA.

Potvin, J.Y., Bengio, S., 1996. The vehicle routing problem with time windows—part II: Genetic search. INFORMS Journal of Computing 8 (2), 165–172.

Radcliffe, N.J., Surry, P.D., 1994. Formal memetic algorithms. In: Fogarty, T. (Ed.), Evolutionary Computing: AISB Workshop. Springer-Verlag.

Sevaux, M., Dauzère-Pérès, S., 2003. Genetic algorithms to minimize the weighted number of late jobs on a single machine. European Journal of Operational Research 151 (2), 296–306.

Taillard, É.D., 1995. Comparison of iterative searches for the quadratic assignment problem. Location Science 3, 87–105.

Wolpert, D.H., Macready, W.G., 1997. No Free Lunch theorem for optimization. IEEE Transactions on Evolutionary Computation 1 (1), 67–82.