OR Spectrum (2005) 27: 21–41
DOI: 10.1007/s00291-004-0173-7

OR Spectrum

A Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows*

Giselher Pankratz

Department of Business Administration and Economics,
FernUniversität – University of Hagen, Profilstraße 8, 58084 Hagen, Germany
(e-mail: giselher.pankratz@fernuni-hagen.de)

Abstract. The Pickup and Delivery Problem with Time Windows (PDPTW) is a generalization of the well studied Vehicle Routing Problem with Time Windows (VRPTW). Since it models several typical planning situations in operational transportation logistics and public transit, the PDPTW has attracted growing interest in recent years. This paper proposes a Grouping Genetic Algorithm (GGA) for solving the PDPTW which features a group-oriented genetic encoding in which each gene represents a group of requests instead of a single request. The GGA is subject to a comparative test on the basis of two publicly available benchmark problem sets that comprise 9 and 56 PDPTW instances, respectively. The results show that the proposed GGA is competitive.

Keywords: Genetic Algorithms – Group-oriented encoding – Pickup and Delivery Problem with Time Windows

1 Introduction

The Pickup and Delivery Problem with Time Windows (PDPTW) can be described as follows (Savelsbergh and Sol, 1995): A set of transportation requests that is known in advance has to be satisfied by a given fleet of vehicles. Each request is characterized by its pickup location (origin), its delivery location (destination) and the size of the load that has to be transported from the origin to the destination. For each pickup and delivery location, a time window and loading and unloading times are specified. The load capacity, the maximum length of its operating interval, a start location and an end location are given for each vehicle. In order to fulfill the requests, a set of routes has to be planned such that each request is transported from its origin to its destination by exactly one vehicle. A reasonable objective

* The author wishes to thank the anonymous referees for their valuable comments.

function may use optimization criteria such as, number of vehicles employed, total distance traveled, total schedule duration or combinations of these. Basically, the PDPTW differs from the VRPTW by the additional precedence constraints, i.e. the restriction that the origin of each request has to be visited before the corresponding destination.

Formulations of the PDPTW as an integer program have been presented by, e.g., Dumas et al. (1991) and Savelsbergh and Sol (1995). Because this contribution concentrates on solution approaches, a review of formal models for the PDPTW is omitted here. Since the PDPTW is a generalization of the Vehicle Routing Problem with Time Windows (VRPTW), it is at least as complex as the latter, which has been proven by Lenstra and Rinnoy Kan (1981) to be NP-hard.

The PDPTW models a variety of operational planning problems in transportation logistics. Applications range from local area courier services to less-than-truckload transportation and long-distance haulage. In addition, the PDPTW also matches typical situations in public transit. For example, the Dial-a-Ride Problem with Time Windows (DARPTW) is a PDPTW in which people instead of goods are to be transported. While in the past research mainly concentrated on the VRPTW, increasing interest in the PDPTW can be observed in recent years. Nevertheless there is still relatively little attention paid to the PDPTW in the literature.

In accordance with definitions of the PDPTW usually found in the literature and in order to make the proposed approach comparable to other recently published approaches, it is assumed that the number of vehicles is unlimited and all vehicles have the same capacity and the same home location where all routes start and end. Note that there are several variants of the PDPTW in the real-world which differ from this definition in some ways. For example, the vehicles may have different capacities or the start and the end location of a route may not coincide.

The paper is organized as follows. Section 2 gives a review of previous work on the PDPTW and similar problems. In Section 3 the proposed solution approach, a Grouping Genetic Algorithm (GGA), is described in detail. Section 4 reports computational results obtained by the GGA for two publicly available benchmark data sets. Finally, the main findings are summarized in Section 5.

2 Literature review

In this section, an overview of important previous approaches for solving the PDPTW is given. Compared to the comprehensive review of models and approaches of Savelsbergh and Sol (1995), a rather brief survey is presented here, taking into account some more recent developments. The methods can be divided into optimal approaches, problem specific heuristics and metaheuristics.

2.1 Optimal approaches

The first optimization algorithms for the PDPTW were restricted to the single vehicle case, which is considerably easier to solve than the regular PDPTW with multiple vehicles. Important approaches of this kind are the dynamic programming

algorithms by Psaraftis (1980, 1983a) and Desrosiers et al. (1986). In the 1990s, the first exact approaches for the multi-vehicle PDPTW were published. Based on a formulation as a set partitioning problem, Dumas et al. (1991) proposed a branch-and-price algorithm for the multiple-vehicle PDPTW. Feasible routes are found by solving a constrained shortest path problem. As the authors point out, the algorithm is appropriate for instances in which each request occupies a relatively large percentage of vehicle capacity. By means of this algorithm, they were able to optimally solve two practical problems with 19 and 30 requests respectively. Savelsbergh and Sol (1998) also developed a branch-and-price algorithm for the multiple-vehicle PDPTW. In this algorithm, several sophisticated techniques, such as embedded heuristics and a special column management scheme, are employed in order to speed up search. Optimal solutions to several test cases with 30 requests each are reported. Since Savelsbergh and Sol observe significant differences in solution times, even for instances in the same problem class, they classify their optimization algorithm as unsuitable for practical planning situations.

2.2 Problem specific approximation methods

Due to the complexity of the PDPTW, several heuristic algorithms for the PDPTW have been published. Some of them construct solutions following a "cluster first, route second" approach. For example, Bodin and Sexton (1986) propose an iterative procedure, which, in each iteration, first (re)assigns requests to vehicles and then solves a single-vehicle PDPTW for each vehicle. A more fine-grained approach is presented by Desrosiers et al. (1988) who introduce the notion of a mini-cluster, i.e. a feasible route segment that serves one or more requests starting and ending with an empty vehicle. Each mini-cluster is treated as a single full truck load. After a large number of mini-clusters has been generated heuristically, a set partitioning problem is solved. Ioachim et al. (1995) pick up the idea of mini-clusters and present a method that employs an exact algorithm to produce an optimal set of mini-clusters.

Since insertion heuristics have proven successful for the VRPTW (Solomon, 1987), it appears reasonable to apply similar principles to the PDPTW as well. A frequently cited insertion procedure for the DARPTW, which can be easily adapted to the PDPTW, is presented by Jaw et al. (1986). Madsen et al. (1995) develop an insertion algorithm for the DARPTW which is a generalized and improved version of the heuristic of Jaw et al.

In order to improve the quality of solutions obtained by a construction heuristic, Van der Bruggen et al. (1993) propose a local improvement procedure for the single-vehicle PDPTW based on arc exchanges following the variable-depth search procedure of Lin and Kernighan (1973) for the Traveling Salesman Problem (TSP). Similar arc exchange operators for routing problems with precedence constraints are described by Psaraftis (1983b). In contrast to these arc-centered approaches, Toth and Vigo (1996) choose a request-oriented improvement method. They distinguish between intra-route movements that only affect the relative order of the locations in a route and inter-route movements that shift requests between routes.

2.3 Metaheuristics

Problem specific heuristics hold the danger of the search being trapped in a local optimum. Metaheuristics have become widely accepted as an appropriate means of avoiding this undesirable behavior. Compared to the VRPTW for which a considerably large number of metaheuristic approaches is reported in the literature, until now only a few metaheuristics have been reported for the PDPTW. Most of them are adaptations of the tabu search algorithm.

In the tabu search heuristic of Gendreau et al. (1998) the neighborhood definition is based on the concept of ejection chains. In an ejection chain, a request is eliminated from one route and inserted into another route, which in turn causes a request from that route to be shifted to a third route, and so on. As a diversification strategy, the authors make use of a so-called adaptive memory, which contains a number of the best solutions found thus far. After the search has terminated, a new solution is constructed from the routes in the memory, serving as a new starting solution. The algorithm is embedded into a rolling horizon framework and tested under different operating scenarios.

Based on the definition of three distinct moves, Nanry and Barnes (2000) propose a reactive tabu search for the PDPTW in which the search progress is continuously analyzed and search parameters such as the length of the tabu list are adjusted accordingly during search. Additionally, a special procedure is employed in order to detect and escape from chaotic attractor basins in the search space. In order to evaluate their approach, the authors use problem instances that were derived from Solomon test cases for the VRPTW (Solomon, 1987) for which optimal solutions are known. During problem transformation, for each PDPTW instance, the feasibility of its corresponding optimal VRPTW solution was maintained. Thus, the results can be directly compared to the optimal VRPTW solutions.

Yet another tabu search is presented by Lau and Liang (2001). A special construction algorithm, called a "partitioned insertion heuristic", is used in order to build starting solutions. The neighborhood definitions in this approach are adapted from Nanry and Barnes (2000). Instead of moving or exchanging single requests, they allow the exchange of so-called clusters, which are very similar to the notion of mini-clusters introduced by Desrosiers et al. (1988). The approach is tested on the nine 100-node instances provided by Nanry and Barnes (2000). Furthermore, the authors give a detailed description of a transformation scheme by which they generate additional PDPTW instances based on a selection of another 27 Solomon VRPTW instances. The results for these instances are also reported.

Li and Lim (2001) developed a hybrid metaheuristic called tabu-embedded simulated annealing. Three different request swapping moves are used to define local search neighborhood structures. Basically, a simulated annealing procedure is performed, and is restarted from the current best solution every time a given number of iterations without improvement has been observed. In addition, a tabu list keeps track of the recently visited solutions in order to avoid cycling. The approach is evaluated using the nine 100-node test cases of Nanry and Barnes. Additionally, Li and Lim report results on 56 own problem instances generated on the basis of the full set of all 56 Solomon VRPTW instances with 100 customers.

Despite the success of Genetic Algorithms for the VRPTW (see e.g. Blanton and Wainwright, 1993; Thangiah, 1995; Potvin and Bengio, 1996; Berger et al., 1998), until now only few applications of Genetic Algorithms have been reported for varieties of the PDPTW.

A Genetic Algorithm (GA) for the Dial-a-ride Problem (DARP) is presented by Potter and Bossomaier (1995). Since the DARP does not comprise time windows, it is a simplified special case of the PDPTW. In this approach, the problem is solved by two interconnected Genetic algorithms: On the upper level, a GA is responsible for assigning transportation requests to vehicles, whereas on the lower level a second GA for each vehicle determines a route covering the requests assigned to this vehicle. For lack of any DARP benchmark instances, the approach is evaluated using Vehicle Routing (VRP) instances from the Travelling Salesman Problem Library (TSPLib).

Jih and Hsu (1999) have developed a hybrid GA for solving the single-vehicle PDPTW. The initial population contains a set of routes generated by a dynamic programming algorithm which is aborted prior to completion. The Genetic Algorithm is tested on five single-vehicle PDPTW instances comprising from 10 up to 50 requests.

Schönberger et al. (2003) propose a GA for solving the so-called Pickup and Delivery Selection Problem (PDSP) which extends the PDPTW by the decision of acceptance or rejection of transportation requests. The goal is to maximize overall profit. To this end, a revenue value is assigned to each request, so this problem significantly differs from the PDPTW considered in this contribution. For testing purposes PDSP instances have been generated on the basis of the PDPTW instances of Nanry and Barnes (2000) by adding revenue values to all instances.

To our knowledge, up to now only Jung and Haghani (2000) have reported a GA for solving the multi-vehicle PDPTW. In contrast to the PDPTW treated by this paper, the authors consider so-called soft time windows, i.e. the violation of time window constraints leads to penalty cost but does not compromise feasibility of a solution. The algorithm was tested on 24 randomly generated problems of varying problem size, ranging from 5 up to 30 requests.

3 A Grouping Genetic Algorithm for solving the PDPTW

As shown by the literature review, only a few approaches exist which apply Genetic Algorithms to variants of the PDPTW, most of them treating simplified special cases of the PDPTW. Probably the major reason why Genetic Algorithms for solving the PDPTW are rare is the fact that it is very difficult to find an appropriate genetic representation for this complex problem.

Thus, it is an interesting task to verify the appropriateness of Genetic Algorithms for this problem class. The concept of Genetic Algorithms was first introduced by Holland (1975). In particular, Genetic Algorithms appear attractive because of their population-based approach, which allows the parallel exploration of different paths through the solution space of a problem. In addition, Genetic Algorithms are considered to be robust and easy to implement (Rayward-Smith, 1994).

In this section, a new Genetic Algorithm for solving the PDPTW is presented. Throughout the following description it is assumed that the reader is familiar with the terminological and conceptual basics of Genetic Algorithms. Introductory treatments of Genetic Algorithms as a class of optimization algorithms can be found in Goldberg (1989) and Davis (1991).

Because the proposed Genetic Algorithm employs an adaptation of the group-oriented genetic encoding introduced by Falkenauer (1998), it is denoted as a Grouping Genetic Algorithm (GGA). This section gives a top-down description of the proposed GGA: First, the basic search scheme of the GGA is outlined. Second, the group-oriented genetic encoding for PDPTW solutions is explained. Third, the genetic operators are described in detail. Finally, a rough characterization of the embedded insertion algorithm that is used at several stages of the search is given.

3.1 Genetic search scheme

Figure 1 describes the general search scheme of the proposed GGA.

initialize population P;
while not (termination criterion is met)

select a pair of individuals x,y from P as parents with regard to their fitness value;
generate two children x',y' applying the crossover operator to x and y with probability p^{cross};
generate two modified children x'',y'' applying the mutation operator to x' and y', respectively, with probability p^{mut};
insert x'' and y'' into P and in turn remove the two worst individuals from P;

return best individual from P as solution.

Fig. 1. General search scheme for the proposed GGA

An initial population is generated using the embedded insertion heuristic described below. The population size, n^{pop}, is a parameter of the GGA. Unlike the classical GA, which employs generational replacement, the population management is done following the steady-state approach without duplicates (Syswerda, 1989; Whitley, 1989). According to this approach, each newly generated pair of offspring is inserted immediately into the current population where it replaces the two worst individuals. This incremental approach ensures the survival of the best solution over the whole search and prevents the occurrence of duplicate individuals. Thus, it often reveals higher performance than generational replacement (Davis, 1991). In order to detect duplicates, a simple comparison of objective values proved satisfactory for the proposed GGA, so it was preferred over any other time consuming procedure seeking genotypical or phenotypical differences between individuals. Note that crossover and mutation are employed in a sequential fashion. The crossover operator is applied to each selected pair of parent chromosomes with probability p^{cross}, whereas the mutation operator is applied to each offspring with probability p^{mut}. Both p^{cross} and p^{mut} are parameters of the GGA. If the crossover

operator is not applied according to its execution probability p^{cross}, the children are simply clones of their parents. Similarly, if the mutation operator is not applied, the offspring leave the mutation operator unchanged. The search terminates after a given total number, $\tilde{n}^{max}$, of individuals has been generated without improvement but no later than after a given maximum number, n^{max}, of generated individuals has been reached.

The adaptation of the described genetic search scheme to the problem at hand involves the following components of the algorithm:

- The genetic encoding, i.e. the way solutions to the PDPTW are represented by chromosomes (strings);
- the genetic operators, i.e. selection, crossover and mutation;
- the embedded heuristic, i.e. the subordinate heuristic procedure that is employed by the GGA in order to generate an initial population and to produce feasible offspring.

The design decisions concerning these components are dealt with in the following sections.

3.2 Group-oriented genetic encoding for the PDPTW

The way in which solutions to the problem at hand are encoded is crucial for the success of any Genetic Algorithm. One difficulty of finding a good genetic encoding for solutions to the PDPTW is the fact that the PDPTW – like the VRPTW – is a combination of two interdependent subproblems: On the one hand, the subproblem of clustering requests and assigning them to a vehicle has to be solved (clustering or grouping problem). On the other hand, for each cluster of requests, a feasible route has to be specified in which the corresponding pickup and delivery nodes are visited (ordering or routing problem). It is not obvious how these two aspects of a solution can be represented simultaneously by a homogeneous encoding such as the standard binary representation of the classic Genetic Algorithm.

For pure routing problems like the TSP sometimes a representation is chosen which encodes the sequence in which the nodes are visited as a permutation of all locations. A number of specialized crossover operators that preserve the permutation property of the offspring have been developed. However, such an encoding brings about some troubles when being applied to highly constrained multi-vehicle routing problems like the PDPTW:

- Because only the routing part of the PDPTW can be represented by such an encoding, the grouping part must be covered otherwise, e.g. by external heuristics.
- Due to capacity, precedence and time window constraints a large portion of all possible permutations will turn out to be infeasible solutions. Thus, special mechanisms have to be implemented in order to maintain feasibility of the generated solutions.

Among the Genetic Algorithms mentioned in the literature review, Potter and Bossomaier (1995), Jih and Hsu (1999) as well as Schönberger et al. (2003) use such a permuation-based encoding.

Jung and Haghani (2000) used a so-called random key representation in order to simultaneously encode both the routing and the grouping part of the PDPTW. Under this encoding, a random four-digit key number is assigned to each location. The first digit denotes the index of the vehicle visiting that location whereas the last three digits represent a sort key to determine the position of that location within the route. A solution is decoded by sorting all locations in ascending order of their key numbers. When generating the chromosome, it is assured that the value of the pickup location is always less than the value of the corresponding delivery location. In addition, the pickup and delivery nodes for the same request always have the same number in the first digit. However, many of the individuals generated are infeasible with regard to capacity and time restrictions and therefore have to be discarded.

Alternatively, a representation could be chosen which encodes a solution as a permutation of all requests instead of the nodes. At the phenotype level, a chromosome of this kind can be decoded by a heuristic which constructs a solution considering the requests in the order given by the permutation. In the GA literature this rather indirect encoding is quite popular. Applications to the VRP and the VRPTW are reported for example by Kopfer et al. (1994) and Blanton and Wainwright (1993), respectively. To our knowledge, this order-based representation has not yet been applied to the PDPTW. Nevertheless, this order-based representation with a heuristic decoder has some important drawbacks, especially for problems with a strong grouping component (see Falkenauer, 1998):

- Identical solutions are often represented by a large number of different permutations. This redundancy obviously weakens the efficiency of the genetic search.
- The phenotypical meaning of a gene strongly depends on its genotypical context, e.g. the genes that precede it in the permutation. Most order-based crossover operators are context-insensitive and do not take into account such contextual information during recombination. Thus, in many cases, even small modifications in a chromosome, e.g. by a crossover operator, will cause the offspring to have almost no phenotypical similarities to its parents. As a consequence, it is hard for the GA to sample meaningful building blocks.
- Due to the strong context-dependency of the genes, good schemata are long, which in turn increases the probability of them being destroyed by a syntactic crossover operator. This severely obstructs the search progress of the GA.

With regard to the above mentioned disadvantages of prevailing encodings, and due to the fact that it is very difficult to find an encoding which represents all the aspects of a PDPTW solution, a different way to encode solutions of the PDPTW is chosen here. It traces back to the consideration if there is probably one of the subproblems of the PDPTW which has more influence over the solution quality than the others. If so, it appears reasonable to concentrate on this dominating aspect when seeking

an appropriate representation. This would significantly alleviate the problem of finding an encoding.

Obviously, the grouping part of the PDPTW is of great importance for the overall quality of a solution to the PDPTW (Savelsbergh and Sol, 1998). In particular, because both the time windows and the precedence constraints considerably restrict the number of routing alternatives for a given allocation of requests, the grouping aspect turns out to be dominant over the routing aspect in many cases. Because of this dominance of the grouping part of the PDPTW, and in order to avoid the drawbacks of an order-based encoding, a representation of solutions to the PDPTW is proposed which refers to the encoding principles of the so-called Grouping Genetic Algorithm (GGA) introduced by Falkenauer (1998). In this representation, each gene represents a group of objects instead of a single object. Thus, the groups are the building blocks that are sampled and recombined by the genetic operators to produce solutions of potentially higher quality. Falkenauer originally developed his GGA for pure grouping problems such as the Bin Packing Problem (BPP). The following describes how the original encoding was adapted to the PDPTW.

Each gene in a chromosome represents a cluster of all requests that are assigned to a single vehicle. The length of a chromosome, i.e. the number of genes is variable and depends on the number of vehicles required by a given solution. Figure 2 illustrates the encoding of a solution by means of a simple example with seven requests that are carried out by three vehicles.

By representing a partitioning of all transportation requests, a chromosome under this encoding only covers the grouping aspect of a PDPTW solution. As a consequence, the remaining routing aspect of a solution has to be added while decoding the chromosome. This is done by associating each chromosome with separate data structures containing the routing information for each gene. Since this information is hidden from the GGA, it cannot be manipulated directly by the genetic operators. Instead, the routes are constructed and maintained using a separate heuristic according to the grouping information encoded in the chromosome.

3.3 Genetic operators

In the proposed GGA, the binary tournament selection mechanism (see e.g. Goldberg et al., 1990) is applied as the selection operator. In this scheme, two individuals are chosen randomly and the one which represents the better solution is selected as the first parent. To obtain the second parent, the procedure is repeated. An important advantage of tournament selection is its low time complexity compared to the classical "roulette wheel" selection scheme. Furthermore, because a simple comparison of the objective values is sufficient, tournament selection does not require the calculation of any transformation of the objective value in order to control selection pressure.

The crossover operator for the proposed GGA is an adaptation of the general group-oriented crossover scheme presented in Falkenauer (1998) and proceeds in five steps, as illustrated in Figure 3.

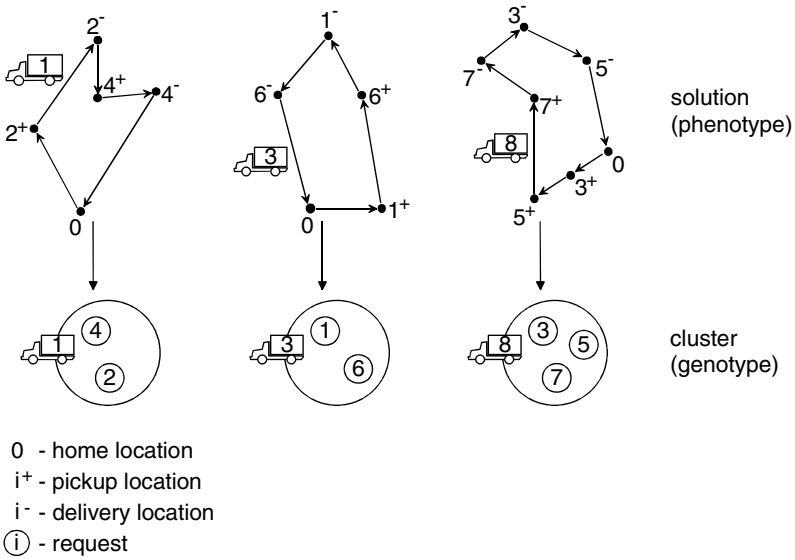

Fig. 2. Group-oriented encoding (example)

1. Specify a crossing section, i.e. a coherent segment of clusters, by randomly selecting two crossing points in each of the two parent chromosomes.
2. Insert the clusters in the crossing section of the second parent at the first crossing point in the chromosome of the first parent. The respective routes are directly adopted from the second parent without reconstruction. As a result of this operation a number of requests may occur twice (requests 1, 2, 6 and 7 in Fig. 3). Moreover, two clusters may be assigned to the same vehicle (vehicle 1 in Fig. 3).
3. To resolve the conflicts mentioned above, remove all clusters originally belonging to the first parent if they refer to a vehicle that is already allocated by any of the newly inserted clusters. If afterwards some requests still occur twice, eliminate them from those clusters that originally belong to the first parent. At the phenotype level, all eliminations are reproduced immediately in the routes appendant to the changed clusters. Note that the clusters and the respective routes imported from the second parent are left unchanged. After this step, some requests may remain unassigned (requests 5 and 8 in Fig. 3).
4. Reinsert the unassigned requests in random order into the individual applying the insertion heuristic. This may require the allocation of an additional vehicle in order to assure feasibility. As a result of this step, a complete offspring is obtained.
5. Generate the second offspring by repeating steps 2 through 4 with reversed roles of the parents.

The group-oriented mutation operator for the PDPTW works as follows:

1. Select a cluster at random in the individual.

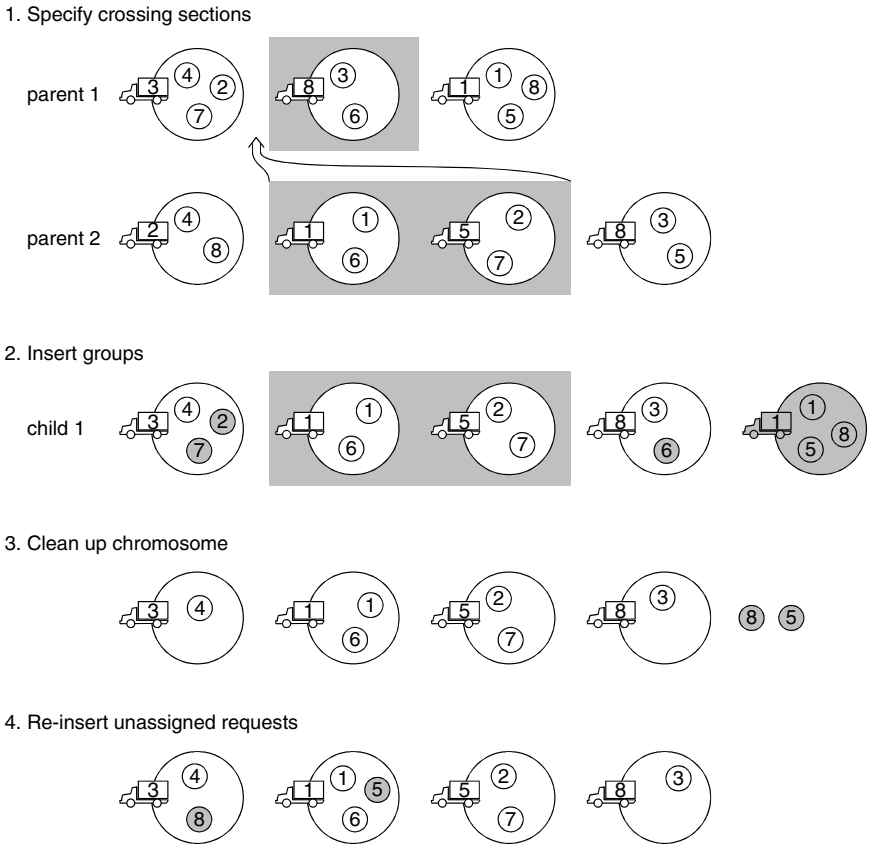

Fig. 3. The group-oriented crossover operator

2. Eliminate this cluster from the chromosome and remove the associated route from the phenotype.
3. Re-insert all removed requests into the individual by means of the insertion heuristic, allocating a new vehicle if necessary to maintain feasibility.

3.4 Embedded insertion heuristic

At several stages during search, the GGA employs a subordinate insertion heuristic while constructing and modifying solutions. The general procedure of this heuristic can be described as follows: For each request to be inserted, all feasible insertions in all existing routes of the current (partial) solution are examined. To this end, all possible insertion positions for the pickup and the delivery node in a route are tested taking into consideration precedence, capacity and time constraints. Additionally, a new vehicle is allocated and a new route for this vehicle is tentatively initialized with the current request. Among all feasible insertions, the one that causes minimal additional cost is selected and implemented.

In order to generate an initial solution by means of this procedure, all transportation requests are considered in random order, starting with a single, empty vehicle. After all transportation requests have been inserted, a feasible initial solution is available. The procedure is repeated until an initial population of the desired size is obtained.

In order to re-adjust the routing part of a solution due to grouping modifications caused by the genetic operators, the insertion procedure is also used. Instead of constructing a new solution from scratch every time the grouping is modified by crossover or mutation, each request that is subject to regrouping is deleted from its original route and reinserted without disturbing the relative order in which the remaining nodes are visited. Thus, the original routing decisions in the unchanged parts of a solution are preserved.

4 Computational study

This section reports the results of the proposed GGA for a number of benchmark problems when compared to previous metaheuristics for the PDPTW. First, the benchmark data sets are described, then the numerical results are discussed.

4.1 Benchmark data sets

In order to test the proposed GGA, two benchmark data sets for the PDPTW for which results have been published in the literature are used.

1. The nine 100-node-instances provided by Nanry and Barnes (2000), in the remainder referred to as problem set 1. Results for these instances are also reported by Li and Lim (2001) and Lau and Liang (2001).
2. The 56 100-node-instances provided by Li and Lim (2001), in the remainder referred to as problem set 2.

In these problem sets, each PDPTW instance was derived from a certain instance out of Solomon's 100-customer VRPTW problems by randomly pairing up customers within the routes of a given reference solution. Supplementary dummy nodes were introduced for coupling purposes if an odd number of customers is present in a route of the original solution. In order to keep the reference solution feasible, the dummy nodes are collocated with their paired nodes. So each instance has at least 100 nodes, which is equivalent to at least 50 requests.

Nanry and Barnes restricted themselves to Solomon instances for which optimal solutions are known. They only generated instances from Solomon's C1 class. In all problems of this class, the customers are spatially clustered and the problems have a short scheduling horizon. For example, the instance denoted as nc101 is derived from Solomon's VRPTW instance c101. For consistency reasons, minor adjustments were necessary to some of the instances of Nanry and Barnes (2001) because, for some requests, the pickup and delivery quantities did not sum up to zero (see also Li and Lim, 2001; Lau and Liang, 2001). This could easily be fixed

Table 1. GGA parameter values

Parameter	Value	Description
n^{pop}	200	population size
p^{cross}	1.0	crossover probability
p^{mut}	0.5	mutation probability
n^{max}	15,000	max. number of individuals
$\tilde{n}^{max}$	3,000	max. number of individuals without improvement

by setting the delivery quantity of these reqests equal to the negative value of the corresponding pickup quantity.

In order to generate PDPTW test problems from all Solomon instances, Li and Lim (2001) refer to near-optimal solutions in cases where the optimal solution is unknown. Following the classification of Solomon, they organized their instances into six classes, denoted as LC1, LC2, LR1, LR2 LRC1 and LRC2, indicating the respective spatial distribution of the nodes and the length of the scheduling horizon. In LC1 and LC2 problems, nodes are clustered, whereas in LR1 and LR2 problems, they are randomly distributed. The nodes in LRC1 and LRC2 instances are partially clustered and partially randomly distributed. While the instances in LC1, LR1 and LRC1 have a short scheduling horizon, the horizon is longer in instances of classes LC2, LR2 and LRC2.

4.2 Numerical results

The GGA was implemented in JAVA and evaluated using problem set 1 and problem set 2. All tests were carried out on a PC (Pentium 4 processor, 2 GHz) under the objective of minimizing total travel distance. Because the GGA is stochastic in nature, the GGA was run 30 times on each instance in order to be able to judge the robustness of the GGA. Euclidean distances and travel times were calculated to two decimal places as is standard practice in the recent VRPTW literature. The parameter settings for the GGA, which were determined in a small number of preliminary experiments, are shown in Table 1. The values were chosen mainly with the idea of performing a thorough search but also to keep computation times reasonable.

In the remainder, the results obtained by the GGA on problem set 1 will be compared to the results reported by Nanry and Barnes (2000), Li and Lim (2001) and Lau and Liang (2001). The GGA results for the instances of problem set 2 are compared to the results of Li and Lim (2001). However, it is difficult to directly compare the results of the different approaches.

One important problem is that the approaches apply slightly different objective criteria. Nanry and Barnes (2000) conducted their experiments under the objective of minimizing the sum of total travel time and total service time. Since vehicle

speed is set to 1 and total service time is a constant if all nodes are visited, this objective is basically equivalent to minimizing total travel distance, which was also chosen as the objective for the GGA. In contrast, Li and Lim (2001) as well as Lau and Liang (2001) consider the number of vehicles as the first criterion, followed by total travel distance. Furthermore, the approaches differ in the way euclidean distances are calculated. Even though it is not explicitly stated in the paper, the objective values of the optimal VRPTW solutions used as a benchmark reveal that in Nanry and Barnes (2000), distance is rounded or truncated to one decimal position. In contrast, it seems that Lim and Li (2001) as well as Lau and Liang (2001) calculate distance to at least two decimal points. Therefore, even identical solutions may lead to significantly different objective values. Moreover, only best results are reported, even if the proposed method obviously employs stochastic components, as in the tabu-embedded simulated annealing approach of Li and Lim (2001). The overall number of independent runs per instance and average solution quality, which are important pieces of information when judging a stochastic method, are not disclosed. Finally, due to different computer platforms and programming languages, a comparison of computing times is difficult. The approach of Nanry and Barnes (2000) was coded in C and the runs were performed on an IBM RISC 6000 workstation. Li and Lim (2001) have coded their method in C++ with Standard Template Library, using an i686 under Linux. Lau and Liang (2001) do not specify their experimental environment.

The results obtained by the GGA for problem set 1 and problem set 2 are listed in Table 2 and Table 3, respectively. In these tables, td denotes total travel distance, nv the number of vehicles, and ct the computing time in seconds. Since Lau and Liang (2000) did not report a solution to nc102 nor any computing times, the respective cells in Table 2 are empty. For the GGA, aside from the values of the respective best solution (td^{best} and nv^{best}), average solution quality and average computing time is reported for each instance (td^{avg}, nv^{avg}, and ct^{avg}).

Under the assumption that the optimal VRPTW solutions remain very good, if not optimal solutions to the derived PDPTW instances, each of them provides a high quality benchmark for the total travel distance of the corresponding PDPTW instance. For this reason, Table 2 also shows the optimal values for the corresponding VRPTW instances of problem set 1 (columns two and five). In order to allow a careful comparison of the approaches despite different distance calculations, the results of Nanry and Barnes are compared to the optimal VRPTW solutions of Desrochers et al. (1992) and Kohl and Madsen (1997), which were calculated using distance truncation after one decimal place ("Opt A" in the second column). When calculated to two decimal places, the objective values of the optimal VRPTW solutions are slightly different ("Opt B" in the fifth column, taken from Homberger, 2000). The latter values should be taken into account when comparing the results of the rest of the approaches, including the GGA.

The results presented in Table 2 can be summarized as follows:

- Interestingly, for certain instances (nc104, nc107, nc108, and nc109) some methods find solutions of less travel distance when compared to the corresponding optimal VRPTW solution. Due to alternating loading and unloading activities, the load of the vehicles in a typical PDPTW solution is oscillating.

Table 2. Results for problem set 1

Inst.	Opt. A	Nanry/Barnes (2000)		Opt. B	Li/Lim (2001)		Lau/Liang (2001)		GGA		
	td	td	ct	td	td	ct	td	ct	td^{best}	td^{avg}	ct^{avg}
	nv	nv		nv	nv		nv		nv^{best}	nv^{avg}	
nc101	827.3	827.3	0.25	828.94	828.9	35	828.9	–	828.94	828.94	64.93
	10	10		10	10		10		10	10	
nc102	827.3	827.3	1.91	828.94	828.9	130	–	–	828.94	828.94	83.1
	10	10		10	10				10	10	
nc103	826.3	829.9	16.50	828.06	831.9	156	837.7	–	831.87	831.87	117.13
	10	10		10	10		10		10	10	
nc104	822.9	834.7	328.87	824.78	869.5	1539	989.7	–	816.74	816.74	143.67
	10	10		10	9		9		10	10	
nc105	827.3	827.3	0.51	828.94	828.9	25	829.8	–	828.94	828.94	69.83
	10	10		10	10		10		10	10	
nc106	827.3	827.3	0.54	828.94	828.9	31	828.9	–	828.94	828.94	74.73
	10	10		10	10		10		10	10	
nc107	827.3	826.1	9.36	828.94	827.8	22	828.9	–	827.82	827.82	74.10
	10	10		10	10		10		10	10	
nc108	827.3	826.1	31.49	828.94	827.8	39	826.1	–	827.82	829.11	84.9
	10	10		10	10		10		10	10	
nc109	827.3	827.3	184.76	828.94	827.8	85	828.9	–	827.82	827.82	116.87
	10	10		10	10		10		10	10	

Thus, the capacity constraint obviously has lost its original tightness, and the optimal solutions for the corresponding VRPTW instances are no longer optimal for the generated PDPTW instances. Nevertheless, they remain useful benchmarks for comparison purposes.

– As mentioned above, because of different distance calculation, any comparison of the GGA with the method of Nanry and Barnes can only be done indirectly by referring to the respective optimal VRPTW solutions as a benchmark. Taking this into account, the GGA achieves results of better solution quality for problems nc104 and nc109, while it is only slightly behind with regard to problem nc103: For problem nc103, the GGA, like the competing methods, fails to find a solution of the same quality as the corresponding optimal VRPTW solution ("Opt B"). The relative error is 0.5 percent. The solution reported by Nanry and Barnes for this instance has a lower relative error (0.4 percent). For problem nc104, total travel distance of the GGA solution is about 1 percent less compared to the corresponding optimal VRPTW solution ("Opt B"), whereas the total travel distance of the solution of Nanry and Barnes is about 1.4 percent longer than its associated benchmark ("Opt A"). In the case of problem nc109, the method of Nanry and Barnes measures up to the optimal VRPTW solution

("Opt A"), but the GGA solution in this case is slightly shorter than its reference solution ("Opt B").
- Unlike the method of Nanry and Barnes, the algorithms of Li and Lim and Lau and Liang can be directly compared to the GGA. Except for nc104, the best results of Li and Lim are identical with those of the GGA. Note that for problem nc104, Li and Lim as well as Lau and Liang report solutions with greater travel distance but one vehicle saved. This is due to the fact that the methods of these authors consider the number of vehicles as the primary objective while travel distance is the secondary criterion. When compared to the method of Lau and Liang, the GGA finds solutions of better quality for 4 problems (nc103, nc105, nc107, and nc109). Only for problem nc108, Lau and Liang report a solution which is about 0.2 percent shorter than the best solution found by the GGA.
- An interesting observation is the robustness of the proposed GGA with regard to problem set 1, since except for instance nc108, the GGA finds its best solution to each problem consistently in all 30 runs. Due to this stable and consistent behavior of the GGA, it appears justifiable to refer to average computing times when comparing the computing times of the GGA with those of the competing approaches. However, the GGA needs more time in most cases. But, since the average computing times are less than 2.5 minutes even in the worst case, the GGA throughout keeps computational cost within reasonable limits. This also holds for difficult problems like problem nc104, for which all competing methods require significantly more computing time.

For problem set 2, Li and Lim only report the best results obtained by their tabu-embedded simulated annealing algorithm. In order to facilitate comparison, in Table 3, the respective best values with regard to travel distance and number of vehicles for each instance are highlighted as bold figures. From Table 3 the following conclusions can be drawn:

- With respect to total travel distance, which is the original objective of the GGA, the best results of the GGA are superior to those of Lim and Li in 15 cases. On the other hand, in 9 cases the method of Li and Lim yields solutions with less travel distance than the GGA. However, because this actually is not the objective function used by Li and Lim, a fair comparison should also take into account the hierarchical criterion used by the method of Li and Lim.
- In 12 cases the GGA finds solutions with less travel distance without an increase in the number of vehicles reported by Li and Lim (printed in bold italics in Table 3). In these cases, the GGA has found new best results with regard to the objective criterion used by Li and Lim. For problem lrc102, the GGA even finds solutions that have less total travel distance and a smaller number of vehicles than the best solutions of Li and Lim. On the other hand, in 12 cases the method of Li and Lim yields better solutions than the GGA with respect to their objective.
- Even though the GGA fails to find its best solution for each instance of problem set 2 throughout all runs, the GGA turns out to be reasonably robust: for almost 90 percent of all instances, a standard deviation of total travel distance of less than 2 percent was observed. Interestingly, it appears that the GGA

Problem Class	Li/Lim (2001)		GGA	
	td^{best} (mean)	nv^{best} (mean)	td^{best} (mean)	nv^{best} (mean)
nc1	833.40	9.89	827.53	10.00
lc1	832.09	9.89	827.27	10.00
lc2	589.23	3.00	589.86	3.00
lr1	1222.20	11.92	1220.21	12.00
lr2	973.87	2.73	969.30	2.82
lrc1	1387.60	11.63	1386.04	11.63
lrc2	1187.82	3.25	1135.13	3.50

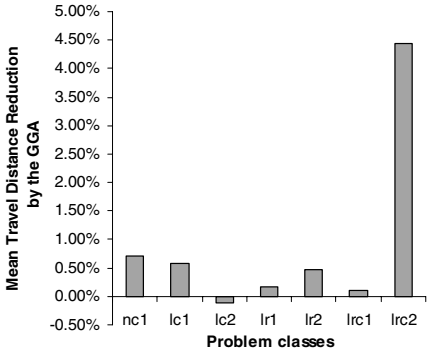

Fig. 4. Comparison of aggregated computational results

has a tendency to find better solutions especially for instances of the lrc1 and lrc2 problem classes, in which the nodes are partially clustered and partially randomly distributed.

– In order to solve the instances of problem set 2, the GGA requires moderate average computational times that range from less than one minute to less than nine minutes. Unfortunately, the description of Li and Lim does not indicate whether the computing times reported for their method are average running times or whether each time value denotes the computing time required for the particular run which resulted in the respective best solution. However, when compared to these times, the average computing times of the GGA are significantly shorter for almost all instances of problem classes lr1, lr2, lrc1, and lrc2. On the other hand, the times reported by Li and Lim are less for the majority of the lc1 and lc2 instances.

In order to further facilitate the comparison of the GGA and the method of Li and Lim, in Figure 4, the respective results are presented as averages over each problem class.

Beside the results for the six problem classes of problem set 2 (lc1 through lrc2), the table in Figure 4 also contains aggregated results for problem set 1, denoted here as problem class nc1. For each problem class, the mean travel distance and mean number of vehicles are displayed. The calculation of these values is based on the best solutions that were found by the method of Li and Lim (2001) and the GGA, respectively. With respect to total travel distance, which is the objective of the GGA, in the average, the GGA yields better solutions for all problem classes except for problem class lc2, where the average travel distance achieved by the GGA is 0.11 percent longer. This is also illustrated by the diagram on the right in Figure 4. For problem class lrc2, the GGA achieves the highest average savings in travel distance when compared to the method of Li and Lim (almost 4.5 percent). As the table shows, except for problem class lrc1, average travel distance reduction is achieved at the expense of a slightly higher number of vehicles.

To sum up one may say that for problem set 1, the GGA reaches the solution quality of the method of Li and Lim. Taking into account the different reference optima, the best results of the GGA are at least comparable to those found by

Table 3. Results for problem set 2

Inst.	Li and Lim (2001)			GGA				
	td^{best}	nv^{best}	ct	td^{best}	nv^{best}	td^{avg}	nv^{avg}	ct^{avg}
lc101	828.94	10	33	828.94	10	828.94	10.00	55.03
lc102	828.94	10	71	828.94	10	828.94	10.00	68.67
lc103	827.86	10	191	827.86	10	827.86	10.00	85.33
lc104	861.95	***9***	1254	***818.60***	10	818.67	10.00	139.90
lc105	828.94	10	47	828.94	10	828.94	10.00	58.33
lc106	828.94	10	43	828.94	10	828.94	10.00	63.60
lc107	828.94	10	54	828.94	10	828.94	10.00	65.97
lc108	826.44	10	82	826.44	10	826.44	10.00	81.00
lc109	827.82	10	255	827.82	10	827.82	10.00	118.10
lc201	591.56	3	27	591.56	3	591.56	3.00	59.27
lc202	591.56	3	94	591.56	3	591.56	3.00	115.50
lc203	***585.56***	3	145	591.17	3	591.17	3.00	191.67
lc204	591.17	3	746	***590.60***	3	620.38	3.00	346.13
lc205	588.88	3	190	588.88	3	588.88	3.00	94.07
lc206	588.49	3	88	588.49	3	588.49	3.00	124.20
lc207	588.29	3	102	588.29	3	588.35	3.00	137.03
lc208	588.32	3	178	588.32	3	588.75	3.00	141.93
lr101	***1650.78***	19	87	1650.80	19	1650.80	19.00	67.00
lr102	1487.57	17	1168	1487.57	17	1488.74	17.00	89.53
lr103	1292.68	13	169	1292.68	13	1292.67	13.00	82.87
lr104	***1013.39***	9	459	1013.99	9	1037.16	9.43	136.83
lr105	1377.11	14	69	1377.11	14	1377.11	14.00	70.97
lr106	1252.62	12	87	1252.62	12	1252.63	12.00	81.33
lr107	1111.31	10	287	1111.31	10	1112.26	10.03	95.67
lr108	968.97	9	415	968.97	9	969.02	9.00	102.50
lr109	1239.96	11	348	***1208.96***	11	1233.99	12.00	103.80
lr110	***1159.35***	***10***	547	1165.83	11	1176.20	11.13	142.13
lr111	1108.90	10	179	1108.90	10	1113.60	10.20	105.53
lr112	1003.77	9	638	1003.77	9	1040.34	10.27	168.70
lr201	1263.84	4	193	***1253.23***	4	1260.41	4.20	103.43
lr202	1197.67	3	885	1197.67	3	1255.78	4.03	234.50
lr203	***949.40***	3	1950	952.29	3	962.82	3.00	381.50
lr204	849.05	2	2655	849.05	2	879.01	2.27	486.50
lr205	1054.02	3	585	1054.02	3	1078.84	3.03	166.37
lr206	931.63	3	747	931.63	3	941.67	3.00	259.03
lr207	***903.056***	2	1594	903.60	2	927.58	2.43	418.77
lr208	***734.85***	2	3572	736.00	2	760.95	2.03	531.07
lr209	937.05	3	2773	***932.43***	3	955.96	3.07	236.90
lr210	964.22	3	1482	964.22	3	972.34	3.00	286.13
lr211	927.80	***2***	4204	***888.15***	3	897.03	3.07	478.53
lrc101	1708.80	***14***	119	***1703.21***	15	1704.05	15.00	76.07
lrc102	1563.55	13	152	***1558.07***	***12***	1559.65	12.17	95.53
lrc103	1258.74	11	175	1258.74	11	1260.45	11.00	92.80
lrc104	1128.40	10	202	1128.40	10	1129.32	10.00	108.23
lrc105	1637.62	13	179	1637.62	13	1639.87	13.10	88.17
lrc106	1425.53	11	459	***1424.73***	11	1441.82	11.77	95.63
lrc107	1230.15	11	154	***1230.14***	11	1237.91	11.07	97.73
lrc108	1147.97	10	650	***1147.43***	10	1159.82	10.40	111.67
lrc201	1468.96	4	266	***1407.21***	4	1438.12	4.77	101.47
lrc202	***1374.27***	***3***	987	1385.25	4	1400.11	3.97	161.70
lrc203	***1089.07***	***3***	1605	1093.89	4	1114.40	4.00	268.17
lrc204	827.78	3	3634	***818.66***	3	838.55	3.00	457.30
lrc205	1302.20	4	639	1302.20	4	1305.37	4.00	147.20
lrc206	1162.91	3	445	***1159.03***	3	1176.90	3.63	139.57
lrc207	1424.60	3	607	***1062.05***	3	1070.56	3.03	218.97
lrc208	852.76	3	4106	852.76	3	882.78	3.20	322.13

Nanry and Barnes. When compared to the method of Lau and Liang, the GGA also comes off slightly better. For problem set 2, the GGA proves to be competitive with the method of Li and Lim as well. The time requirements of the GGA are within reasonable limits. However, in some cases, the GGA needs more time than the competing methods. In all, the computational study shows the robust solution behavior of the GGA.

5 Conclusions

In this paper, a Grouping Genetic Algorithm (GGA) for the Pickup and Delivery Problem with Time Windows (PDPTW) has been presented. To our knowledge this is the first time a Genetic Algorithm has been applied to the multi-vehicle PDPTW with hard time window constraints. Furthermore, it appears to be the first published application of the group-oriented genetic encoding of Falkenauer (1998) to a generalization of the Vehicle Routing Problem. The GGA was tested using two publicly available sets of benchmark problems for the PDPTW. The experimental results have demonstrated that the GGA is able to find high quality solutions when compared to previous metaheuristic methods for solving the PDPTW. The overall findings seem to justify the employment of Genetic Algorithms in general, as well as the group-oriented encoding in particular, as suitable techniques for solving the PDPTW.

Future research will be dedicated to the further improvement of the proposed approach. In particular, we would like to extend the embedded insertion heuristic by fast local search techniques, e.g. arc-exchange procedures (Van der Bruggen et al., 1993). This could lead to better results in some cases. Furthermore, the current JAVA implementation of the GGA has not been optimized with regard to time requirements. In order to reduce the computing times, the GGA should be re-implemented in C or C++ using optimized data structures. Finally, the GGA should be tested using larger problem instances. In the real-world, e.g. in the parcel service business, problems with several hundreds of transportation requests have to be solved.

References

Berger J, Salois M, Begin R (1998) A hybrid genetic algorithm for the vehicle routing problem with time windows. In: Mercer RE, Neufeld E (eds) Advances in artificial intelligence – Proceedings of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, pp 114–127. Springer, Berlin Heidelberg New York

Blanton JL, Wainwright RL (1993) Multiple vehicle routing with time and capacity constraints using genetic algorithms. In: Forrest S (ed) Proceedings of the Fifth International Conference on Genetic Algorithms, pp 452–459. Morgan Kaufmann, San Mateo, CA

Bodin L, Sexton T (1986) The multi-vehicle subscriber dial-a-ride problem. TIMS Studies in the Management Sciences 26: 73–86

Davis L (ed) (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York

Desrochers M, Desrosiers J, Solomon, MM (1992) A new optimization algorithm for the Vehicle Routing Problem with Time Windows. Operations Research 40: 342–354

Desrosiers J, Dumas Y, Soumis F (1986) A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with Time Windows. The American Journal of Mathematical und Management Sciences 6: 301–325

Desrosiers J, Dumas Y, Soumis F (1988) The multiple vehicle dial-a-ride problem. In: Daduna JR, Wren A (eds) Computer-aided transit scheduling: Proceedings of the Fourth International Workshop on Computer-Aided Scheduling of Public Transport, pp 15–27. Springer, Berlin Heidelberg New York

Dumas Y, Desrosiers J, Soumis F (1991) The Pickup und Delivery Problem with Time Windows. European Journal of Operational Research 54: 7–22

Falkenauer E (1998) Genetic algorithms und grouping problems. Wiley, Chichester

Gendreau M, Guertin F, Potvin JY, Séguin R (1998) Neighborhood search heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups und Deliveries. Technical Report CRT-98-10, Centre de recherche sur les transports, Université de Montréal, Montréal

Goldberg DE (1989) Genetic algorithms in search, optimization, und machine learning. Addison-Wesley, Reading, MA

Goldberg DE, Korb B, Deb K (1990) Messy genetic algorithms: motivation, analysis, und first results. Complex Systems 3: 493–530

Holland JH (1975) Adaptation in natural und artificial systems – An introductory analysis with applications to biology, control, und artificial intelligence. The University of Michigan Press, Ann Arbor, MI

Homberger J (2000) Verteilt-parallele Metaheuristiken zur Tourenplanung – Lösungsverfahren für das Standardproblem mit Zeitfensterrestriktionen. Deutscher Universitäts-Verlag, Wiesbaden

Ioachim I, Desrosiers J, Dumas Y, Solomon MM, Villeneuve D (1995) A request clustering algorithm for door-to-door handicapped transportation. Transportation Science 29: 63–78

Jaw JJ, Odoni AR, Psaraftis HN, Wilson, NHM (1986) A heuristic algorithm for the Multi-Vehicle Advance Request Dial-A-Ride Problem with Time Windows. Transportation Research Part B 20: 243–257

Jih W-R, Hsu Y-J (1999) Dynamic Vehicle Routing Using Hybrid Genetic Algorithms. In: IEEE Computer Society (ed) Proceedings of the 1999 IEEE International Conference on Robotics & Automation, pp 453–458. IEEE Computer Society, Los Alamitos, CA

Jung S, Haghani A (2000) A Genetic Algorithm for Pick-up and Delivery Problem with Time Windows. In: Transportation Research Record 1733, Transportation Research Board, pp 1–7

Kohl N, Madsen OBG (1997) An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation. Operations Research 45: 395–406

Kopfer H, Pankratz G, Erkens E (1994) Die Entwicklung eines hybriden Genetischen Algorithmus für das Tourenplanungsproblem. OR Spektrum 16: 21–32

Lau HC, Liang Z (2001) Pickup und Delivery with Time Windows, Algorithms und Test Case Generation. In: IEEE Computer Society (ed) Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp 333–340. IEEE Computer Society, Los Alamitos, CA

Lenstra JK, Rinnoy Kan AHG (1981) Complexity of vehicle routing und scheduling problems. Networks 11: 221–227

Li H, Lim A (2001) A metaheuristic for solving the Pickup und Delivery Problem with Time Windows. In: IEEE Computer Society (ed) Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp 160–167. IEEE Computer Society, Los Alamitos, CA

Lin S, Kernighan B (1973) An effective heuristic algorithm for the Traveling Salesman Problem. Operations Research 21: 498–516

Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for a Dial-a-Ride Problem with Time Windows. Multiple Capacities and Multiple Objectives. Annals of Operations Research 60: 193–208

Nanry WP, Barnes JW (2000) Solving the Pickup und Delivery Problem with Time Windows using Reactive Tabu Search. Transportation Research Part B 34: 107–121

Potter T, Bossomaier T (1995) Solving Vehicle Routing Problems with Genetic Algorithms. In: IEEE Computer Society (ed) Proceedings of the 1995 IEEE Interational Conference on Evolutionary Computing, pp 788–793. IEEE Computer Society, Los Alamitos, CA

Potvin JY, Bengio S (1996) The Vehicle Routing Problem with Time Windows – Part II: Genetic Search. INFORMS Journal on Computing 8: 165–172

Psaraftis HN (1980) A dynamic programming solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. Transportation Science 14: 130–154

Psaraftis HN (1983a) An exact algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows. Transportation Science 17: 351–357

Psaraftis HN (1983b) k-Interchange procedures for local search in a Precedence-Constrained Routing Problem. European Journal of Operational Research 13: 391–402

Rayward-Smith VJ (1994) A unified approach to Tabu Search, Simulated Annealing und Genetic Algorithms. In: Unicom Seminars Ltd (ed) Adaptive computing und information processing, Vol I, pp 55–78. Unicom Seminars Ltd, London

Rego C, Roucairol C (1996) A Parallel Tabu Search Algorithm using ejection chains for the Vehicle Routing Problem. In: Osman IH, Kelly, JP (eds) Meta-heuristics: theory and applications, pp 661–675. Kluwer, Norwell et al.

Savelsbergh MWP, Sol M (1995) The General Pickup und Delivery Problem. Transportation Science 29: 17–29

Savelsbergh MWP, Sol M (1998) DRIVE: Dynamic Routing of Independent Vehicles. Operations Research 46: 474–490

Schönberger J, Kopfer H, Mattfeld DC (2003) A combined approach to solve the Pickup und Delivery Selection Problem. In: Leopold-Wildburger U, Rendl F, Wäscher G (eds) Operations Research Proceedings 2002, pp 150–155. Springer, Berlin Heidelberg New York

Solomon M (1987) Algorithms for the Vehicle Routing und Scheduling Problem with Time Window Constraints. Operations Research 35(2): 254–265

Syswerda G (1989) Uniform crossover in genetic algorithms. In: Shaffer JD (ed) Proceedings of the Third International Conference on Genetic Algorithms, pp 2–9. Morgan Kaufmann, San Mateo, CA

Thangiah SR (1995) Vehicle Routing with Time Windows using Genetic Algorithms. In: Chambers L (ed) Practical handbook of genetic algorithms, 2 vol. New frontiers, pp. 253–277. CRC Press, Boca Raton

Toth P, Vigo D (1996) Fast Local Search Algorithms for the Handicapped Persons Transportation Problem. In: Osman ICH, Kelly JP (eds) Meta-heuristics: theory und applications, pp 677-690. Kluwer, Norwell

Van der Bruggen LJJ, Lenstra JK, Schuur PC (1993) Variable depth search for the Single-Vehicle Pickup und Delivery Problem with Time Windows. Transportation Science 27: 298–311

Whitley LD (1989) The GENITOR Algorithm und Selection Pressure: Why rank-based allocation of reproductive trials is best. In: Shaffer JD (ed) Proceedings of the Third International Conference on Genetic Algorithms, pp 116–121. Morgan Kaufmann, San Mateo, CA