[Schö04] Schönberger, J.; Kopfer, H.: Planning the Incorporation of Logistic Service Providers to Fulfill Precedence- and Time Window-Constrained Requests in a Most Profitable Way. In: Fleischmann, B.; Klose, A. (eds.): Distribution Logistics: Advanced Solutions to Practical Problems. Springer, Berlin Heidelberg, 2004, pp. 141-158

Planning the Incorporation of Logistics Service Providers to fulfill Precedence- and Time Window-Constrained Transport Requests in a Most Profitable Way

Jörn Schönberger and Herbert Kopfer

University of Bremen, Department of Economics and Business Administration, Chair of Logistics, Wilhelm-Herbst-Straße 5, 28359 Bremen, Germany, {sberger,kopfer}@logistik.uni-bremen.de

Abstract. We investigate an extension of the Pickup and Delivery Problem with Time Windows (PDPTW) in which only a part of the available request portfolio can be fulfilled by own equipment and the remaining requests are served by subcontracted carriers (Logistic Service Providers). The costs for the fulfillment of the request portfolio result in the sum of travel costs for the self-served requests and the charges paid to the Logistic Service Providers. A Memetic Algorithm (MA) is configured to determine solutions of this combined selection, assignment and sequencing problem. Special genetic operators are proposed to identify those requests that should be served by Logistics Service Providers in order to minimize the overall fulfillment costs. By means of suitable benchmark instances, the capability of the algorithmic approach is assessed and the impacts of the selection feature are analyzed.

1 Introduction

In traditional vehicle routing and scheduling problems routes have to be determined in order to fulfill a given set of requests at costs as low as possible using a fleet of vehicles. However, this approach is not adequate anymore for today operations planning of a carrier company.

Former routing and scheduling problems arose from the distribution of goods of a company maintaining an own fleet. Today, the fleet is out-sourced and operates for several companies. Temporal contracts are made or even single requests are served for a certain charge. On the other hand, the Internet-based request acquiring and offering allows an ad-hoc adjustment of the available request stock, especially by incorporating subcontractor for selected requests.

More general planning approaches are required to exploit these new opportunities for carrier companies (Cruijssen (2003)). Beside generating routes, it has to be decided which subset of requests is served with carrier-owned equipment and which requests are unattractive and should be sub-contracted in order to reduce costs.

In a pickup and delivery problem, goods have to be moved between customer specified locations in order to fulfill certain customer demands. Such a demand is specified within a pickup and delivery request. To reduce the overall travel costs of the considered carrier company, several requests are consolidated into routes. Limited transport resources and time windows that must be incorporated in order to provide an appropriate customer service often hinder the route generation. Requests not served within such a route are fulfilled by another carrier that receives a charge from the original carrier that had acquired the requests.

The partition of the request portfolio into the set of self-served and into the set of subcontractor-served requests has crucial impacts for the overall costs. This article is about the identification of a cost minimal partition of a portfolio of Less-Than-Truckload (LTL) pickup and delivery requests. Section 2 surveys the general possibilities to incorporate the selection feature into a route optimization model. The Profitable Pickup and Delivery Selection Problem (PPDSP), introduced in section 3, describes the general problem of separating self-served requests from those that are served by a subcontractor. In section 4, a genetic algorithm framework is proposed to identify the most profitable portfolio partition and to consolidate simultaneously adequate routes for the self-served requests. Results obtained from numerical experiments are shown in section 5. This article terminates with a conclusion of the main results and provides some hints for future research topics.

2 Model-Based Request Selection

Subject of this section is the compilation of general modeling approaches to establish a model-based selection of the most profitable requests.

The available transport requests are collected in the set $\mathcal{R}$. A *selection* is an ordered pair $\mathcal{S} = (\mathcal{R}^+, \mathcal{R}^-)$ representing the partition of $\mathcal{R}$ into the set $\mathcal{R}^+$ of self-served requests and the set $\mathcal{R}^-$ of the remaining requests, which are typically served by a sub-contracted carrier. It is denoted as *feasible selection*, if the requests in $\mathcal{R}^+$ can be fulfilled by the equipment of the company so that all given constraints are satisfied.

2.1 Determination of Costs and Revenues

Let $C(\mathcal{S})$ represent the costs associated with the execution of routes to serve the requests accepted according to $\mathcal{S}$. The considerable costs $L(\mathcal{S})$ arising from the accepted requests mainly consists of the travel expenses. They are in general depending upon the length of the routes executed by the vehicles within the available fleet. To minimize $L(\mathcal{S})$, the requests in $\mathcal{R}^+$ are consolidated into routes so that existing constraints on the available capacity, visiting times and visiting orders are satisfied. The minimization of $L(\mathcal{S})$ is a routing and scheduling problem. It is referred to as the routing problem.

Requests contained in $\mathcal{R}^-$ are not served by the available fleet. If the rejection of requests is not possible or not recommended then the requests in $\mathcal{R}^-$ are served by an external carrier service that is paid for fulfilling certain transport demands. The paid amounts $F(\mathcal{S})$ are denoted as *carrier service costs* or *charges*. Different problems concerning the determination of a requests selection that separates the self-served requests from those, which are served by carrier services are investigated by Diaby and Ramesh (1995) for a pure distribution, respectively Kopfer and Pankratz (1998), Greb (1998) and Schönberger et al. (2002) for problems with pickup and delivery requests.

If the consolidation of externally served requests into bundles leads to savings due to degressive carrier service charges, a Freight Optimization Problem (FOP) must be solved (e.g. Kopfer (1984)).

The determination of the least costs $C(\mathcal{S}) := L(\mathcal{S}) + F(\mathcal{S})$ for serving a selection $\mathcal{S}$ therefore requires the solving of a combined sequencing and assignment problem (the routing problem) for the accepted requests and the solving of an FOP for the rejected requests (Pankratz (2002)).

The amount of gained revenues $R(\mathcal{S}) := A(\mathcal{S}) + E(\mathcal{S})$ includes the revenues $A(\mathcal{S})$ associated with each self-served request and, in case of a carrier service incorporation, the revenues $E(\mathcal{S})$ of requests served by external carriers. Subcontracting is often not desired and leads therefore to reduced revenues for externally served requests.

The determination of the selection $\mathcal{S}$ remains an open issue. To decide whether a request r belongs to $\mathcal{R}^+$ or $\mathcal{R}^-$, a combined selection (of self-served requests), assignment (of self-served requests to vehicles) and sequencing (of the visiting locations) problem has to be solved. In the following subsections 2.2 - 2.5 general approaches are presented to merge both goals revenue maximization and costs minimization into one mathematical programming decision model. The corresponding literature for LTL pickup and delivery request consolidation is surveyed.

2.2 Cost-Constraint Selection

The realization of routes comprising the most promising requests in $\mathcal{R}^+$ is typically hindered by a budget B_{up} of available amounts of money, time or other consumed resources. As long as the budget is not bailed out, additional requests can be incorporated within the routes. If the budget is exhausted, the routes must be reorganized to fulfill the so far incorporated requests at lower costs.

The goal is then to find a requests selection $\mathcal{S}$ so that the overall achieved benefit is maximized respecting the available budget. In terms of a formal optimization model, the outlined problem can be written as

$$\begin{aligned} & \max R(\mathcal{S}) \\ s.th. \quad & C(\mathcal{S}) \leq B_{up}, \\ & \mathcal{S} \text{ is a feasible selection,} \end{aligned} \tag{1}$$

The problem (1) is a combined request selection and routing problem, constrained by a restricted budget. It represents the problem of selecting the most promising requests if at least one constrained resource is incorporated. This principle of selection is labeled as **cost-constraint selection**. In general, the costs are allowed to be split into several components each restricted by a budget. It must be ensured, that every budget is not exceeded.

Cost-constrained selection is applied to vehicle operations planning if the available transport capacities are scarce and/or tight time windows hinder a sequential request fulfillment. Schönberger and Kopfer (2003) study the impacts of scarce transport capacity in an LTL pickup and delivery scenario.

2.3 Fulfillment Selection

Some vehicle routing problems aim at determining a request selection to satisfy a certain goal, expressed in the least benefit quantity B_{low}. The realizable benefit is typically not known in advance and it cannot be ensured that a certain benefit quantity B_{low} is realizable, especially if the costs for achieving B_{low} are limited by an upper bound. In such a case it is more promising to try to minimize the costs for realizing the predefined goal B_{low}. The problem (2) represents the task to search for a least cost selection $\mathcal{S}$ to satisfy the predetermined goal B_{low}. This selection principle is referred to as **fulfillment selection**.

$$\begin{aligned} & \min C(\mathcal{S}) \\ s.th. \quad & R(\mathcal{S}) \geq B_{low}, \\ & \mathcal{S} \text{ is a feasible selection,} \end{aligned} \tag{2}$$

$$\tag{3}$$

Since the specified budget hinder the visitation of every available location, the requests are separated into those, which are visited and those that are not visited. Often, additional restrictions must be taken into account in order to ensure that all unvisited locations lie within an acceptable distance to the nearest visited location. This is important in order to ensure a reasonable service for unvisited customers (e.g. in case of routing of mobile health care units in developing countries).

2.4 Profit Maximization

If B_{up} is fixed too low, some promising requests have to be ignored. A determination of the goal B_{low} at a too low level also leads to a refusal of promising requests whereas an overestimated goal enforces the incorporation of unprofitable requests.

One possibility to overcome with these deficiencies is to leave both costs and benefits unbounded and to search for a selection, which maximizes the overall profit contribution, defined as the difference between the collected revenues and necessary costs. The task is then to identify those requests, whose incorporations lead to positive contributions to the profit of the considered carrier company, representing its success.

$$\begin{aligned} &\max R(\mathcal{S}) - C(\mathcal{S}) \\ s.th.\ &\mathcal{S} \text{ is a feasible selection.} \end{aligned} \tag{4}$$

The principle of request selection described just above is called **profit maximization**.

The selection of the most profitable pickup and delivery requests is described for the single vehicle case by Verweij and Aardal (2000), whereas Schönberger et al. (2002) investigates the multi-vehicle case.

Frantzeskakis and Powell (1990) and Kleywegt and Papastavrou (1998) investigate the problem of accepting full truckload pickup and delivery requests.

2.5 Multi-Objective Formulations

If the benefits and costs are incompatible and cannot be merged into a single objective, profit maximization is not possible. In such a case it is often first tried to find an auxiliary measurement for the benefit or the costs, which substitutes at least one of the aspects, so that both are again compatible.

A bi-criteria-formulation is necessary if no adequate substitution is possible. The determination of a selection that fulfills both single criteria, benefit maximization and cost minimization, at reasonable levels is aimed at. These selections are situated on the so-called Pareto frontier or efficient frontier. They occupy the property that an improvement of one goal (e.g. costs) requires degradation of the other goal (e.g. benefits) and vice versa. In terms of a mathematical programming formulation, this situation can be formulated as

$$\begin{aligned} &opt\ Z(\mathcal{S}) = (R(\mathcal{S}), \mathcal{C}(\mathcal{S})) \\ s.th.\ &\mathcal{S} \text{ is a feasible selection.} \end{aligned} \tag{5}$$

Solutions of this problem optimize the vector-value objective function Z. This kind of request selection is denoted as **pareto selection**.

Different goals can be combined in bi-criteria request selection models. Promising combinations are: travel duration and travel costs for a travel cost function that increases if travel speed is accelerated, collected requests and travel costs, travel distance and minimal distance to the visited routes.

3 Profitable Pickup And Delivery Selection Problem

The remainder of this article is dedicated to a special LTL pickup and delivery request selection problem. In the following, it is distinguished between carrier-owned vehicles whose routes can be determined and vehicle of subcontracted logistics service providers (LSP). The route of an LSP-vehicle cannot be affected. A freight charge is paid to an LSP for each served request. It is assumed that an LSP is available for each request. A request that is not selected for being served by a carrier-owned vehicle is assigned to an LSP that becomes responsible for its reliable fulfillment. A charge is paid to the LSP. Since all requests are served, the sum $R(\mathcal{S})$ of revenues is fixed independently from the chosen selection $\mathcal{S}$. Therefore, the achieved profit is maximized if the sum of costs (travel costs and freight charges) is minimized.

Combined request selection and route generation problems for less than truckload pickup and delivery requests have received only minor attention so far as seen in the previous section 2.

3.1 Problem Description

Assume a carrier company with m vehicles. The request portfolio $\mathcal{R}$ consists of the available n requests $r_1, \ldots, r_n$. To obtain a maximal profit the most promising requests are consolidated into at most m trips. Each trip is served by exactly one of the available vehicle v with capacity C_v^{max}. Unconsidered requests are given to an LSP, which receives a previously known charge. It is assumed that exactly one LSP is available for each request, so that it is not necessary to select between different LSP-charges.

Every customer request r_i is specified by the triple (PU_i, DL_i, c_i). The pickup activity PU_i takes place at p_i^+ whereas the delivery activity DL_i is demanded at location p_i^-. A time window $[t_{min}, t_{max}]$ is specified for each activity. Load of volume c_i is to be picked up at p_i^+ and to be delivered to p_i^-.

An operation is a triple $\pi := (p, a(p), e(p))$, where p represents the location of a pickup, a delivery, a start or a stop activity. The expression $a(p)$ refers to the determined arrival time of the vehicle at location p and $e(p)$ denotes the leaving time from p. If the vehicle arrives at p before the associated time window has been opened it has to wait until the earliest allowed operation time $t_{min}(\pi)$ for π has passed. The leaving time of π has to precede the latest allowed operation time $t_{max}(\pi)$.

A sequence of operations $\Pi = (\pi_1^\Pi, \ldots, \pi_{n_\Pi}^\Pi)$ is called a route. In the remainder of this article the first component of π_i is denoted by p_i. The route Π includes N_Π requests. The initial starting operation and the final terminating operation are not stored within the route for the sake of simplification.

Let $t_{p_i,p_{i+1}}$ be the travel time between p_i and p_{i+1}. The arrival and the leaving times are calculated recursively: $a(p_1) = e(p_1) := 0$ and $a(p_i) := e(p_{i-1}) + t_{p_{i-1},p_i}$ $(i > 1)$.

The leaving time is achieved by $e(p_i) := \max\{a(p_i), t_{min}(\pi_i)\}$.

The vector $\delta^\Pi = (\delta_1^\Pi, \ldots, \delta_{N_\Pi}^\Pi)$ describes the volumes that are collected along the route Π. For a pickup operation at p_i it is $\delta_i \geq 0$ and for the associated delivery operation at p_j we define $\delta_j := -\delta_i$. The capacity usage along Π is determined recursively: $\omega_1^\Pi := 0$ and $\omega_i^\Pi := \omega_{i-1}^\Pi + \delta_{i-1}$ $(i > 1)$.

The route Π is called pd-path for vehicle v if it holds the following restrictions (cf. Savelsbergh and Sol (1995)). Either both operations of request r or none of them are contained in Π (*pairing*), a pickup operation precedes its associated delivery operation (*precedence*), the maximal load is not exceeded for all i: $\omega_i^\Pi \leq C_v^{max}$ (*capacity*) and the leaving time for operation π_i lies in the specified time window: $t_{min}(p_i) \leq e(p_i) \leq t_{max}(p_i)$ (*time window*).

The set Π^* contains the requests incorporated into the route Π.

A pd-schedule Ω is a set of pd-paths $\Pi_1, \ldots, \Pi_m$ so that each customer request is assigned to at most one of these paths.

Filling $\mathcal{R}^+$ with the requests contained in one of the pd-paths $\Pi_1, \ldots, \Pi_m$ and $\mathcal{R}^-$ with the remaining requests, $\mathcal{S} := (\mathcal{R}^+, \mathcal{R}^-)$ is a request selection. The costs associated to a pd-path only depend upon the traveled distance. For each driven distance unit, one monetary unit is spent. The expression $C(\Pi)$ represents the costs for executing the pd-path Π and $F(\mathcal{R}^-)$ denotes the freight charges paid for the sub-contracted requests that forms the set $\mathcal{R}^-$.

The PPDSP can now be formulated as the mathematical optimization problem

$$\min F(\mathcal{R} \setminus \bigcup_{i=1}^{m} \Pi_i^*) + \sum_{i=1}^{M} C(\Pi_i), \tag{6}$$

$$s.th.\ (\Pi_1, \ldots, \Pi_m) \text{ is a pd-schedule.} \tag{7}$$

3.2 Generation of Benchmark Instances

A set of benchmark instances for the PPDSP is generated adopting an idea found in Nanry and Barnes (2000). The main concept is to derive instances for the Pickup and Delivery Problem with Time Windows (PDPTW) from optimal or near optimal solutions of the famous Solomon instances for the Vehicle Routing Problem with Time Windows (VRPTW) (cf. Solomon (1987)). Additionally, an adequate freight charge is assigned to each generated request that has to be paid if this request is completed by an LSP.

Customer locations are paired randomly within the routes of the considered solution to obtain pickup and delivery requests. The first visited location becomes the pickup location whereas the remaining one becomes the delivery place (Fig. 1). The demand at the selected pickup location becomes the volume to be moved between the pickup and the delivery location.

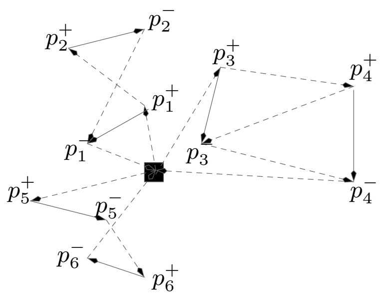

Fig. 1. Generation of a benchmark instance: Derivation of six pickup and delivery requests (solid arcs) from three tours (dashed arcs).

The freight charge q_{r_i} that represents the price for the incorporation of an LSP to complete request r_i is determined with regard to the demanded distance $d(p_i^+, p_i^-)$ between the pickup and the delivery location of this request. Consider the available solution of the Solomon instance. Let v_* be the index of the vehicle serving r, D_{v_*} be the sum of distances between the pickup and the delivery location of the requests assigned to v_* and let L_{v_*} be the driven distance of v_*. For each request r served by v_* the tariff coefficient $\bar{m}$ is set to $\bar{m} := L_{v_*}/D_{v_*}$ and the freight charge is defined as $q_{r_i} := \bar{m} \cdot d(p_i^+, p_i^-)$. It is assumed that the available transport capacities are not scarce. In this case a competition among requests does not take place. Furthermore, the travel costs are only affected marginally by the moved quantities. Greb (1998) proposes a freight tariff that can be applied if the capacities are scarce.

The used solution of the PDPTW-instance is evaluated as a PPDSP solution calculating the overall profit contribution. The achieved value serves as a reference.

Benchmark instances are derived from 18 Solomon VRPTW instances and their solutions. Problems with tight time windows and scattered (R1), semi-scattered (RC1) and clustered customer locations (C1) are considered as well as problems in which the associated time windows are more relaxed (R2, RC2 and C2). From each class, three problems are selected. The pairing is seeded by $\sigma \in \mathcal{G} = \{0, 1, 2\}$.

With the probability α, the LSP-tariff q_r of a certain request r is modified. The updated q_r is given by $q_r := (1-\beta) \cdot q_r$. If $\beta > 0$, then a discounted tariff is applied otherwise a surcharge has to be paid. Instances $I(\phi, \sigma, \alpha, \beta)$ are generated for the probabilities $\alpha = 0.5, 0.75, 1$ and the discount/surcharge values $\beta = -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75$. Overall, $18 \cdot 3 \cdot 3 \cdot 7 = 1134$ different instances are generated.

The capacity of the incorporated vehicles is set to the sufficiently large value of 300 capacity units.

4 Genetic Search Approach

Evolutionary algorithms and especially Genetic Algorithms have been rarely investigated in the context of solving pickup and delivery type problems. This is caused by a missing problem representation and appropriate operators that allow the simultaneous coding of the assignment, the sequencing and, for PPDSP-type problems, the selection decisions.

Pankratz (2002) proposes a Grouping Genetic Algorithm (Falkenauer (1999)) in which only the assignment of requests to vehicles is left for evolutionary improvement. The routes are constructed then by specialized insertion heuristics.

The applicability of Genetic Algorithms to highly constrained complex optimization problems is often hindered by infeasibility problems. The evolved individuals violate one or more of the problem inherent constraints. To overcome this problem, the combination of GAs with powerful special heuristics to produce feasible solution instances is proposed (Radcliffe and Surry (1994)). The so-called Memetic Algorithms (MAs) often outperform the pure genetic search but, above all, they enable the application of genetic search because they ensure the feasibility of the maintained and evolved individuals.

The so far observed success of MAs motivates the configuration of an MA to tackle the PPDSP. Tentative routes that comply with the pairing and the precedence conditions are obtained from the evolutionary process. They are transformed into the pd-paths removing constraint violating requests. These requests are assigned to an LSP.

4.1 Problem Representation

Encoding A pd-schedule Ω consists of m pd-paths $\Pi_1, \ldots, \Pi_m$ and a collection of rejected requests. Let T_{Π_i} denote the sequence of operations executed by vehicle i. The dummy route T_+ of length N_+ includes the sub-contracted requests. To achieve a string representation $G(\Omega)$ of Ω the sequences are subsequently written in a vector, followed by the dummy route, the lengths of the sequences and of the dummy route and the vector $(D_1, \ldots, D_m)$ of selected termination depots. Vehicle i finishes its operations at the depot number D_i. The representation of Ω is then given by

$$G(\Omega) := (\underbrace{T_{\Pi_1}, \ldots, T_{\Pi_m}, T_+}_{\text{route segment}}, \underbrace{2N_{\Pi_1}, \ldots, 2N_{\Pi_m}, N_+}_{\text{number segment}}, \underbrace{D_1, \ldots, D_m}_{\text{depot segment}}). \quad (8)$$

If requests are shifted between pd-paths or between a pd-path and the dummy route, the length of $G(\Omega)$ does not vary. The length of the route segment is $2n$; the number segment has the length m+1 whereas the depot segment consists of m components. Altogether, the pd-schedule is coded in a string of length $2(n+m)+1$.

Decoding To decode a string representation into a pd-schedule it is proceeded as follows. Tentative routes are derived from the string. Such a route holds the pairing and the precedence constraints but not necessarily the capacity or time window conditions. Initially, the number N_1 of requests assigned to vehicle one is derived from the string representation. The first N_1 pickup operations and the corresponding delivery locations in the route segment form the tentative route of vehicle number one. The tentative route of the second vehicle consists of the next N_2 pickup and its corresponding delivery operations executed in the order determined in the string representation. The tentative routes for the remaining vehicles are simultaneously generated.

In the second step, the tentative routes are converted into pd-paths. First, a modified 2-opt improvement procedure is applied in order to reduce the travel distance and travel time and hence to decrease the number of too-late-arrivals. Afterwards, all requests that cause a capacity or time window constraint violation are successively shifted from the tentative routes into the dummy route and remain unconsidered in the current pd-schedule. The obtained routes now fulfill also the capacity and time window constraints, thus they are pd-paths. The order in which the requests are checked is externally determined. Finally, the 2-opt procedure is re-applied in order to achieve additional travel distance savings. The second step is referred to as repair step in the remainder of this article.

4.2 Operators

Initial Population To generate an initial population of pd-schedules, the parallel path construction heuristic described in detail in Schönberger and Kopfer (2003) is applied. This procedure is parameterized by a permutation of the available requests and by a permutation of the vehicles. It is applied with different request and vehicle permutations to obtain a diversified population.

Crossover The crossover operator derives a new pd-schedule (offspring), coded in a string representation, from two parental pd-schedule representations $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. First experiments with a syntactical crossover and a syntactical mutation operator (cf. Schönberger et al.(2002)) have not led to convincing results. Therefore, problem specific knowledge is used for generating new pd-schedules.

Initially, the string representations of the parental pd-schedules are split into the routes for the vehicles and into the dummy route. For each individual, $m+1$ routes are then available (some of them are contingently empty).

Offspring routes are generated successively for each vehicle. The parental routes of the vehicle i are denoted as r_i^1 and r_i^2 with l_i^1 and l_i^2 included locations. Let δ_i^{12} be the number of stops that are included in both routes. The offspring route r_i^{off} of length $l_i^{off} := l_i^1 + l_i^2 - \delta_i^{12}$ is initialized. To fill this route it is proceeded as followed. A binary string $\boldsymbol{b} = (b_1, \ldots, b_{l_i^{off}})$ is generated at random. A '0' in the l-th position indicates, that the l-th stop is taken from r_i^1, in case of $b_l = 1$ the next stop included in the offspring route is taken from r_i^2. The probability to select '0' for an arbitrary position in $\boldsymbol{b}$ is set to $\frac{l_i^1 - \delta_i^{12}/2}{l_i^{off}}$, that is the relative length of r_i^1 with regard to r_i^{off}. Starting from $l = 0$ each position of r_i^{off} is successively filled, distinguishing three cases:

1. There are unconsidered stops in both parental routes; Set the l^{th} stop in r_i^{off} to the first so far unconsidered stop in r_i^1 ($b_l = 0$) or r_i^2 ($b_l = 1$). These stops are labeled as considered in both parental routes. It is continued with the next stop ($l := l + 1$).
2. If r_i^1 contains no more unconsidered stops, the remaining stops in the offspring route are filled with the so far unconsidered stops from r_i^2.
3. If r_i^2 contains no more unconsidered stops, the remaining stops in the offspring route are filled with the so far unconsidered stops from r_i^1.

This crossover operator produces a new path that fulfills the pairing and the precedence constraint. Additionally, it does not destroy sequences appearing in both parental routes. A precedence relation of two locations that is included only in one parental route, maybe r_i^1, survives with the probability $\frac{l_i^1 - \delta_i^{12}/2}{l_i^{off}}$.

Each request included into an offspring route is labeled as used and cannot be considered for another subsequently generated offspring route. This ensures that no request is served more than once.

After determining all offspring routes (including the route of the so far unassigned requests) they are stored into the chromosome representation and the number segment is updated. Finally the terminating points of the routes are merged applying a uniform crossover operator.

Mutation With a certain probability each offspring is affected by mutation. Three slight changes are performed.

1. The termination point of an arbitrarily selected route is replaced at random.
2. Within a randomly selected route (including the dummy route) an arbitrarily selected location is re-positioned at random. The precedence feasibility is not violated.
3. A request is moved from the dummy route to the route of an arbitrarily selected vehicle. Therefore, a randomly selected request is deleted from

the dummy route and inserted in the route of the determined vehicle at random, so that the precedence feasibility is preserved.

Feature number 3 is the counterpart of the repair step. The latter one removes requests from the current schedule whereas the former feature injects additional requests into the current schedule.

Determination of the Fitness After the application of the repair step all individuals are feasible with respect to the pairing, the precedence, the capacity and the time windows constraints and no customer requests is assigned to more than one vehicle. A suitable fitness value is then obtained by the objective value determined according to (6).

Selection A $\mu + \lambda$ scheme (Baeck (2000)) is used to derive a new population of individuals. In a first step a temporal population is filled with the offspring individuals generated from the parental individuals from the original population. Then the individuals from the original and from the temporal population are sorted in one list by decreasing fitness values. The best individuals within the list form the new population. It substitutes the original population. The population size remains unchanged throughout the generations.

5 Computational Experiments

Several numerical experiments have been performed in order to asses the capability of the proposed genetic search framework and in order to analyze the impacts of varying LSP charges.

5.1 Algorithmic Setup

The Memetic Algorithm (MA) evolves a population of 100 individuals. The initial population is seeded by randomly generated request permutations. While generating the request sequences it is ensured, that requests with surcharge tariffs q_i have a higher probability to be selected for the first components of the sequences (biased request permutations). The parameterization of the construction heuristic with these permutations leads to a diversified initial population. The available fleet is homogeneous, so that it is not necessary to determine different vehicle permutations. The MA generates a sequence of 200 populations each containing 100 individuals. Computational experiments have shown that a crossover probability of 1.0 and a mutation frequency of 0.5 produce the averagely best results. The repair step applied to each constraint violating offspring is parameterized by a biased request permutation, so that requests for which LSP incorporation is most expensive, are checked first.

5.2 Results

Numerical experiments are performed for each instance of the benchmark field. Since the MA is a randomized procedure average results are taken from three independent runs applied to each of the 1134 instances. Therefore, overall 3402 instances are evaluated.

Table 1 shows the averagely obtained results for the six problem classes if the freight charge is neither discounted nor enlarged ($\alpha = 0$). The first line represents the results compared to the reference objective values. Only for problems in the R2 class, the averagely observed results are lying above the reference values. For all other problem classes, the MA is able to identify reduced cost solutions by incorporating an LSP. The largest improvements are observed for C1 problems in which a cost reduction of 16% is realized. The second line represents the percentages of requests that are fulfilled by LSPs. This percentage increases significantly if the spreading of the pickup and delivery locations decreases. In the third row, the percentage of customer locations at which the serving vehicle can execute the corresponding pickup or delivery operation without waiting time for the opening of the corresponding time window is shown. This value tends to decrease if the spreading of the customer locations is reduced. For problems with completely scattered customer locations (R1 and R2) and for problems with clustered locations, the number of no-wait-operations reduces if the time windows are relaxed. This phenomenon is not observed for problems with semi-scattered locations.

Table 1. Results obtained for different problem classes without discounts or surcharges ($\alpha = 0$, $\beta = 0$).

group	R1	R2	RC1	RC2	C1	C2
relative costs	0.98	1.03	0.98	0.91	0.84	1.00
LSP-served	0.19	0.15	0.21	0.27	0.33	0.39
no-wait-service	0.70	0.68	0.65	0.73	0.63	0.50

Table 2 shows the results obtained for experiments with diversified LSP-charges. In all tables, the divergences from the case without discounted or surcharged values ($\beta = 0$) are shown. The tabular on top shows the variation of the sum of costs for a medium diversification ($\alpha = 0.5$), the middle tabular represents the results from the $\alpha = 0.75$ experiment whereas the last tabular contains the results observed for the complete variation experiment ($\alpha = 1.0$).

Two main observations can be stated. If the surcharge is increased then the overall costs also increase and if the LSP-charge discount is enlarged then the savings also increase. Secondly, if the frequency of surcharge-requests is increased (increase of α) then the additional costs also increase. The achieved savings significantly increase if the frequency of discounted LSP-charges is increased.

The largest savings are realized for problems with relaxed time windows (R2, RC2 and C2). Their savings are significantly larger than the savings observed for the problems with tight time windows. This is mainly caused by a reduced number of self-served requests and therefore by an intensified exploitation of the discounted LSP-charges (cf. Table 3).

If the frequency of expensive LSP-charges is enlarged then additional costs are observed. If the extra charge is too large then the MA recognizes that the corresponding requests can be served by own equipment in a cheaper fashion. However, the savings by not using the LSP is so large that it is not necessary to insert these requests in the most profitable way in the existing routes. For this reason, significant additional costs are observed for very large surcharges.

Table 2. Overall costs for the request fulfillment

	medium variation ($\alpha = 0.5$)						
	surcharge ($-\beta$)				discount (β)		
	0.75	0.50	0.25	0.00	0.25	0.50	0.75
avg. R1	0.08	0.04	0.04	0.00	-0.05	-0.14	-0.27
avg. R2	0.13	0.12	0.03	0.00	-0.07	-0.20	-0.35
avg. RC1	0.12	0.08	0.06	0.00	0.01	-0.09	-0.22
avg. RC2	0.11	0.10	0.01	0.00	-0.10	-0.22	-0.35
avg. C1	0.09	0.05	0.01	0.00	-0.06	-0.10	-0.24
avg. C2	-0.01	0.07	0.06	0.00	-0.04	-0.14	-0.27
	medium variation ($\alpha = 0.75$)						
	surcharge ($-\beta$)				discount (β)		
	0.75	0.50	0.25	0.00	0.25	0.50	0.75
avg. R1	0.09	0.08	0.05	0.00	-0.08	-0.21	-0.36
avg. R2	0.14	0.12	0.10	0.00	-0.08	-0.25	-0.47
avg. RC1	0.12	0.10	0.06	0.00	-0.03	-0.13	-0.29
avg. RC2	0.12	0.08	0.03	0.00	-0.13	-0.29	-0.46
avg. C1	0.13	0.06	0.01	0.00	-0.10	-0.19	-0.32
avg. C2	0.08	0.03	-0.01	0.00	-0.06	-0.22	-0.39
	complete variation ($\alpha = 1.0$)						
	surcharge ($-\beta$)				discount (β)		
	0.75	0.50	0.25	0.00	0.25	0.50	0.75
avg. R1	0.14	0.09	0.06	0.00	-0.08	-0.24	-0.43
avg. R2	0.23	0.15	0.09	0.00	-0.12	-0.35	-0.61
avg. RC1	0.16	0.08	0.08	0.00	-0.04	-0.14	-0.32
avg. RC2	0.14	0.15	0.05	0.00	-0.16	-0.38	-0.63
avg. C1	0.15	0.08	0.03	0.00	-0.08	-0.19	-0.35
avg. C2	0.09	0.07	-0.00	0.00	-0.09	-0.26	-0.50

Table 3. Self-served requests ($\alpha = 1.0$ case).

	complete variation ($\alpha = 1.0$)						
	surcharge				discount		
	0.75	0.50	0.25	0.00	0.25	0.50	0.75
avg. R1	0.15	0.14	0.06	0.00	-0.22	-0.49	-0.64
avg. R2	0.25	0.26	0.12	0.00	-0.44	-0.81	-0.85
avg. RC1	0.09	0.10	0.06	0.00	-0.08	-0.23	-0.54
avg. RC2	0.67	0.41	0.38	0.00	-0.59	-0.77	-0.82
avg. C1	0.07	0.06	0.02	0.00	-0.14	-0.29	-0.53
avg. C2	0.32	0.24	0.34	0.00	-0.39	-0.62	-0.78

6 Conclusions and Future Works

We addressed the partition of a portfolio of pickup and delivery requests into a set of self-served and into a set of externalized requests. Three one-modal modeling approaches are found: selection caused by limited budgets, least input fulfillment and profit maximization. The latter one is appropriate if no given budget or goal is specified.

The Profitable Pickup and Delivery Selection Problem represents an adequate problem formulation of a combined route generation and request selection striving for the minimization of the request fulfillment costs. The proposed Memetic Algorithm achieves reasonable and comprehensible results.

An interesting problem occurs if online-instances of the PPDSP must be handled. These problems are subject of our current research activities.

References

Bäck, T. Fogel, D.B., Michalewicz, Z. (2000): Evolutionary Computation 1 – Basic Algorithms and Operators. Institute of Physics Publishing, Bristol and Philadelphia

Cruijssen, F. (2003): A survey on European inter-organizational data sharing implementations in transportation. internal report, CentER Applied Research, Tilburg University

Diaby, M., Ramesh, R. (1995): The Distribution Problem with Carrier Service: A Dual Based Penalty Approach. ORSA Journal on Computing **7(1)** 24–35

Falkenauer, E. (1999): Genetic Algorithms and Grouping Problems. Wiley, Chichester

Frantzeskakis, L.F., Powell, W.B. (1990): A Successive Linear Approximation Procedure for Stochastic, Dynamic Vehicle Allocation Problems. Transportation Science **24(1)** 40–57

Greb, T. (1998): Interaktive Tourenplanung mit Tabu-Search. Dissertation University of Bremen

Kleywegt, A.J., Papastavrou, J.D. (1998): Acceptance and Dispatching Policies for a Distribution Problem. Transportation Science **32** 127–141

Kopfer, H. (1984): Lösung des Frachtoptimierungsproblems im Gewerblichen Güterfernverkehr. Dissertation Universität Bremen

Kopfer, H., Pankratz, G. (1998): Das Groupage-Problem kooperierender Verkehrsträger. In Kall, P. and Lüthi H.-J. (eds.) Proceedings of Operations Research 1998, 453–462

Nanry, B., Barnes, J.W. (2000): Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research Part B **34** 107–121

Pankratz, G. (2002): Dynamische speditionelle Transportdisposition unter besonderer Berücksichtigung der Fremdvergabe. DUV, Gabler Ed. Wissenschaft, Wiesbaden

Radcliffe, N.J., Surry, P.D. (1994): Formal Memetic Algorithms. In Fogarty, T.C. (ed) Evolutionary Computing 1–16

Savelsbergh, M.W.P., Sol, M. (1995): The General Pickup and Delivery Problem. Transportation Science **29** 17–29

Schönberger, J., Kopfer, H. (2003): A Memetic Algorithm for a Pickup and Delivery Selection Problem. In Mohammadian, M. (ed.) Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation, published on CD

Schönberger, J., Kopfer, H., Mattfeld, D. (2002): A Combined Approach to Solve the Pickup and Delivery Selection Problem. In Leopold-Wildburger, U., Rendl, F., Wäscher, G. (eds) Proceedings of Operations Research 2002, 150-155

Solomon, M.M. (1987): Algorithms for the vehicle routing and scheduling problem with time window constraints Operations Research **35(2)**, 254–265

Verweij, B., Aardal, K. (2000): The Merchant Subtour Problem. Research Report UU-CS-2000-35, Utrecht University

Williams, H.P. (1999): Model Building in Mathematical Programming. Wiley, Chichester, fourth ed.