Computers & Operations Research 36 (2009) 1740 – 1750

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Online decision making and automatic decision model adaptation☆

Jörn Schönberger*, Herbert Kopfer

University of Bremen, Chair of Logistics, Wilhelm-Herbst-Straße 5, 28359 Bremen, Germany

ARTICLE INFO

Available online 1 May 2008

Keywords:
Transportation
Vehicle routing
Online optimization
Adaptation
Dynamic decision problem
Supply chain

ABSTRACT

The paper investigates an online version of the vehicle routing problem with time windows, in which additionally arriving requests cause the revision of so far followed routes and schedules. An extended online optimization framework is proposed, which automatically adapts to problem variations and enables the explicit consideration of up-to-date knowledge about the current performance of the controlled system. Actually, we use the mean punctuality observed in the transportation system to adjust the objective function utilized for solving the next decision problem instance. The search is guided toward least cost solutions coming along with high punctuality. We prove the applicability of this approach within comprehensive numerical experiments.

1. Introduction

The supply chain of a product describes the sequence of activities to be carried out in order to create the desired output from one or several inputs factors. Supply chain planning (SCP) aims at achieving the highest possible efficiency of a supply chain by coordinating and consolidating the necessary material flows ("processes'') so that economies of scale are exploited to the largest possible extent.

Recent trends in the management of supply chains compromise the successful application of existing concepts for computational planning support.

- The ability to consider unexpected events in an ad hoc fashion is propagated as significant competitive advantage. The continuous incorporation of recent problem data requires a continuous plan revision.
- The partners forming the supply chain are not willing to give up the responsibility and self-reliance for the material flow decisions in their part of a supply chain. Consequently, the centralized supply chain wide top-down material flow determination and the goals of the incorporated partners are sometimes contradicting.

Contracts between supply chain partners are fixed for several months and must consider both the responsiveness of the involved partners to dynamics (e.g. demand variations and peaks) and the partners' autonomy in the operational deployment planning. Although a supply chain is built by independent partners, one of them, the supply chain coordinator is dedicated and entitled to persuade the independent partners to behave and act in the sense of the superior supply chain goals instead of the subordinate partner's aims.

Computer-supported decision making is necessary for all supply chain partners. The definition of a suitable mathematical decision model is a prerequisite for the successful application of automatic decision making tools like optimization algorithms. However, the fine-tuning of such a model is a sophisticated task that typically requires some trial-and-error runs in order to identify the best parameter setting. Solving a concatenated sequence of decision problem instances is referred to as online optimization. The definition and the solving of a new instance are triggered by events that compromise the realization of the so far optimal solution. There is no time to experiment on the right parameters for the decision model of the new problem. Here, the right parameters have to be adjusted automatically.

Within this article, we investigate the impacts of different configurations for the interaction between the supply chain coordinator and a transport-providing partner in a given supply chain. We analyze the implications of different intervention rights that enable the coordinator to bias the planning decisions of the transport partner by adjusting relevant decision model parameters. We show that a performance-oriented adaptation of the transport partner's decision logic has positive impacts on the overall supply chain reliability. An extension of the well-established online decision making framework is proposed. It enables a planning system of the coordinator to

☆ This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes'' (Subproject B7).

* Corresponding author. Fax: +49 421 218 4271.
E-mail addresses: jsb@uni-bremen.de (J. Schönberger), kopfer@uni-bremen.de (H. Kopfer).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.04.009

detect supply chain performance variations and to implement autonomously the necessary decision model adaptations in the planning system of the subordinate transport partner. Numerical simulation experiments are reported in order to assess the proposed extension.

Section 2 introduces the investigated problem. Section 3 reports the results of numerical simulations using the pure online decision making framework without performance feedback. Section 4 presents the concept for automatic decision model adaptation. Section 5 reports results of comprehensive numerical experiments.

2. Vehicle deployment planning problem with uncertain demand

We introduce the investigated supply chain setting in this section. Related scientific literature is compiled in Section 2.1. The supply chain layout is described in Section 2.2. Three configurations for the interaction between the coordinator and a transport partner are described and motivated in Section 2.3. A model for the deployment problem of the transport partner is presented in Section 2.4. The derivation of artificial testcases used in the simulation experiments is explained in Section 2.5.

2.1. Literature

A major strategic (long-term) decision task in SCP is the composition of a portfolio of adequate partners cooperating in a value creation project. Some partners are selected from this portfolio to agree mid-term contracts describing the rights and responsibilities of each partner for the management of a single supply chain project. A typical topic is the agreement about the provided resources and capacities as well as the performance and refunding [1]. In the short-term planning, the next completed tasks are selected, tasks are assigned to resources (persons or machines) and their processing sequences are determined for each resource.

Fleischmann et al. [1] identify the ability to handle uncertainties as the most important competency in the planning of logistic activities. Two generic methodological approaches for dealing with decision problems associated with known or expected incompleteness and inaccuracy of data are mentioned in the scientific literature. A *single deciding strategy* derives decisions merging the incomplete data ignoring that additional data might become known later on. If a decision is made it is tried to keep the necessary modifications of the once made decisions as small as possible. A typical representative in transportation planning is given by the a priori route concept [2]. An *online solving strategy* allows and emphasizes the revision of (all) previously made decisions with the goal to derive the best decision for the actual problem. Consequently, a sequence of decision problems (a sequence of instances) has to be solved consecutively. Such a sequence is called a dynamic decision problem [3]. In rolling horizon planning [4], the revision times are fixed and known in advance but in an event-oriented planning [5], a decision revision cycle in triggered by pre-specified events whose occurrence times cannot be anticipated. In online traveling salesman problems [6], in online vehicle routing problems [7] and in pickup and delivery problems [8], uncertainty is typically caused by the nescience about future transport requests.

The research on dynamic decision problems and their management with online solution strategies follows two directions. First, research on competitive analysis [9,10] investigates algorithmic aspects of decision revision. In particular, algorithms are searched that keep the additional computational efforts caused by the need for re-planning on a low level. Second, the modeling of the dynamic decision problems is a focal point of the research. Here, the incorporation of additional data into the formal decision models reflecting the changed decision situation is addressed. Static rules for updating the so far used model instance are investigated in [9,10]. Wasserburger [11] proposes to define the next instance with respect to the current time of day.

A performance-based re-adjustment of a formal problem representation in order to adapt automatically the representation to the changed real world equivalence has received only minor attention so far. Arnold et al. [12] survey corresponding approaches for optimizing a compiler on a virtual machine and Šegviċ et al. [13] apply a reformulation approach in specific software to retrieve information from pictures (computer vision). Gutenschwager et al. [14] and Gutenschwager [15] propose a concept, which enables the definition of a new (and potentially revised) objective function to be used only for representing and solving the next instance of a dynamic planning problem.

2.2. Transport request fulfillment in a supply chain

Fig. 1 outlines the order fulfillment in a supply chain and demonstrates the role of each partner. Customers express their demand in terms of external orders submitted to the supply chain coordinator. This coordinator receives the external customer orders and overtakes the responsibility for their reliable fulfillment [16,17]. The coordinator splits each customer order into the necessary internal purchasing, production, distribution (transport), and retailing tasks. Tasks associated with different customer orders are combined into internal purchasing, production and transport requests. Then, each department involved in the supply chain is instructed to execute the specified requests according to their competencies in order to contribute to the fulfillment of the customer orders.

The supply chain including the coordinator as well as each department is considered as a group of agents who cooperatively fulfills a set of tasks (the external customer orders). None of the agents is able to fulfill the complete tasks without the support of the others. The coordinator agent has no knowledge, resources and abilities, to setup and execute the material flow processes, while neither the purchasing agent nor the production agent nor the distribution agent is able to acquire customer orders.

In this contribution we investigate formal methods to support the interactions between the supply chain coordinator and the partner who is providing distribution services (transport partner). The coordinator generates streams of internal transport requests from customer demands and the distribution partner has to ensure that the required physical movements are executed respecting service guidelines previously agreed in a cooperation contract. Special attention is dedicated to maintain a high reliability level even if spontaneous workload peaks appear.

2.3. Compared supply chain configurations

The main task of the supply chain coordinator is to ensure that the cooperative actions of the independent partners in the supply chain accord with superior supply chain wide goals stated in the directives for the cooperation. In this investigation, the achievement and maintaining of a least transport process punctuality rate of p^{target} is agreed as the transport partner's contribution to the overall supply chain wide goals. Typically, a punctuality rate $p^{\text{target}} < 1$ is agreed because the hedging of the complete risk caused by the uncertainty of future demand causes extremely increasing marginal costs [18].

As long as this threshold is achieved, the transport partner might aspire its own benefit without explicitly considering any supply chain wide service quality goals. Uncontrollable external events, especially workload peaks, endanger the achievement of the agreed service level. As soon as the observed punctuality rate runs into danger to fall below the specified threshold or if it even has fallen below

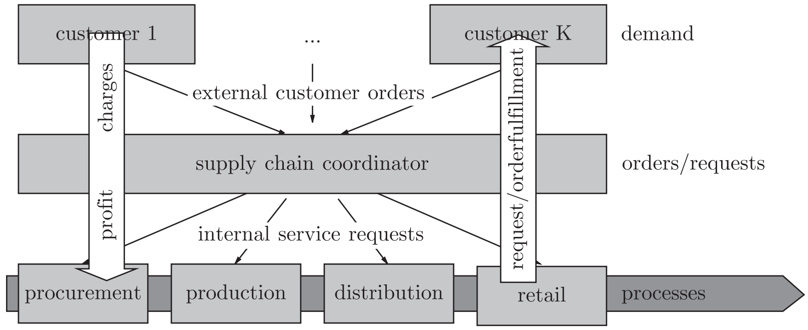

Fig. 1. Customer transport demand fulfillment in a supply chain.

then a coordinator intervention into the deployment planning of the transport partner becomes necessary.

Depending on the agreed contracts, the coordinator is provided with different opportunities to intervene into the deployment decisions of the transport partner in order to ensure the achievement of the supply chain wide goals. In order to analyze the impact of different configurations of the interactions between the supply chain coordinator and the transport partner, we distinguish three different configurations. Every configuration represents a strategy used by the coordinator to maintain or immediately recover the punctuality rate p^{target} of the transport processes even in extreme workload peak situations.

2.3.1. Reference configuration

The direct way to ensure the achievement of the desired least punctuality rate is to force the service provider to generate processes fulfilling the least punctuality requirement. Every process proposal which does not obey the least punctuality condition and which leads to a lower punctuality will be rejected by the coordinator. The minimization of costs is only addressed as a second rank desire but not as mandatory planning requirement. The achievement of the least punctuality is the superior planning goal. We refer to this configuration as the hard condition configuration (HARD-configuration) of the investigated supply chain scenario. The decision task of the transport partner is a covering problem.

The least punctuality rate has been agreed between the coordinator and the transport-providing partner. Therefore, an average workload has been assumed while deriving the agreed service quality. A significant increase in the number of customer sites (workload peak) augments the process costs and therefore lowers the profit of the transport partner. It is quite unfair that the additional expenses are not shared with the coordinator (and thereby among all supply chain partners). Thus, the strict enforcement of the least punctuality discriminates the transport partner and enforces him to leave the cooperation as soon as possible in order to prevent serious financial damage. For this reason, the HARD configuration is not realistic. We use it as reference configuration to provide comparable results for simulation experiments with sufficiently high profitability rates. These reference results enable an estimation of the costs necessary for ensuring the achievement of the service goal.

2.3.2. Penalization of late arrivals

Actually, the coordinator must provide incentives to each partner in order to act in the sense of the common goals instead of acting only in the sense of its own interest. For each partner, the main motivation to contribute to the supply chain is to maintain or increase the own profit. Vice versa, the strive for a profit maximization enables the supply chain controller to influence and regulate the behavior of a partner. The partner gets a higher benefit if it acts in accordance with the common supply chain wide goals but its profit is reduced if the partner acts contrarious.

The supply chain coordinator receives charges paid by the customers for the fulfillment of the customer orders. Using the sum of earned charges, budgets are funded that are used to cover the material flow process costs specified by the service center agents. In order to stimulate a partner agent to determine processes of highest efficiency, the difference between the budget and the process costs remains in the service center as its gain (profit). The main idea of the penalty configuration (PEN-configuration) is to penalize the transport partner for each request whose on-site fulfillment starts with delay. Thereby, this partner is motivated to fulfill as much requests as possible on time so that the punctuality rate p^{target} can be guaranteed. If a demand peak occurs then the service-providing partner can freely decide whether to accept the profit reduction or to spend more efforts to maintain or even increase the service level. Here, the negative impacts of a workload peak are shared between the coordinator (and therefore among all partners) and the transport partner: The latter pays penalties for late arrivals but the supply chain consortium accepts a temporarily reduced punctuality.

2.3.3. Adaptive accounting schemes

The partner responsible for the transport decides about the transport request fulfillment. High quality (HQ) services (express courier or individual same-day delivery) are quite reliable but very costly. On the other hand, standardized request execution processes are cheaper due to the realization of economies of scale but cannot fulfill individual requirements. In case that the punctuality rate is higher than p^{target} the transport partner's expenses are refunded similarly for each fulfillment mode (individual service and consolidated transport) and accounted to the budget designated for covering the transport costs. However, if the punctuality rate is at risk to fall below p^{target} or has even fallen below this threshold, then expenses for the reliable services are reimbursed at a higher percentage or even completely but expenses for the cheap and unreliable transport services are only partly covered.

An accounting scheme describes how expenses of a subordinate agent are accounted to the given budget. The main idea of the accounting scheme adaptation is to define a rule that determines the refunding of the transport partner only taking into account its reliability and not its actually incurred costs. The accounting scheme is adapted to the currently observed punctuality rate. Consequently, if the transport partner's performance varies the rule for refunding the transport partner's expenses also varies.

Each participating agent decides independently about the planning of its processes (resource deployment, etc.) but the coordinator agent (as the superior agent) modifies the accounting scheme in order to make the exploitation of expensive express services more attractive for the subordinate agent because the additional expenses are not or only partly charged to his budget. Consequently, the process determination carried out by the subordinate agent is biased by the coordinator agent by means of the accounting scheme variation. This forces the subordinate agents to adopt its process decisions to the guidelines of the superior coordination agent.

The application of an accounting scheme in a process revision adapts the decision making process of the transport partner (the benefit of a specific decision depends upon the currently applied accounting scheme). We refer to this strategy as Search Direction ADaptation configuration (SDAD-configuration).

2.4. The transport partner's planning problem

The coordinator receives customer demand continuously over time. Every Δt time units he generates internal requests from the customer demand and forwards the requests to the transport partner. The transport partner has to incorporate the additional requests into the so far followed transport processes. The process-planning problem of each service partner is therefore a dynamic decision problem, which is solved in online fashion, e.g. a process revision is carried out in event-driven fashion in response to the additionally submitted requests. Consequently, a sequence of concatenated decision problems is stated. Each instance is formulated as an optimization model. Solving such a model means to find the most profitable process decisions for the transport operations. Each instance represents a generalized common vehicle routing problem with time windows [19]. It is the multi-vehicle version of the traveling repairman problem [20], which is additionally extended by subcontraction. The transport partner's dynamic decision problem has been previously formulated and investigated in [21].

2.4.1. Subcontracting

The transport partner compiles and schedules requests in routes executed by the vehicles belonging to its own available fleet $\mathscr{V}$ (self-fulfillment by guided vehicles [21]). However, requests that do not fit into the routes of the guided vehicles are forwarded to a logistics service provider (subcontraction) [22,23]. The logistic service provider (LSP) is a trustful partner of the transport partner and is paid for the reliable fulfillment of the subcontracted requests. It receives a certain previously known amount of payment for this service and ensures the reliable fulfillment of these requests (similarly to an individual express service request fulfillment). A subcontracted request remains unconsidered while constructing the routes for the own vehicles. If a request has been subcontracted then this decision cannot be revised later while solving forthcoming decision problem instances.

2.4.2. Uncertain demand

Only a subset of all requests is known to the planning authority at the time when the decision concerning subcontracting is made and the routes for the own vehicles are compiled. The planning authority decides about subcontracting or self-fulfillment of a request as soon as it becomes known. A release of one or more additional requests initiates the revision of the so far constructed routes for the own vehicles. If needed, some requests so far planned for self-fulfillment are excluded from the routes of the own vehicles and forwarded to an LSP. The arrival times at some customer sites may be postponed in order to serve one or several additional customers earlier by the same vehicle. Furthermore, the number of additionally released requests temporarily increases unpredictably, so that workload peaks occur from time to time.

2.4.3. Soft time windows

Lateness at a customer site is possible but causes penalty costs. Let f_t' be the number of requests fulfilled (completed) during the interval from $t-t^-$ until t and let f_t'' be the number of requests whose completion time is later than t but not later the $t+t^+$. Although a particular request is allowed to be late, it is required that the portion p^{target} of the $f_t := f_t' + f_t''$ requests is served timely. Let f_t^{comp} be the number of the requests completed timely within the last t^- time units and let f_t^{expec} be the number of punctually scheduled requests within the next t^+ time units, then $p_t := (f_t^{\text{comp}} + f_t^{\text{expec}})/f_t \geqslant p^{\text{target}}$ has to be achieved. In a *HQ period* the requirement for the least punctuality is fulfilled ($p_t \geqslant p^{\text{target}}$) but in a *low quality* (*LQ*) *period* the required punctuality is not attained anymore ($p_t < p^{\text{target}}$).

A transportation plan describes how the known requests are fulfilled. Subsequently arriving requests are accepted and handled by updating the existing transportation plan. A sequence of transportation plans $TP_0, TP_1, TP_2, \ldots$ is generated reactively at the ex ante unknown update times $t_0, t_1, t_2, \ldots$ and each single transportation plan is executed as long as it is not updated.

A single request r attains consecutively different states. Initially, r is known but it is not yet scheduled (K). Then, r is assigned to an own vehicle (I, short for internal fulfillment) or subcontracted (E, short for externalization). If the operation at the corresponding customer site has already been started but not yet been finished the state S (short for started request) is assigned to r. The set $R^+(t_i)$ is composed of additional requests released at time t_i. Requests completed after the last transportation plan update at time t_{i-1} are stored in the set $R^C(t_{i-1}, t_i)$. The new request stock $R(t_i)$ is determined by $R(t_i) := R(t_{i-1}) \cup R^+(t_i) \backslash R^C(t_{i-1}, t_i)$. Each request belongs at each time to exactly one of the sets $R^K(t_i)$, $R^E(t_i)$, $R^I(t_i)$ or $R^S(t_i)$, in which the requests having a common state are collected.

The problem of updating a transportation plan at time t_i is as follows. Let V denote the set of all own vehicles, $P_v(t_i)$ the set of all paths (sequence of visiting sites beginning with the position of the vehicle at time t_i and ending with the central depot) executable by vehicle v in TP_i and let $P(t_i)$ denote the union of the sets $P_v(t_i)$ ($v \in V$). If the request r is served in a path p then the binary parameter a_{rp} is set to 1, otherwise it is set to 0. A request r, already known at time t_{i-1} that is not subcontracted in TP_{i-1} is served by vehicle v_r. The travel costs associated with path p are denoted as $C^1(p)$. Finally, $C^3(r)$ gives the subcontracting costs of request r.

In order to code the necessary decisions for determining a transportation plan, we deploy two families of binary decision variables. Let $x_{pv} = 1$ if and only if path $p \in P(t_i)$ is selected for vehicle $v \in V$ and let $y_r = 1$ if and only if request r is subcontracted.

$$\sum_{p \in P(t_i)} \sum_{v \in \mathscr{V}} C^1(p) x_{pv} + \sum_{r \in R(t_i)} C^3(r) y_r \to \min \tag{1}$$

$$\sum_{p \in P_v(t_i)} x_{pv} = 1 \quad \forall v \in \mathscr{V} \tag{2}$$

$$x_{pv} = 0 \quad \forall v \in \mathscr{V}, \ \ p \notin P_v(t_i) \tag{3}$$

$$y_r + \sum_{p \in P(t_i)} \sum_{v \in \mathscr{V}} a_{rp} x_{pv} = 1 \quad \forall r \in R(t_i) \tag{4}$$

$$y_r = 1 \quad \forall r \in R^E(t_i) \tag{5}$$

$$\sum_{p \in P_{v(r)}} a_{rp} x_{pv_r} = 1 \quad \forall r \in R^S(t_i) \tag{6}$$

$$p_t \geqslant p^{\text{target}} \tag{7}$$

For the HARD-configuration the process-planning problem is represented by the mathematical optimization model (1)–(7). The costs for TP_i are minimized (1). One route is selected for each vehicle (2) and vehicle v is able to execute the selected path p (3). Each single request known at time t_i is either served by a selected vehicle or forwarded to the LSP (4) but a once subcontracted request cannot be re-inserted into the paths of the own vehicles (5). An (S)-labeled request cannot be re-assigned to another vehicle or LSP (6) and overall, the percentage p^{target} of all requests must be scheduled in time (7).

The model (1)–(7) is NP-hard to solve since it represents the traveling salesman problem in a specific parameter setting.

$$\sum_{p\in P(t_i)}\sum_{v\in\mathcal{V}}(C^1(p)+C^2(p))x_{pv}+\sum_{r\in R(t_i)}C^3(r)y_r\rightarrow\min \tag{8}$$

In the PEN-configuration the punctuality constraint (7) is skipped and the objective function (1) is replaced by the evaluation function (8) that incorporates the penalty payments $C^2(p)$ for lateness. Penalties associated with p are summed up to $C^2(p)$ from all late customer site visits according to p. If the request is performed in time, the penalty is zero for the associated single customer site, it increases proportionally up to 25 monetary units for a delay of 100 time units. Further delays do not lead to additional charges.

$$\lambda_t^a\cdot\sum_{p\in P(t_i)}\sum_{v\in\mathcal{V}}(C^1(p)+C^2(p))x_{pv}+\lambda_t^b\sum_{r\in R(t_i)}C^3(r)y_r\rightarrow\min \tag{9}$$

In the model of the SDAD-configuration the constraint (7) is skipped and (9) replaces (1). Since the two coefficients λ_t^a and λ_t^b are re-calculated before the decision model of instance TP_i (2)–(6), (9) is defined, the ordered pair $(\lambda_t^a,\lambda_t^b)$ represents an adaptive accounting scheme.

If λ_t^a is increased relatively to λ_t^b then the fulfillment of a request with own vehicles becomes less attractive than the subcontraction of this request. In case that λ_t^a is reduced relatively to λ_t^b the attractivity of self-fulfillment increases (because the refunding of the associated cost grows). However, (9) does not represent the actually occurred costs but this objective function allows the adaptation of the decision preferences of the transport partner to the current punctuality. In case that the punctuality is (too) low, the usage of the more profitable subcontraction opportunities will support the re-increase of p_t.

2.5. Test cases

The construction of artificial test cases from the Solomon instances [19] is described in [24]. An incoming stream of successively arriving requests is simulated in these scenarios and the additional requests have to be served by self-fulfillment or subcontraction. Demand peaks interrupt balanced streams of incoming requests and lead to significant changes in the decision situations.

The costs for self-fulfillment are normalized to one monetary unit for each traveled distance unit. The amount of α monetary units has to be paid to the LSP for each subcontracted distance unit. Each subcontracted request r causes overall costs of $F_r := C^3(r) = \alpha\cdot d(dep,\iota_r)$ monetary units calculated by multiplying the distance $d(dep,\iota_r)$ between the LSP depot dep and the customer site location ι_r of r with α. Four different request streams are generated from the Solomon instances $\{R103, R104, R107, R108\}$ for different tariff levels $\alpha\in\{1, 1.25, 1.5, 1.75, 2, 3\}$. Each test case simulates requests from the demand released from $t_0=0$ until $T_{\max}=5000$. Additional requests are released every $\Delta t=100$ time units.

3. Online planning in HARD- and PEN-configurations

This section reports the experimental setting and findings for the HARD- and the PEN-configurations. In Section 3.1, we describe the configuration of a hybrid search algorithm for solving the instances $TP_0, TP_1,\ldots$. Section 3.2 compares techniques for ensuring the considerations of the constraints (2)–(7) during the solving of $SP(t_i)$. Section 3.3 describes the experimental setup. The achieved results are presented and discussed in Section 3.4.

3.1. Memetic Algorithm for the schedule generation

We use a Memetic Algorithm (MA) realizing a hybrid search strategy consisting of a global genetic search space sampling and a local 2-opt improvement procedure for solving the scheduling model instances $SP(t_0), SP(t_1),\ldots$ of the online decision problem introduced in Section 2.4.

The genetic search uses a $\mu+\lambda$-population model evolved by the application of the PPSX-crossover-operator [23] and a mutation operator that (a) arbitrarily switches fulfillment modes of requests, (b) shifts requests between selected routes of own vehicles and (c) reverses the visiting order of randomly chosen subsequences of arbitrarily selected routes.

The construction of the initial population is generated using the Push Forward Insertion Heuristic [19]. One half of the initial set of solution proposals is generated by deploying the heuristic followed by some random proposal modifications and the other half is generated purely at random without applying any biasing procedure. The evolution process is stopped dynamically if the average fitness of the evolved population does not improve for 10 generations.

Every time a new decision model instance $SP(t_i)$ has arrived, the MA is re-started to solve the model of the recent instance. Computational experiments, in which parts of the final population of the last instance solved are used to seed the initial population of the recent instance, failed because this initial population leads to rapid convergence on a bad level even if the crossover and mutation probability are determined adaptively. An analysis of the population development has shown that the significantly changed decision situation requires the re-initialization of the genetic material so that the new decision aspects are considered explicitly. For this reason, a complete new initial population is formed using the seeding approach described above.

3.2. Constraint handling techniques

Solutions of the models introduced in Section 2.4 are defined as sets of decision variables, instantiated by a value taken from their associated domains. Local search (improvement) algorithms first generate an initial instantiation of the decision variables. Then they apply one or more search operators and generate an offspring solution from one or more existing solutions. The sequence of generated solutions is called the search trajectory and the algorithm generates one (e.g. Tabu Search) or several trajectories in parallel (e.g. Genetic Algorithms). It is necessary that the generated solutions comply with the constraints (2)–(7) (feasibility). From a certain solution on, all further solutions within the search trajectory have to comply with the constraints specified in the model. Three different approaches enforcing the search trajectory to stay in the set of feasible solutions are presented below.

3.2.1. Selection of an adequate solution representation and of suitable operators

The first idea is to design a problem representation, which only allows solution proposals that comply with all given constraints. If all operators can only generate solutions within the given representation then all maintained and generated solutions stay compatible with the associated constraints.

We use a direct problem route-based representation [23], which ensures that no violations of the constraints (2)–(6) occur in the maintained set of solutions.

(1) All requests r associated to a too-late customer site arrival are collectd in the set S_1.
(2) For each request $r \in S_1$ the savings s_1 are calculated. Here, s_1 is defined as the difiierence between the travel cost savings and the subcontracting costs F_r.
(3) If S_1 is empty or if the current punctuality rate p_{t_i} is at least p^{target} then goto (5). Otherwise, the requests contained in S_1 are sorted, so that the first request in the order has the maximal saving of all requests in S_1, the second request in the order has the second highest saving and so on.
(4) Finally, the fulfillment mode of the first request in the sorted list is switched to subcontraction, the request is deleted from the list and the punctuality rate p_{t_i} is updated. Goto (2)
(5) The repair has been completed.

Fig. 2. Procedure REPAIR().

3.2.2. Repairing constraint violations

Local hill-climbers are incorporated into the superior memetic search algorithm. They repair constraint violations by modifying the generated offspring solutions and transform them to the nearest solution that is feasible with respect to the constraints to be considered.

The application of the MA to the HARD-configuration requires the call of a repair procedure for each generated offspring solution if the percentage of in-time arrivals is smaller than p^{target}. In this case the procedure REPAIR() shown in Fig. 2 is executed for each offspring solution.

In general (but not in the problem investigated here) it is unclear in advance whether a repair procedure call can completely repair a given solution using the given repair function. For this reason, the repair attempt is a search process itself. Its goal is to identify the nearest solution, which complies with the given constraints. The computational effort for this (maybe unsuccessful) search is often quite high.

3.2.3. Penalization of constraint violations

Solutions that propose late customer site arrivals are penalized by depreciating their evaluation value. The penalization lowers the attractivity of such an individual and decreases the individual's chance of being used as source of an offspring solution. It is expected that, on the long run, penalized individuals will not be used any more so that at the end, only solutions without any constraint violations form the search trajectories.

In order to generate preferentially solution proposals that maintain a very high punctuality rate we penalize each delayed customer site arrival of a vehicle. Therefore, we deploy the piecewise-linear penalty function introduced in Section 2.4 in the evaluation function (8). This MA realization is applied if the supply chain scenario is PEN-configured.

3.3. Experimental setup

The HARD-configuration of the supply chain setting deploys the decision model (1)–(7) and the MA incorporating the procedure REPAIR(). Similarly, the PEN-configuration uses the decision model (2)–(6), (8) and the MA incorporating the penalty fitness function to derive new transportation plans.

In order to assess the performance of the two supply chain configurations, we perform several simulation experiments using the artificial test cases introduced in Section 2.5. The target punctuality p^{target} is set to 0.8.

An experiment (α, exp) is defined by the combination of the tariff level $\alpha \in \{1, 1.25, 1.5, 1.75, 2, 3\}$ and the assumed supply chain configuration $exp \in \{PEN, HARD)\}$. We simulate each scenario (α, exp, P, ω) in three independent runs seeded by $\omega \in \{1, 2, 3\}$. From the Solomon instances $P \in \mathscr{P}\{R103, R104, R107, R108\}$ we derive the set P of consecutively released requests. Overall, we have defined $6 \times 2 = 12$ experiments leading to $12 \times 4 \times 3 = 144$ simulated scenarios.

The punctuality recorded at time t within the scenario (α, exp, P, ω) is denoted as $p_t(\alpha, exp, P, \omega)$. Let

$$p_t(\alpha, exp) := \frac{1}{12} \sum_{\omega=1}^{3} \sum_{P \in \mathscr{P}} p_t(\alpha, exp, P, \omega)$$

denote the average punctuality in experiment (α, exp) observed at time t.

In order to study the impact of the demand peak on the punctuality, we calculate the deviation of $p_t(\alpha, exp, P, \omega)$ from the reference value $p_{1000}(\alpha, exp, P, \omega)$ for all times in the observation time interval $[1000, 5000]$ by

$$p_t(\alpha, exp, P, \omega)/p_{1000}(\alpha, exp, P, \omega) - 1.$$

The largest past-peak deviation from the reference value is then calculated by $min_{t \geqslant 1500}\{p_t(\alpha, exp, P, \omega)\}/p_{1000}(\alpha, exp, P, \omega) - 1$. The average of the largest observed deviation from the reference values in the scenarios of the experiment (α, exp) is then given by

$$\delta(\alpha, exp) := \frac{1}{12} \sum_{\omega=1}^{3} \sum_{P \in \mathscr{P}} \left(\frac{\min_{t \geqslant 1500}\{p_t(\alpha, exp, P, \omega)\}}{p_{1000}(\alpha, exp, P, \omega)} - 1 \right).$$

Let $T^{\text{below}}_{\alpha, exp}$ denote the first time in which $p_t(\alpha, exp)$ falls below p^{target} and $T^{\text{heal}}_{\alpha, exp} := \min\{t \in [1000, 5000] \mid \nexists\, l \in [t, 5000], p_l < p^{\text{target}}\}$ referring to the time in which an HQ state is finally re-achieved by $p_t(\alpha, exp)$. We define

$$\pi(\alpha, exp) := \frac{T^{\text{heal}}_{\alpha, exp} - T^{\text{below}}_{\alpha, exp}}{4000}$$

as the percentage of LQ periods within the observation interval $[1000, 5000]$.

Throughout the simulation time, we have recorded the percentage of subcontracted requests in $q_t(\alpha, exp, P, \omega)$. These values have been summarized in

$$q_t(\alpha, exp) := \frac{1}{12} \sum_{P \in \mathscr{P}} \sum_{\omega=1}^{3} q_t(\alpha, exp, P, \omega)$$

for each experiment (α, exp). The maximally observed subcontraction rate is defined by $\sigma(\alpha, exp)$, calculated by

$$\sigma(\alpha, exp) := \max_{t \geqslant 1500} q_t(\alpha, exp).$$

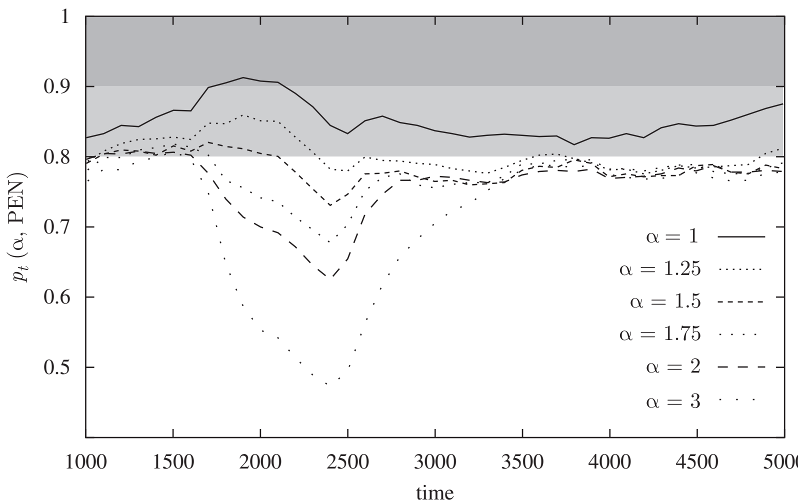

Fig. 3. Development of the punctuality $p_t(\alpha, PEN)$.

Table 1
Maximal punctuality deviation $\delta(\alpha, exp)$.

exp	α					
	1	1.25	1.5	1.75	2	3
HARD (%)	3.6	−0.0	5.3	4.4	4.2	3.5
PEN (%)	−1.0	−1.8	−7.8	−13.6	−22.0	−38.8

Table 2
Portion $\pi(\alpha, PEN)$ of low quality situations.

exp	α					
	1	1.25	1.5	1.75	2	3
PEN (%)	–	60.0	70.0	97.5	82.5	97.5

Table 3
Maximal externalization rate $\sigma(\alpha, exp)$.

exp	α					
	1	1.25	1.5	1.75	2	3
HARD (%)	22.2	14.9	10.0	8.0	7.2	8.0
PEN (%)	21.4	15.5	10.0	5.8	5.1	4.1

It indicates the exploitation of the subcontraction fulfillment mode. Finally, we trace the costs $C(\alpha, exp, P, \omega)$ for the request fulfillment and calculate the deviation $\gamma(\alpha, PEN, P, \omega) := C(\alpha, PEN, P, \omega)/C(\alpha, HARD), P, \omega) - 1$ of the costs from the reference value observed in the *HARD*) experiment. The average deviation $\gamma(\alpha, PEN)$ for this experiment (α, exp) is calculated.

3.4. Numerical assessment of HARD- and PEN-configurations

The HARD-configuration outperforms the PEN-configuration. Clearly, by definition, it is $\pi(\alpha, HARD)) = 0$ for all investigated tariff levels α. In addition, the HARD-configuration is able to slightly enlarge the punctuality compared to the referential value at time $t = 1000$ (Table 1). The additional knapsack constraint allows the memetic search to evaluate different separations of the request portfolio into self-fulfilled and subcontracted requests. Since the constraint (7) ensures that at least 80% of the requests are in time, no penalty costs contradict the route composition.

The PEN-configuration can guarantee the 80% punctuality only for comparable tariff levels $(\alpha = 1)$ as shown in Fig. 3. Just after the demand peak is over $(t \geqslant 1800)$ the punctuality even increases because the routes of the own vehicles are compiled from a larger number of available requests. So, a higher number of matching requests can be found. As soon as the tariff level α increases, subcontracting becomes more and more unattractive. Its costs are higher than the travel costs for the own vehicles plus penalty payments. The self-fulfillment mode is preferred although it does not come up with an in-time service. Actually, decreases of p_t below 50% are observed for $\alpha = 3$. Consequently, the maximal decreasing rate of p_t falls from −1.0% $(\alpha = 1)$ down to 28% $(\alpha = 3)$ as it can be seen in Table 1. Due to the parameter sensitivity of the punctuality $p_t(\alpha, PEN)$ to tariff level increases, the percentage $\pi(\alpha, PEN)$ of LQ-situations increases from $\pi(0, PEN) = 0$ up to $\pi(3, PEN) = 97.5\%$ (Table 2). Therefore, a short-time demand peak has a long-lasting negative impact on the punctuality of the service.

The maximal rate of subcontracted requests decreases with increasing tariff level α (Table 3). For comparable freight tariffs $(\alpha \leqslant 1.5)$ both configurations behave similarly with respect to the subcontraction of requests. The maximal externalization rate $\sigma(\alpha, exp)$ is nearly the same in both cases for each $\alpha \leqslant 1.5$. However, if the tariff levels climb further then $\sigma(\alpha, HARD))$ remains stable at ≈ 8% while $\sigma(\alpha, PEN)$ falls further down to ≈ 4%.

The main reason for the bad performance of the PEN-configuration is its non-observance of the reliable subcontracting services if its costs are significantly higher than the sum of self-fulfillment costs and penalty payments. For $\alpha = 1$ the maximal percentage of subcontracted requests is $\sigma(1, PEN) = 21.4\%$ but for large tariff levels, this portion is significantly reduced down to $\sigma(3, PEN) = 4.1\%$ (Table 3).

The temporal increase of the number of additional requests entering the considered logistic system from 50 up to 150 for the duration of 200 time units leads to a significant increase of waiting requests. If the tariff level is quite high $(\alpha = 3)$ then the non-consideration of the subcontracting services causes a blockage of the request fulfillment. While in off-peak situations the average number of pending requests is 70, this number escalates up to nearly 230 in the HARD-configuration and even up to 300 in the PEN-configuration. In the first mentioned configuration, the number of 70 waiting requests is re-achieved at time 2000 but in the latter one the limit of nearly 70 waiting requests is reached not before time 3000.

Overall, the PEN-configured supply chain setting is not able to maintain the target punctuality if the tariff levels for request subcontraction are lifted. The penalization of delayed arrivals does not ensure the target punctuality in situations with a non-comparable tariff level. The search for least cost transportation plans ignores the selection of request subcontraction if the sum of costs for self-fulfillment and lateness is less than the freight charge to be paid to the LSP. In previous simulation experiments [25] we have shown that the punctuality in the PEN-configuration can be slightly increased for $\alpha = 3$ cases if the maximal penalty value is increased. However, the observed additional process costs are much too high to justify the slight reliability increase. Improved results are only considered in singular experiments. The determination of an adequate penalty value requires many tests and is therefore not appropriate to be applied in an online approach to a problem in which the scarceness of resources varies significantly over time.

The application of both configurations leads to nearly the same costs. There is no empirical evidence that one of the configurations generally produces a larger amount of process realization costs. We observe the following cumulated costs: $\gamma(1, PEN) = -8.7\%$, $\gamma(1.25, PEN) = -5.5\%$, $\gamma(1.5, PEN) = 0.1\%$, $\gamma(1.75, PEN) = 4.9\%$, $\gamma(2, PEN) = 5.0\%$ and $\gamma(3, PEN) = -1.0\%$. The decisions observed in the PEN-configuration yield less costs than for the HARD-configuration if $\alpha = 1$, 1.25, 1.5 or 3. In the remaining cases, the HARD-configured supply chain setting produces a sequence of solutions causing fewer costs than the PEN-configured consortium.

4. Image modification approaches

PEN-configured as well as HARD-configured supply chain settings produce transportation plans causing nearly the same costs. Nevertheless, the two configurations show a noticeable difference in the quality of the generated transportation plans if the tariff level α climbs up. In this case a HARD-configuration performs significantly better with respect to the reliability of request fulfillment. First, the HARD-configuration produces less severe punctuality decreases (Table 1). Second, the duration of LQ service periods is significantly reduced compared to the PEN-configuration (Table 2). The conclusion of these observations is that the performance of the HARD-configured logistic system (with respect to the service goal) remains unaffected by a tariff level variation but the reliability of the transportation plans generated in the PEN-configured systems depends on the tariff level height. The PEN-configuration is quite more sensitive to variations of the problem to be tackled than the HARD-configuration. Although the performance of the HARD-configuration convinces with respect to efficiency as well as the service goal, its application is not possible because the partners forming the supply chain under consideration are not treated fair in case of demand peaks.

The situation is quite different with respect to the PEN-configuration, where the negative impacts of a workload peak are shared among the partners of the supply chain. However, the penalization scheme must fit to the current system state and performance. In the investigated problem at hand, as soon as the search space is modified, then the evaluation scheme requires an adaptation, which turns out to be a re-definition of the used formal optimization model. This section describes approaches for a model-based rescheduling, which automatically and explicitly maps the variation of the problem severeness into the used formal decision model. This mechanism is called *image modification* [26] because it affects the representation of the real world problem. Actually, it manipulates the formal decision model for the current decision problem instance. Here, we refine the idea of Gutenschwager [15] and automatically define a new objective function for each new problem instance.

4.1. Static, dynamic and adaptive modeling rules

To meet the least punctuality requirement, the PEN- and the HARD-configuration deploy different mechanisms for enforcing the search algorithm (imitating the rational search behavior of the transport partner) to select transportation plans in which at least p^{target} percent of all requests are served in time. As soon as the new model TP_i is built (by updating the so far used model TP_{i-1}), these features are considered explicitly: determining the maximal penalty value (PEN-configuration) or specifying the constraint (7) exploited in the HARD-configuration. The determination of the corresponding parameters is done in advance before the first instance is stated independently from the current time and from the current system performance or current system workload. Such a rule for defining components of the optimization model for the next decision instance is called a *static rule*. The PEN-configuration as well as the HARD-configuration represent realizations of static model definition rules.

Since static rules are defined a priori, it cannot be guaranteed that they remain suitable after a significant change of the decision problem. In this investigation, the PEN-configuration works well as long as the system load remains on its initial level. In such a situation, the used penalty value is large enough to depreciate tardy visits so that detours or even subcontracting are preferred. However, if the system load increases during and after the demand peak, the pre-determined penalty value is now too low to enforce the subcontraction of late requests. The penalty and the travel costs have to get a higher impact than the subcontracting costs in order to let the subcontracting mode become the preferred completion mode so that additional requests are subcontracted (and served in time).

A rule that distinguishes several temporal phases and which acts differently in these phases is called a *dynamic model definition rule*, if the intervals to be distinguished as well as the modeling tasks to be carried out, are known at time t_0. We cannot apply any dynamic rule to the problem investigated in this contribution because we do not know the intervals in advance, in which the system load is too high.

Instead of determining a priori how a modeling rule will behave and evolve, it is more promising to decide reactively how to parameterize a decision model in response to a change of the performance of the controlled system. This adaptation allows the consideration of latest and recent problem variations for the definition of the next model instance. Model definition rules that exploit contextual data are referred to as *adaptive model definition rules*.

The original online decision making framework [9] is neither capable to detect changes in the considered problem requiring an adjustment nor is it equipped to implement the necessary model modifications. In order to overcome these deficiencies, we propose and test an extension of the online decision making framework.

4.2. Algorithmic model adaptation

The adaptation of a formal decision model to a changed real world problem setting initially requires the recognition of the problem variation. In order to prepare an automatic detection of an HQ or LQ period, we first specify the intended system development starting from the current time t_i. Therefore, we select N indicators that map the performance of the considered logistic system at a time t into the N-tuple $(i_1(t), \ldots, i_N(t))$ of real values (the system's state at time t). Let Im_u denote the set of possible values for the indicator i_u : $t \mapsto i_u(t)$. Furthermore, the set $\mathscr{F}(t) \subseteq Im_1 \times \cdots \times Im_N$ is defined to contain exactly all those system states that fulfill the HQ period property. The set $\mathscr{D}(t_i) := [t_i; \infty) \times \mathscr{F}(t_i)$ contains all future states of the system that have a punctuality $p_t \geqslant p^{\text{target}}$. It is called the *system development corridor at time t_i*.

The system development corridor for the problem introduced in Section 2 is defined as follows. We use the only indicator p_t, $Im_1 :=$

(1) $i=0$;
(2) t_i:=current_time();
(3) M_i:=generate_initial_model();
(4) TP_i:=solve(M_i,t_i);
(5) wait until re-planning becomes necessary;
(6) $i=i+1$;
(7) t_i:=current_time();
(8) p_{t_i}:=calculate_current_punctuality();
(9) determine current model adaptation intensityh(p_{t_i});
(10) M_i:=$M_{i-1} \oplus H(t_i, h(p_{t_i}))$;
(11) TP_i:=solve(M_i,t_i);
(12) if ($t_i \leq T_{max}$) then goto (5);
(13) stop;

Fig. 4. Basic re-planning procedure with decision model adaptation.

$[p^{\text{target}}; 1]$ and set $\mathscr{F}(t) := [p^{\text{target}}; 1]$. The corridor $\mathscr{D}(t_i)$ is then given by $\mathscr{D}(t_i) := [t_i; \infty) \times [p^{\text{target}}; 1]$. The gray shaded area in Fig. 3 represents this system development corridor.

Countermeasures maintaining a sufficiently high punctuality should be established before the system's performance leaves the system development corridor. In order to be able to start the necessary actions as early as possible, we define the *core* $\mathscr{C}(t_i) \subseteq \mathscr{D}(t_i)$ of the system development corridor as $\mathscr{C}(t_i) := [t_i; \infty) \times [p^{\text{target}} + 0.1; 1]$. The core $\mathscr{C}(t_i)$ serves as a reference that is used to decide whether model adaptations are required or not. The dark grey shaded area in Fig. 3 represents the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$.

The intensity of the model adaptation is determined by measuring the distance of the current system state from the core of the system development corridor. If this distance is zero then no model modifications are required. If the distance is small, then slight modifications of the formal problem representation (model) are to be established but if the distance is large then a significant re-definition of the so far used model is necessary. A function h mapping a system state to the (real) value expressing the model modification severity, is called the *intensity function*.

We propose the following function h as intensity function. It is defined according to the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$ and is 0 as long as $p_t \geqslant p^{\text{target}} + 0.1$ (HQ period, $(t, p(t)) \in \mathscr{C}(t)$), $h(p_t) = 1$ if $p_t \leqslant p^{\text{target}} - 0.1$ (LQ period, $(t, p_t) \notin \mathscr{D}(t)$) and it decreases linearly from 1 down to 0 if p_t increases from $p^{\text{target}} - 0.1$ up to $p^{\text{target}} + 0.1$ (transition phase).

The *implementation function H* describes the model modifications to be implemented depending on the current time t, the system performance at this time and the intervention intensity expressed by the current intensity function value.

Fig. 4 shows the online decision making framework extended by the model adaptation feature exploiting the intensity function as well as the implementation function. At first, the iteration counter i is initialized (1) and the current time t_i is fetched (2). Then, the initial decision model M_0 is formulated (3) and solved afterwards (4). The solution of M_0 is the transportation plan TP_0 whose execution starts immediately. Whenever an update of the transportation plan becomes necessary, a new model is stated, adapted and solved afterwards. The loop consisting of the instructions (5)–(12) in the basic algorithm represents this iteration. The so far existing transportation plan TP_i is executed as long as no update is necessary (5). In case that a plan revision is started the iteration counter i is increased by 1 (6), the current time t_i is saved (7) and the current system performance is checked by determining the system's current punctuality p_{t_i} (8). Next, $h(p_{t_i})$ delivers the model adaptation intensity (9). Then, the new model instance M_i is derived from the last used instance M_{i-1} by instantiating ("$\oplus$") the necessary modifications given by the implementation function H (10). Now, the recent model is solved and the new transportation plan TP_i becomes the process to be followed (11). Finally, it is checked whether the update procedure can be stopped (12) because the simulation time is over. In this case, the algorithm terminates (13). Otherwise, the algorithm waits for the next update by jumping to instruction (5).

4.3. Situation-based adaptation of the objective function

This subsection is about the definition of an implementation function H_1, which modifies the so far used objective function by adapting at the re-planning time t_i the so far used accounting scheme $(\lambda^a_{t_{i-1}}, \lambda^b_{t_{i-1}})$ to the new scheme $(\lambda^a_{t_i}, \lambda^b_{t_i})$ used for re-weighting the costs of the two fulfillment modes. In an LQ period, the re-weighting of the costs associated with the two modes is promising if the subcontraction costs are lowered relatively to the self-fulfillment costs. As soon as the least punctuality p^{target} is achieved again, the equal weighting for the costs of the two fulfillment modes becomes adequate again.

We define the weight $\lambda^b_{t_i}$ of the subcontraction costs to be 1 and do not vary this value anymore. In an HQ period, the weight $\lambda^a_{t_i} := H_1(t_i, p_{t_i})$ of the self-fulfillment mode is also 1. The tariff level α means, that, in average, one additionally subcontracted request produces costs that are α times larger than the additional costs in the self-fulfillment mode. If the weight $H_1(t_i, p_{t_i})$ of the self-fulfillment costs is larger than α then the subcontracting mode will preferentially be selected by the profit-oriented transport partner.

We propose the following procedure to determine the weight H_1. As soon as the punctuality p_{t_1} reaches $p^{\text{target}} + 0.1$ (and tends to leave the core $\mathscr{C}(t_i)$ of the system development corridor $\mathscr{D}(t_i)$) the weight $H_1(t_i, p_{t_i})$ is systematically increased if p_{t_i} is significantly less than p^{target}, and at the end it reaches $H_1(t_i, p_{t_i}) = 1 + \alpha$. Again, we exploit the piecewise-linear intensity function h introduced in Section 4.2 to determine the right intervention degree $h(p_{t_i})$.

If we define the real-valued function $H_1(t_i, p_{t_i})$ as described in Eq. (10) then $H_1(t_i, p_{t_i}) = 1$ in HQ periods. H_1 increases strictly if p_{t_i} decreases. If an LQ period is finally reached, then $H_1(t_i, p_{t_i})$ is close to or even equals $1 + \alpha$. We use the function H_1 as implementation function for adapting the objective function (9).

$$H_1(t_i, p_{t_i}) = \begin{cases} 1, & i = 0 \\ 1 + \alpha \cdot h(p_{t_i}), & i \geqslant 1 \end{cases} \tag{10}$$

5. Computational experiments

The SDAD-configuration prevents long-lasting phases of poor punctuality of the logistic system even if the tariff level for subcontraction is very high ($\alpha = 3$). As shown in Fig. 5, it keeps the maximal number of waiting requests as low as observed for the HARD-configuration. Furthermore, the time required to dismantle the enlarged queue of waiting requests is the same for SDAD- and HARD-configurations.

In order to analyze the impacts of the objective function adaptation (realized by the adaptation of the accounting scheme $(H_1(t_i, p_{t_i}), 1)$), we compare the results $\delta(\alpha, exp)$ achieved by varying exp for a given tariff level α. Table 4 shows a comparison of the maximal punctuality deviations $\delta(\alpha, exp)$. Similar to the results observed for the HARD- as well as for the PEN-configurations, $\delta(\alpha, SDAD)$ decreases with increasing tariff level α. Both the PEN- and the SDAD-configured systems perform similarly as long as $\alpha \leqslant 1.25$ but if α further increases then the SDAD-configuration leads to significantly moderated punctuality collapses ($\delta(3, SDAD) = -5.6$ and $\delta(3, PEN) = -38.8$). The results using the SDAD-configuration are closer to those using the HARD-configuration than to those achieved with the PEN-configuration.

The percentage of LQ periods increases as soon as the tariff level is lifted independently from the applied configuration (Table 5). If

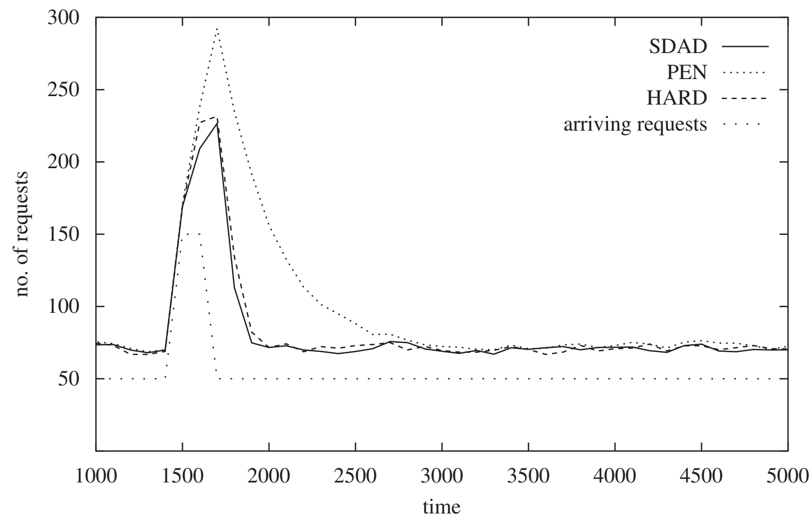

Fig. 5. Arriving and waiting requests ($\alpha = 3$).

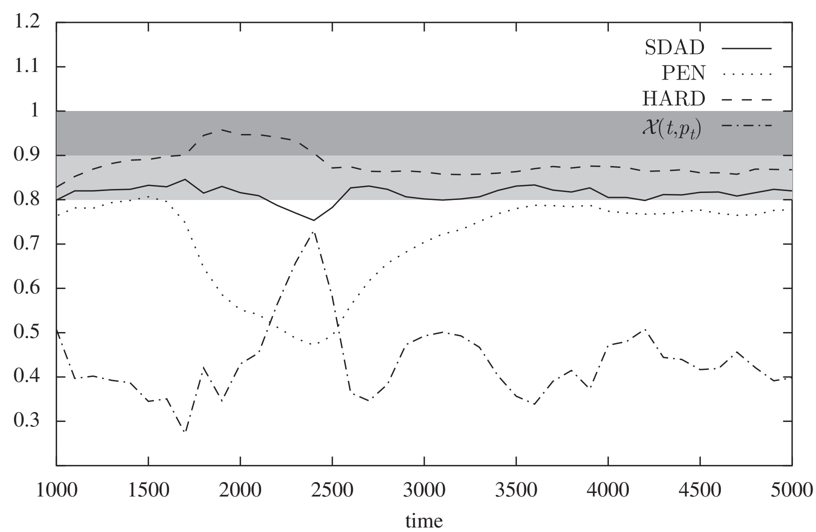

Fig. 6. Development of the punctuality $p_t(3, exp)$ and $\chi(t, p_t)$.

Table 4
Punctuality decrease $\delta(\alpha, exp)$.

exp	α					
	1	1.25	1.5	1.75	2	3
HARD (%)	3.6	0.0	5.3	4.4	4.2	3.5
PEN (%)	−1.0	−1.8	−7.8	−13.6	−22.0	−38.8
SDAD (%)	−0.9	−4.0	−5.4	−4.4	−6.9	−5.6

Table 5
Percentage of LQ-states $\pi(\alpha, exp)$.

exp	α					
	1	1.25	1.5	1.75	2	3
PEN (%)	–	63.9	80.6	91.7	94.4	97.2
SDAD (%)	–	–	2.7	5.6	5.6	16.7

a fixed charge is used to penalize tardy arrivals (PEN) then a slight LSP charge lifting leads to situations in which the least desired punctuality is not achieved ($\pi(\alpha, exp) > 0$). The application of a model adaptation defers the occurrence to higher tariff levels. The SDAD-configuration is able to prevent LQ periods up to a tariff level of 1.25 and the duration of the observed LQ periods for further increased tariff levels is quite short (2.7–16.7%).

Fig. 6 shows the punctuality development for a scenario with $\alpha = 3$. It can be seen, that, in the SDAD-configuration, the adaptation of the objective function leads to an attraction of the punctuality p_t to the system development corridor (grey shaded area). As soon as the punctuality runs out of the core of this corridor (dark grey shaded area) countermeasures are taken that immediately lead the punctuality back into the corridor.

Let $\chi(t_i, p_{t_i}) := (H_1(t_i, p_{t_1}) - 1)/\alpha$ represent the applied percentage of the maximal possible weight. Fig. 6 shows that the highest values for the percentage are applied as soon as the punctuality p_t runs out of the system development corridor. If the punctuality has recovered and re-entered the system development corridor then the applied percentage re-decreases.

The application of the static model definition rules in the HARD- as well as in the PEN-configuration leads to a significant sensitivity of the number of subcontracted requests in response to a tariff level lifting (Tables 1 and 3). The comparison of the results for the different configurations, presented in Table 6, shows that the adaptive

Table 6
Number of subcontracted requests $\sigma(\alpha, exp)$.

exp	α					
	1	1.25	1.5	1.75	2	3
HARD (%)	22.2	14.9	10.0	8.0	7.2	8.0
PEN (%)	21.4	15.5	10.0	5.8	5.1	4.1
SDAD (%)	23.8	18.5	19.2	17.4	15.0	14.5

Table 7
Relative increase $C'(\alpha, exp)$ of the overall costs.

exp	α					
	1	1.25	1.5	1.75	2	3
PEN (%)	−8.7	−5.5	0.1	4.9	5.0	−1.0
SDAD (%)	−3.1	6.1	9.6	15.9	16.0	13.5

rule exploited in the SDAD-configuration is able to shift a significant larger portion of requests into the subcontraction fulfillment mode than the static rules are able to do.

To analyze the cost impacts of a configuration change, we calculate the relative growth $C'(\alpha, exp) := \gamma(\alpha, exp)(5000)/\gamma(\alpha, HARD))(5000) - 1$ of the overall costs with respect to the reference values taken from the HARD-configuration simulation results. The $C'(\alpha, exp)$-values compiled in Table 7 show that the SDAD-configuration leads to additional costs compared to the HARD-configuration if the tariff level is larger than 1. Furthermore, the SDAD-configuration produces higher costs than the PEN-configuration does. Here, the over-weighting of the punctuality in the adapted objective function leads to additional costs because the selection of the cost minimal request fulfillment mode is compromised by the enforced preference for those modes that lead to the least number of late requests.

6. Conclusions and future research

We have investigated the automatic decision support for a dynamic decision problem. From the observed results we conclude that it is necessary to adapt the used formal decision model if the decision making situation has changed and if the used decision logic is not suitable anymore to fulfill the superior supply chain wide planning goals. This is observed in the $\alpha > 1$-cases where the punctuality in workload peak situations cannot be maintained using the static PEN-configuration. We have shown that the adaptation of the search direction by adjusting the deployed objective function is able to achieve a better integration of the two competing goals "cost minimization" and "service quality maximization". If settings like outlined in the HARD-configuration are not possible due to organizational aspects then the adaptation of the objective function makes the used decision support software system more resistant against impacts from varying problem data than a static penalization is able to do. In detail, the duration of LQ periods of the considered logistic system is significantly reduced if the decision model is adapted to the current system performance.

The adaptation of the objective function by emphasizing the degree of service quality turns the search away from cost minimal solutions of the decision problem instance. Future research efforts will be dedicated to refinements of the generic idea to adapt the search direction. Special efforts are required in order to reduce the additional costs caused by the intensified consideration of service quality goals. Additionally, it has to be analyzed whether other negative impacts like repeatedly deferred requests require additional methodological treatment. Although such impacts are not observed in the experiments investigated in this contribution, mechanisms to prevent the unlimited deferment have to be developed.

References

[1] Fleischmann B, Meyr H, Wagner M. Advanced planning. In: Stadtler H, Kilger C, editors. Supply chain management and advanced planning. Springer; 2002. p. 71–104.
[2] Dror M, Laporte G, Trudeau P. Vehicle routing with stochastic demands: properties and solution frameworks. Transportation Science 1989;23:166–76.
[3] Brehmer B. Dynamic decision making: human control of complex systems. Acta Psychologica 1992;81:211–41.
[4] Corsten H. Produktionswirtschaft. 11th ed., Munich: Oldenbourg; 2007.
[5] Vahrenkamp R. Produktionsmanagement. 5th ed., Munich: Oldenbourg; 2004.
[6] Ausiello G, Feuerstein E, Leonardi S, Stougie L, Talamo M. Algorithms for the on-line travelling salesman. Algorithmica 2001;29(4):540–81.
[7] Hiller B, Krumke S, Rambau J. Reoptimization gaps versus model errors in online-dispatching of service units for adac. Discrete Applied Mathematics 2006;154(13):1897–907.
[8] Gutenschwager K, Niklaus C, Voß S. Dispatching of an electric monorail system: applying metaheuristics to an online pickup and delivery problem. Transportation Science 2004;38(4):434–46.
[9] Krumke S. Online-optimisation–-competitive analysis and beyond. Technical University of Berlin; 2001.
[10] In: Fiat A, Woeginger GJ, editors. Online algorithms. Berlin: Springer; 1998.
[11] Wasserburger D. Dynamische Tourenoptimierung durch verkehrstelematische Daten. In: Schrenk M, editor. Proceedings of CORP 2006. 2006. p. 535–42.
[12] Arnold M, Fink SJ, Grove D, Hind M, Sweeney PF. A survey of adaptive optimization in virtual machines. Proceedings of the IEEE 2005;93(2):449–66.
[13] Šegvić S, Remazeilles A, Chaumette F. Enhancing the point feature tracker by adaptive modelling of the feature support. In: Leonardis A, Bischof H, Prinz A, editors. ECCV 2006, Part II. 2006. p. 112–24.
[14] Gutenschwager K, Böse F, Voß S. Effiziente Prozesse im Kombinierten Verkehr–-ein neuer Lösungsansatz zur Disposition von Portalkränen. Logistik Management 2003;5(1):62–73.
[15] Gutenschwager K. Online–-Dispositionsprobleme in der Lagerlogistik. Wurzburg: Physica-Verlag; 2002.
[16] Choi T-M, Li D, Yan H. Optimal returns policy for supply chain with e-marketplace. International Journal of Production Economics 2004;88:205–27.
[17] Sauer J, Appelrath H-J. Scheduling the supply chain by teams of agents. In: Proceedings of the 36th Hawaii international conference on system sciences; 2002.
[18] Thonemann U. Operations management. Pearson Studium; 2005.
[19] Solomon M. Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research 1987;35(2):254–65.
[20] Irani S, Ku X, Regan A. On-line algorithms for the dynamic traveling repairman problem. Journal of Scheduling 2004;7:243–58.
[21] Krumke SO, Rambau J, Torres LM. Real-time dispatching of guided and unguided automobile service units with soft time windows. In: Möhring R, Raman R, editors. Proceedings of ESA 2002, Lecture Notes in Computer Science, vol. 2461. Berlin: Springer; 2002. p. 637–48.
[22] Krajewska MA. Potentials for efficiency increase in modern freight forwarding. Gabler; 2008.
[23] Schönberger J. Operational freight carrier planning. Berlin, Heidelberg: GOR publications, Springer; 2005.
[24] Schönberger J, Kopfer H. On decision model adaptation in online optimization of a transport system. In: Günther H-O, Mattfeld DC, Suhl L, editors. Management logistischer Netzwerke. Berlin, Heidelberg: Springer; 2007. p. 361–81.
[25] Schönberger J, Kopfer H. On the value of objective function adaptation in online optimisation. In: Nickel S, Kalcsics J, editors. Operations research proceedings 2007. Berlin: Springer; 2008.
[26] Bierwirth C. Adaptive search and the management of logistics systems. Dordrecht: Kluwer Academic Publishers; 2000.