[Schö07] Schönberger, J.; Kopfer, H.: On Decision Model Adaptation in Online Optimization of a Transport System. In: Günther, H.-O.; Mattfeld, D.C.; Suhl, L. (eds.): Management logistischer Netzwerke. Entscheidungsunterstützung, Informationssysteme und OR-Tools, Springer, Berlin Heidelberg, 2007, pp. 361-381

# On Decision Model Adaptation in Online Optimization of a Transport System

Jörn Schönberger and Herbert Kopfer

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

## Abstract

This article is about the enhancement of myopic online decision approaches for considering longer term planning goals in the management of logistic processes in a dynamically varying environment. By means of a demand peak we simulate a severe disruption of the environment of a transport system and show that a pure myopic scheduling strategy is not able to ensure an acceptable service level in such a situation. As a remedy, we propose to adapt automatically the short term decision behaviour of the used decision making algorithm. We anticipate the instantiation of a reasonable number of decision variables in a model pre-processing step in order to break the rule of selecting the least cost but also low quality decision alternatives. Within several numerical experiments we prove the applicability and suitability of our approach.

## 1 Introduction

The ability to respond immediately to a customer-reported technical failure becomes a more and more important competitive factor for producers and retailers of consumer as well as industrial technical devices. Wide spread service and maintenance networks are maintained in order to provide a reliable and efficient after-sales technical support. These networks are typically managed by a central dispatching unit that deploys service teams in the field and guide them to the sites of customer-reported failures.

This article is about computer-executed planning for such a dispatching unit and addresses especially the planning support in situations in which the workload is so high that the service quality decreases significantly and leads to inflexible transport processes. To overcome such a situation semantic as well as syntactic adaptations of the underling formal decision model are necessary.

We propose an extension of the commonly known and used myopic online planning approach by an adaptively controlled model pre-processing procedure that reacts to quantitative as well as qualitative problem changes by carrying out anticipated instantiations of selected decision variables. The pre-processed model is then solved by an automatic re-planning method. This hybrid approach reduces the need for a contextually re-parameterisation of the metaheuristic decision algorithm. Furthermore, the intensity of the application of the pre-processor adapts itself to the currently observed degree of the fulfilment of the intended system reliability.

The main research questions for which we want to find initial answers in the remainder of this article are:

1. What are the main pitfalls of pure myopic decision making for managing logistic process in a dynamic environment?
2. How can myopic and longer term superior planning objectives be combined in a way that allows an automatic consideration in short term deployment decisions?
3. What are the impacts of breaking the rule of myopic cost minimization?

In Section 2 we introduce the investigated decision problem. Section 3 is about numerical experiments with a pure myopic decision making strategy. An adaptively parameterized model pre-processor is introduced in Section 4 and the numerical experiments carried out to assess this approach are presented and discussed in Section 5. We terminate with some summarizing remarks in Section 6.

## 2 Dynamic Decision Problem

This section is about the investigated dynamic decision problem. The problem is non-stochastic, e.g. requests are released randomly but we do not known the distribution of their arrival times. In Subsection 2.1 we survey the scientific literature related to the problem considered here. Subsection 2.2 outlines the problem informally. The life cycle model of a request is presented in Subsection 2.3 and the myopic decision problem to be solved whenever at least one additional request arrives is stated in 2.4. Artificial test cases required for a numerical simulation of selected problem instances are introduced in Subsection 2.5.

## 2.1 Literatur

Dynamic vehicle routing and scheduling problems are surveyed in [6]. [19] and [20] discuss the differences between vehicle routing and scheduling problems with deterministic and with probabilistic or incomplete planning data.

Jensen [14] understands *robust planning* as the generation of plans that maintain their high or even optimal quality even if subsequent modifications are required. *Flexible planning* is defined by Jensen as the generation of plans whose quality does not significantly decrease after algorithmic rescheduling has been applied and alterations of the so far used plans have been made.

A robust transport scheduling approach is proposed by Jaillet and Odoni in [13]. They construct optimal *a-priori-routes*. Such a route has a minimal expected length among all possible routes through the potential customer sites. However, this approach assumes that probability distributions about the future events are known.

Flexible planning approaches do not require any knowledge about future events. An existing plan is updated consecutively and reactively. Sequenced planning problem instances $P_i$ are solved sequentially. Such a sequence of decision problems $P_1$, $P_2$,… is called an *online planning problem* according to [3] so that the solving approach is referred to as online planning. A survey of online vehicle routing and scheduling problems is provided by [15]. Theoretical results for online repairmen dispatching strategies are found in [1] and [12].

Dispatching systems for transport planning tasks are proposed by Slater [24] as well as Gayialis and Tatsiopoulos [5]. Ghiani et al. [8], Gendreau et al. [7], Fleischmann et al. [4] and Gutenschwager et al. [10] investigate dispatching systems in which decisions have to be derived in real time without any delay.

An application of operations research methods to a relocation problem in medical rescue service is reported by Brotcorne in [2].

## 2.2 Verbal Problem Outline

Whenever a customer mentions an expected or real technical problem related to a product covered by an on-site service contract, she/he is advised to contact a call centre in order to report the problem. The contacted call centre agent records the information and compiles them within a customer request r, containing all necessary technical and spatial information belonging to the customer and the failure-causing device.

As an initial response to the failure report the customer agrees an on-site visiting time window with the call centre agent. The calling customer is promised that a mobile service team will arrive at the corresponding location and starts with the maintenance work within the determined time window. Since the contacted call-centre-agent typically does not know the complete list of reported errors an exact arrival time cannot be promised during the telephone call but the agreement about the time window allow the customer to organise that somebody will welcome the sent service team and it reduces the number of no-show-visits where customers are not at the site to meet the sent service team when it arrives.

In order to be able to offer a reliable service level and an acceptable response time, several call centre agents receive the calls in parallel. They put the received calls in a central queue of waiting requests. The task of the dispatcher of the group of field service teams is to decide about the way in which given customer demands are fulfiled by the field teams. Therefore, the dispatcher distributes customer site visits among the teams and determines the arrival time at the customer site. In order to keep the necessary operations costs as low as possible the dispatcher composes the requests into routes. Each service team is assigned to exactly one route and each route is assigned to exactly one team. The transportation plan contains all these routes and the routes are propagated to the service teams. A field team fulfils the tasks indicated in its associated route.

In some situations, the broken down device requires an immediate intervention (e.g. alarm systems, telephone systems in a company or medical devices in intensive care units within hospitals). Therefore, the agreed service time window $[e_r,l_r]$ is close to the release time $t_r$ of r. In such a case, the routes of the field teams for the given day have to be revised immediately and the additional visits have to be inserted into the routes [11]. We investigate such a situation in this contribution.

To satisfy the customer demands and to provide a reliable and sustainable service, the service providing company has to ensure

- that a sufficiently high percentage of customer sides are averagely reached in time (global reliability) and
- that each customer whose time window cannot be respected receives a compensation (local reliability).

Both aspects of reliability have to be taken into account while generating and updating a new schedule. A high global reliability supports striving for a sufficiently high market share and competitive advantages. In case that the number of additional customer requests increases rapidly the overall schedule is revised and all service teams are updated. Such a situation follows an unexpected event for example if an alignment peak has damaged TV sets or telephone devices in a complete quarter. Under these exceptional circumstances the available service teams cannot visit every customer site within the agreed time window. To stay in line with the company's punctuality policy and to offer a reliable service, the dispatcher can book external service teams from other companies. The decidable costs for the integration of external teams are quite high but often lower than the compensation payments to be transferred to out-of-time customers and the external service providers guarantee an in-time visit. However, the decision for externalization of a request cannot be revised in subsequent decision situations because the order given to an external service provider is obligatory.

The goal of the planning support to be developed is to establish a planning system that allows the generation and repeated update as well as adaptation of flexible transportation plans for the field teams including decisions about externalization of selected requests. The flexibility is important because the customer requests are received successively and their arrival times cannot be predicted or forecasted so that only a reactive transport plan revision is realizable. Furthermore, in order to maintain the flexibility of the transport plans even in situations with an extreme workload, it is allowed to violate the agreed time windows but the corresponding customers are paid compensation.

### 2.3 Online Request State Update

In order to consider the successively arriving additional requests, we propose to update the existing transportation plan reactively after the additional requests become known.

Let $t_i$ denote the i-th time when additional requests become available and let $R^+(t_i)$ represent the set of additional requests, released at $t_i$. After the last transportation plan update at time $t_{i-1}$, several requests have been completed. These requests are stored in the set $R^C(t_{i-1},t_i)$. Then the request stock $R(t_i)$ at time $t_i$ is determined by $R(t_i) := R(t_{i-1}) + R^+(t_i) - R^C(t_{i-1},t_i)$.

The life of a single request r consists of a sequence of states to which r belongs. Initially, when r enters the transportation system it is known but not yet scheduled (F). If r is assigned to an own vehicle for execution it is labelled by (I) or by (E) if r is assigned to an external service partner. A request whose completion work at the corresponding customer site has been started but not yet finished is labelled as (S). The final stage (C) of r indicates that r is completed.

Every time a transportation plan update becomes necessary, the current states of known requests from $R(t_i)$ are updated. The state (F) is assigned to all new requests from $R^+(t_i)$. For all requests contained in $R^C(t_{i-1},t_i)$, their state is updated from (I) or (E) to (C) and requests whose on-site execution have been started but not yet completed receive the new state (S) that replaces their former state (I) or (E). Now, the scheduling algorithm is started that carries out the necessary transportation plan updates. From the updated transportation plan the information about the intended type of request execution of all requests labelled as (F) or (I) is taken. The state of an (I)-labelled requests is updated to (E) if it has been decided to out source this request. Otherwise, the state of this request remains unchanged. Finally, all (F)-labels of externalized requests are replaced by (E)-labels for subcontracted requests and (I)-labels replace the (F) labels for the remaining requests from $R^+(t_i)$.

## 2.4 Statement of the Scheduling Problem

The decision whether a request should be assigned to an own team or given to an external partner cannot be solved uniquely for each request. A complex decision problem must be solved every time the currently valid transportation plan has to be updated, considering simultaneously all assignable requests, which are labelled by (I) or (F). It has to be decided for all these requests whether they are definitively subcontracted and given to a service partner for execution or if they should be assigned for the first time to one of the available own vehicles represented by the elements of set V(t). In order to find the minimal cost assignment, we propose the following optimization model.

Let $\Omega(t)$ denote the set of all possible request sequences $p=(p_1,\ldots,p_{n(p)})$ representing the order in which the contained customer requests, selected from R(t), are visited. Request r is contained in p if and only if the parameter $\mu(r,p)$ is set to 1. The vehicle v that has been selected for request r in the last transportation plan is denoted as $\Psi(r)$. If r is labelled as (I) then $\Psi(r)\in V(t)$, otherwise $\Psi(r)=\{\}$.

We assume that each $p\in\Omega(t)$ holds for the following two properties:

- The final entry $p_{n(p)}$ of p refers to the depot to which all vehicles return.
- If the first entry $p_1$ refers to a request labelled currently as (S) then the departing time from $p_1$ cannot precede the finishing time of this request.

The following two binary decision variable sets are used to code the necessary decisions. The variable $u_p$ is set to 1 if and only if sequence $p\in\Omega(t)$ is selected for vehicle $v\in V(t)$. Furthermore, $y_r$ is set to 1 if and only if request $r\in R(t)$ is subcontracted.

We are looking then for instantiations of the above decision variables that minimizes the costs $C(\{x_{pv}\},\{y_r\})$ but considering that

1. Each vehicle is assigned to exactly one (maybe an empty) path from $\Omega(t)$.
2. Each request is contained in at most one of the selected paths.
3. A request r labelled by (S) cannot be assigned to another vehicle as $\Psi(r)$.
4. If request r is labelled by (E) then $y_r=1$.
5. If vehicle v is assigned to p then $p_1$ must correspond to the current location of vehicle v.

We desist from giving the formal mathematical statement of the above five constraints since we do not need them in the remaining presentation.

The objective function $C(\{x_{pv}\},\{y_r\})$ calculates the costs associated to the instantiations of the two decision variable sets. It is the sum of the travel costs for the own deployed vehicles plus the sum for subcontraction fees and penalties to be paid for late arrivals at customer sites. Therefore, it denotes the costs for the associated transportation plan.

## 2.5 Artificial Test Cases

In order to evaluate different dispatching approaches and to control the severeness of the observed scenario, we have derived a set of artificial test instances. Each instance is defined by a special instantiation of a set of parameters. Different scenarios can be modelled by adjusting these parameters.

Two different kinds of dynamic routing scenarios are referred in the scientific literature. In the first scenario type, the number of demands that are released during a specific time interval remains unchanged. It is possible to adapt the available resources in such a situation so that all additional demands can be served in time. For this reason, such a scenario is called a *balanced scenario*. In case that the number of additionally released demands during a specific time interval varies, the scenario is denoted as a *peak scenario*. Here, it is hardly possible to adapt the available resources in advance.

Pankratz [18], Lackner [16] as well as Mitrović-Minić et al. [17] propose artificial benchmark instances for evaluating different dispatching strategies. In all these instances the number of additional requests for a given time interval remains equal as described for the balanced scenarios.

Gutenschwager et al. [10] and Sandvoss [22] use real world data sets for their experiments in which the intensity of incoming demands varies over time. Such instances represent examples of a peak scenario. Neither a parameterization nor a classification of these instances is possible.

To simulate peak scenarios we first generate a balanced stream of incoming customer demands over the complete observation time period. A second stream is generated for a part of the observation period. Both streams are than overlaid so that during the period in which the second stream is alive, the balanced stream is interrupted and a higher number of requests must be scheduled.

The balanced stream of incoming demands for the observation period $[0,T_{max}]$ is generated by successively drawing requests from the Solomon instance P [25]. At time $t^{rel}=0$, $n_0$ demands are drawn randomly from P. Then, the release time is updated by $t^{rel}:= t^{rel} +\Delta_t$. For this new release time, n demands are drawn from P at random. For each selected demand r, its release time is set to $t^{rel}$. The original service time window $[e_r,l_r]$ of r is replaced by $[t^{rel}+e_r, t^{rel}+l_r]$. Additional demands are generated as long as $t^{rel}\leq T_{max}$.

The second stream of demands is generated to simulate a peak of demands. For the first generated release time 0, $\Delta_t$ ,$2\Delta_t$ ,…, $n_1\Delta_t$ no demands are released. For the next $n_2$ release times $(n_1+1)\Delta_t$,…,$(n_1+n_2)$ $\Delta_t$ $\Delta_m$ demands are specified as described above. For the remaining release times, no additional demands are given.

All vehicles specified in P can be used.

Consequently, each scenario is described by the triple (P, $d^{peak}$, $\Delta_m$). In this investigation, we use the four Solomon cases R103, R104, R107 and R108 to generate request sets. The peak duration has been set to $d^{peak}$=200 time units and the peak high is fixed at $\Delta_m$=100 additional request.

## 3 Online Optimization Approach

For solving the instances of the online decision problem introduced in 2.4 we use a Memetic Algorithm realizing a hybrid search strategy consisting of a genetic search and a local 2-opt improvement procedure. The genetic search is realized by a $\mu+\lambda$-population model [21] evolved by the application of the PPSX-crossover-operator [23] and a mutation operator that replaces arbitrarily selected operations between routes, moves operations within selected routed as well as inversing the visiting order of subsequences of selected routes.

The initial population is generated using the Push-Forward-Insertion-Heuristic proposed by Solomon [25]. One half of the initial set of solution proposals is generated by using 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 problem instance has been stated the Memetic Search Algorithm is re-started. Initial 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 the recent population converges too rapidly even if the crossover and mutation probability are determined adaptively. For this reason, a complete new initial population is formed using the seeding approach described above.

This re-start approach has been evaluated within a simulation experiment. Within this experiment, we set the costs as follows: The travel costs for an own vehicle (self-fulfilment-mode) are set to one money unit (MU) for each travelled length unit and a too late arrival at a customer site is penalized by 25 MU. For each subcontracted request r the fees to be paid are calculated in two steps. First, the tariff distance d(r) is calculated as the Euclidian distance between the depot location (as defined in the considered Solomon instance) and the customer site location. The tariff to be paid is then determined as $1{,}1 \cdot d(r)$, which means that the costs for each subcontracted distance unit are 10 percent higher than the costs for travelling the distance with an own vehicle as done in the self fulfilment mode. The externally given target punctuality $p^{target}$ is set to 80%.

The scenarios introduced in 2.5 have been simulated in three independent runs each and the achieved punctuality has been recorded. After the runs, the averagely observed punctuality has been calculated for all replanning times. The resulting curve is shown in Fig. 1. The horizontal axis shows the on-going time and the vertical axis represents the averagely observed punctuality, in the remainder referred to as $p_t$.

Immediately after the demand peak is over and the requests released during the peak are executed (starting with time point 1800) the averagely observed punctuality collapses and reduces from over 80% down to less than 45% at time 2600. After this time the punctuality re-increases slowly but hardly reaches the intended value of 80% again within the observation time interval.

The presented results show that the punctuality rate cannot be kept on an acceptable level after the demand peak. Even the penalization of too-late-visits cannot prevent the significant and long lasting reduction in the service quality. Although subcontraction can help to reduce the number of time window violations, this fact is not recognized by the myopic scheduling algorithm.

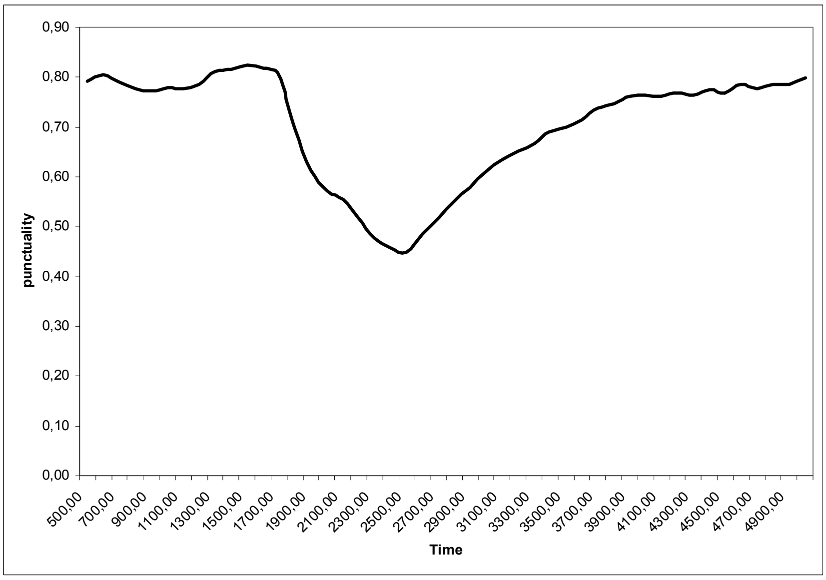


Fig. 1: Average punctuality values $p_t$

## 4 An Adaptive Online Optimization Approach

This section is about the definition and integration of a model pre-processing procedure that supports keeping the punctuality level on an acceptable level even during and after phases in which the incoming demand is significantly increased. Subsection 4.1 is about the definition of the procedure, Subsection 4.2 describes the control of the pre-processing and Subsection 4.3 describes its integration into the existing online optimization framework.

### 4.1 Model Pre-Processing

From the numerical results reported at the end of Section 3 we conclude that the pure myopic re-solving of the same type of optimization model instances is unable to cope with the challenges caused by the environmental shift. The subcontraction is not used before all available own vehicles are deployed because externalizing a request is more costly than serving it with an own vehicle and often the penalties to be paid are lower than the additional costs for request subcontraction. Consequently, subcontraction is not used for reducing the length of the queue of waiting requests if the punctuality decreases. This is caused by the blindness of the pure online approach for detecting quantitative as well as qualitative changes during the transition from one problem instance to the next one. If at a particular time a new problem instance is stated then only the available data is collected for defining the new model assuming that the model or model type is still appropriate for the new situation. No data analysis is carried out in order to compare the new situation with the situation when the last model was stated that fitted appropriately to the existing environment.

In order to enable the online decision approach to cope better with the changing number of released requests we propose a simple data analysis tool that detects early a decrease in the punctuality rate and that enforces an obligatory subcontraction by deciding about the subcontraction before the current model instance is given to the scheduling algorithm for solving. The decisions made in the pre-processing step cannot be revised in the optimization run invoked afterwards, so that the formal decision model given to the scheduling routine is manipulated. In the considered case, the decision model is an optimization model so that the manipulation can only affect the objective function associated to the model, the set of constraints or the domains of allowed values for the decision variables.

Since neither the modification of the objective function nor the adaptation of the constraint set can be controlled adequately, the pre-processing procedure targets the adaptation of decision variable domains. The basic idea for the pre-processing is to determine a percentage of requests that is surely outsourced by shrinking the domain $S_r=\{0,1\}$ of selected binary decision variables $y_r$ representing the decision about subcontraction or internal fulfilment by own vehicles down to $S_r=\{1\}$. Since all subcontracted requests are surely served within the agreed time interval, they can be served as early as possible and do not lead to an increase of requests served after the agreed time windows are closed.

## 4.2 Pre-Processing Control

In case that the punctuality rate is decreasing and runs into danger to fall below the target value $p^{target}$, the model pre-processor selects some requests which are definitively subcontracted. Answers have to be found for the following two questions:

1. How many requests should be selected in the pre-processing step for subcontraction?
2. Which requests should be selected for subcontraction?

In order to determine an adequate number of requests to be subcontracted by the pre-processor, we map the currently observed punctuality $p_t$ to a value $\gamma(p_t)$ between 0 and $\gamma^{MAX}$ representing the percentage of the next released additional requests for which the subcontraction is decided in the pre-processing step. As long as $p_t$ lies significantly above the target value $p^{target}$, no enforced subcontraction is necessary, this means as long as $p_t \geq p^{target}+\alpha$ the percentage $\gamma(p_t)$ should be 0. If the current punctuality has fallen significantly below $p^{target}$, all additionally released requests should be subcontracted, so that $\gamma(p_t)$ should be $\gamma^{MAX}$ as long as $p_t \leq p^{target}-\beta$. In case that $p_t$ increases between $p^{target}-\beta$ and $p^{target}+\alpha$ the percentage $\gamma(p_t)$ of surely subcontracted requests decreases from $\gamma^{MAX}$ down to 0. In case that $p_t$ decreases between the two mentioned values $\gamma(p_t)$ should increase from 0 up to $\gamma^{MAX}$. To achieve this, we use the following piecewise linear function G(y) defined by

$$G(y)=\begin{cases} 1 & , \quad y \leq p^{\text{target}}-\beta \\ \dfrac{-\gamma^{MAX}}{\alpha+\beta}y+\dfrac{\gamma^{MAX}}{\alpha+\beta}(p^{\text{target}}+\alpha) & , \quad p^{\text{target}}-\beta < y < p^{\text{target}}+\alpha \\ 0 & , \quad y \geq p^{\text{target}}+\alpha \end{cases}$$

as shown by an example in Fig. 2 where $\gamma^{MAX}=0.6$, $p^{target}=0.5$, $\beta=\alpha=0.1$ is assumed.

In order to determine the requests that are subcontracted within the pre-processing, we first calculate the number N of additionally released requests at the next time when requests arrive. Then, $\lceil G(y)N \rceil$ of these requests are selected randomly to be outsourced (the function $\lceil \cdot \rceil$ calculates the next larger integer value of a given fractional value).

### 4.3 Pre-Processing Integration in Online Optimization

In order to apply the model pre-processing within the online optimization approach introduced in Section 3, the percentage of requests selected for externalization in the pre-processing step is re-calculated every time a new call of the scheduling algorithm becomes necessary. Let $t-\Delta_t$ denote the time at which additional N requests become known. The so far followed schedule determined at time t requires a revision in order to consider the additional requests. Immediately after the requests are defined, the so far valid punctuality $p_t$ is mapped to the percentage $G(p_t)$ of requests that will be outsourced as the result of the model pre-processing. Then the number $\lceil G(p_t)N \rceil$ of requests to be sourced out is calculated and the corresponding requests are drawn randomly from the just released N additional ones. Af-

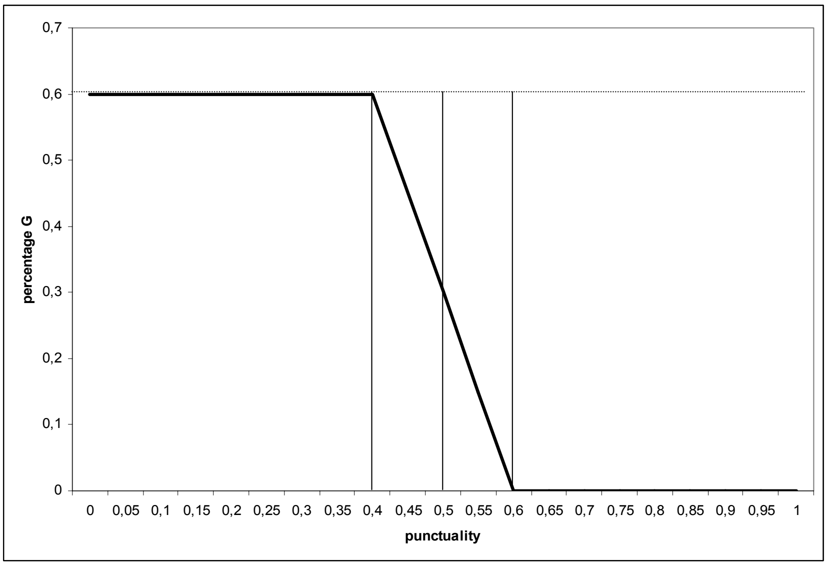


Fig. 2: Mapping the punctuality $p_t$ to the intervention intensity $G(p_t)$

ter that, the scheduling algorithm is re-started for updating the existing schedule. The decisions made in the pre-processing step cannot be revised within the execution of the scheduling procedure.

## 5 Assessment of the Pre-Processing Procedure

In order to evaluate the impacts of the proposed model pre-processing we have carried out several numerical experiments that are reported in this section. Subsection 5.1 is about the setup of the performed experiments whereas Subsection 5.2 presents the main achieved results.

### 5.1 Experimental Setup

We have used the same dynamic environment as described in Section 3. Again, the target punctuality $p^{target}$ is defined to be 0.80. We set $\alpha=\beta=0.05$. Therefore, an intervention within the pre-processing step starts if $p_t$ falls below 0.85 and reaches its maximal intensity $\gamma^{MAX}$ if $p_t$ falls further below 0.75. The maximal intensity $\gamma^{MAX}$ has been varied in the experiments. In addition to the case $\gamma^{MAX}=0.0$ (carried out in Section 3), we have tested the

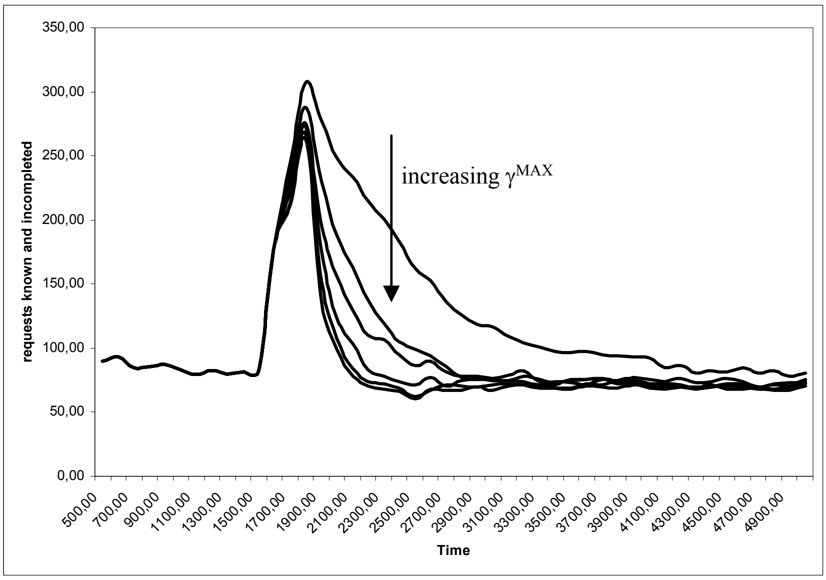


Fig. 3: Number $Q(\gamma^{MAX},t)$ of averagely queued requests

intensities 0.20, 0.40, 0.60, 0.80 and 1.00 so that we can analyse the impacts of an increasing intervention intensity.

As described in Section 3, the penalty value to be paid for a too-late arrival is set to 25 monetary units.

Each of the four scenarios R103, R104, R107 and R108 defined in Subsection 2.5 are simulated three times in order to achieve average results from the randomized memetic algorithm scheduler, so that 4×3=12 simulations are executed for each $\gamma^{MAX}$-value. Since we fixed $\gamma^{MAX}$ to six different values 72 simulation runs are carried out overall.

## 5.2 Presentation and Discussion of Numerical Results

We have recorded the number of queued requests within the executed numerical experiments for different maximal interventional values $\gamma^{MAX}$. For each $\gamma^{MAX}$-value, the development of the averagely observed queue length $Q(\gamma^{MAX},t)$ is shown in Fig. 3. Before the demand peak starts, an average queue-length of values between 80 and 100 waiting requests is reported. Immediately after the demand peak starts at time t=1500, the values of $Q(\gamma^{MAX},t)$ explode and increase up to 270-300 requests at time t=2000. Afterwards, the queue-length becomes smaller and smaller with

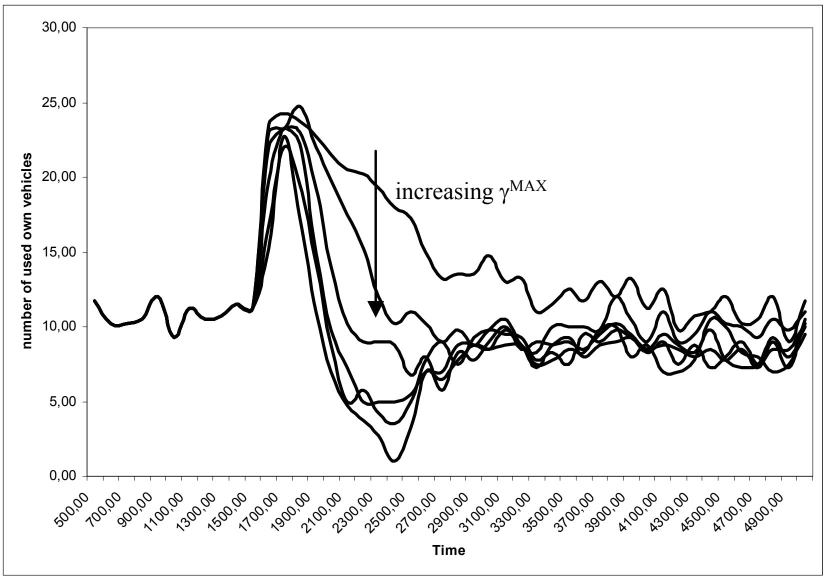


Fig. 4: Number $V(\gamma^{MAX},t)$ of averagely deployed own vehicles

ongoing time. Finally, they stabilise on the initially observed values between 80 and 100 waiting requests.

In order to analyse the impacts of the significant increase of waiting requests, we have observed the three parameters $V(\gamma^{MAX},t)$ representing the average number of routed own vehicles, $p(\gamma^{MAX},t)$ showing the achieved average punctuality as well as the average ratio $R(\gamma^{MAX},t)$ giving the average percentage of requests served by own vehicles. The development of $V(\gamma^{MAX},t)$ is shown in Fig. 4 As a first reaction after the detection of the increasing values of $Q(\gamma^{MAX},t)$, additional own vehicles, which has not been used so far, are deployed for serving the additional requests. During the demand peak from time 1500 until time 1700, $V(\gamma^{MAX},t)$ increases very quickly from 10 up to the maximal available number of vehicles, which is 25. After the demand peak (starting from time 2500), the number $V(\gamma^{MAX},t)$ of routed vehicles reduces. For the larger values of $\gamma^{MAX}$=0.6, 0.8, 1.0, it falls down below 5 and afterwards remains constantly between 5 and 10. For smaller maximal intervention intensities $\gamma^{MAX}$=0.0, 0.2, 0.4 the additionally used vehicles are successively taken out of service.

The averagely observed punctuality values $p(\gamma^{MAX},t)$ are compiled in Fig. 5. Independently of the value of $\gamma^{MAX}$ the punctuality can be saved on a nearly unchanged level above the target level 0.80 for the demand peak

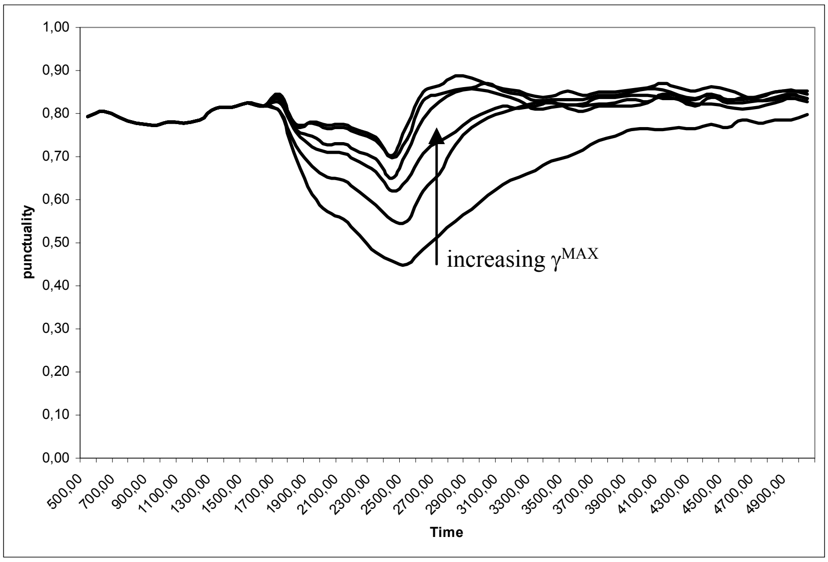


Fig. 5: Average punctuality values $p(\gamma^{MAX},t)$

time. After time t=1700, a significant and rapid decrease of $p(\gamma^{MAX},t)$ is shown. The minimally observed value for $p(\gamma^{MAX},t)$ as well as length of the interval in which $p(\gamma^{MAX},t)$ remains below the target punctuality $p^{target}$ strictly depend on $\gamma^{MAX}$. The most severe decrease is observed for $\gamma^{MAX}$=0.0 and is 0.45. The target value remains unreached throughout the remaining observation period. If $\gamma^{MAX}$ is increased the minimal value of $p(\gamma^{MAX},t)$ also increases. For $\gamma^{MAX}$=1.0 the punctuality falls only down to 0.70. Furthermore, the target value $p^{target}$=0.8 is re-achieved at time 2700 and does not fall below the target value again. The duration of the period, in which the punctuality lies below the target value decreases with increasing $\gamma^{MAX}$.

In Fig. 6 the percentage $R(\gamma^{MAX},t)$ of requests completed by own vehicles is presented. Again, the figure of the curve of $R(\gamma^{MAX},t)$ depends upon the value of $\gamma^{MAX}$. Clearly, for $\gamma^{MAX}$=0.0, R(0.0,t) remains unchanged just below 1.00 (now intervention is performed during the observation period). For $\gamma^{MAX}$-values larger 0, $R(\gamma^{MAX},t)$ decreases immediately after the demand peak has started. As larger $\gamma^{MAX}$ is as quicker is the decrease of $R(\gamma^{MAX},t)$ and the minimal observed value of $R(\gamma^{MAX},t)$ decreases with increasing $\gamma^{MAX}$-value. However, after the minimal value has reached, the percentage of internally served requests does not increase significantly any

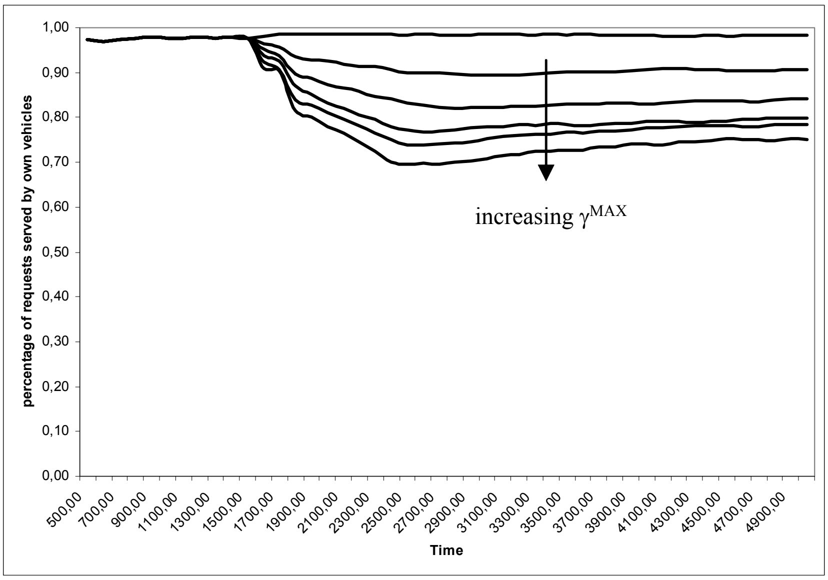


Fig. 6: Averaged percentages R($\gamma^{MAX}$,t) of requests completed by own vehicles

more and remains on a level quite distinct from the initially observed value just below 1.00.

In order to understand the aforementioned results, we study the intervention intensities G($\gamma^{MAX}$,t), that have been recorded during the performed numerical experiments. Fig. 7 shows that independent from the value of $\gamma^{MAX}$, each curve has two peaks immediately following each other. Each peak represents an interval during which the intervention intensities are significantly increased and the obligatory LSP-incorporation is enforced.

The first peak is observed immediately after the demand peak has started. It lasts from time 1500 until 1700. After this time first peak is over and the intensities fall back down to values between 0.10 and 0.15. Just after these values have been reached, the second peak of the $\gamma$-values is initiated at time 1800. It lasts until 2500. From this time on, the $\gamma$-values fall back down to values below 0.30 for the remaining observation interval until time 5000 is reached. Just after the demand peak leads to a decrease in the observed punctuality, the scheduler reacts and deploys more vehicles. However, not enough vehicles are available to serve all additional requests so that the punctuality further decreases which leads to an increase in the intervention intensity according to the intervention control introduced in Section 4. The first peak of the G($\gamma^{MAX}$,t) curves expresses the initial intervention. This overruling of the cost-based deployment heals the punctual-

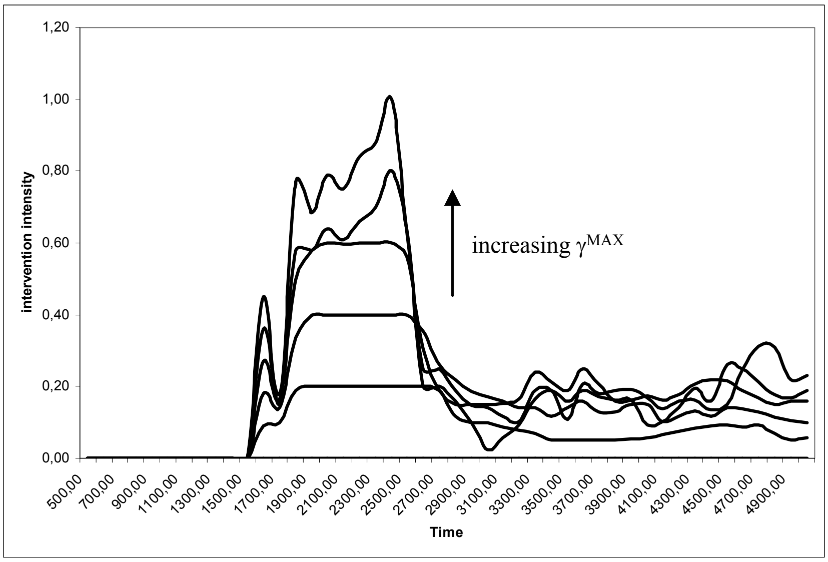


Fig. 7: Averaged intervention intensities G($\gamma^{MAX}$,t)

ity decrease as long as the queue length is not too large (at the beginning of the demand peak). As soon as the number of queued requests increases further, the punctuality P($\gamma^{MAX}$,t) cannot be supported anymore by enlarging the vehicles because their number is limited. For this reason the punctuality falls further which triggers now a significant increase in the intervention intensity expressed in the second peak of the G($\gamma^{MAX}$,t)-curves.

As it can be seen from the experimental results presented above that the autonomous control by adaptively determining the degree of obligatory request intervention supports striving for ensuring the service reliability even in situations with unexpected events, e.g. demand peaks. However, since the costs for the externalization of a requests are larger than the costs for serving this request with an own vehicle, the interventions for ensuring the punctuality causes additional costs as it can be depicted from Fig. 8. During the experiments, we have recorded successively the costs that were realized by letting the own vehicles run, paying external service partner for serving outsourced requests and paying penalties to customers for late arrivals. The averaged costs observed for each maximal intervention level $\gamma^{MAX}$ are denoted as C($\gamma^{MAX}$,t). The qualitative figure of the curves in Fig. 8 is independent from $\gamma^{MAX}$. With the introduction of the interventions the cumulated costs increases overproportionally. After the intervention peaks are over, the costs further increases linearly. At the end, the total costs

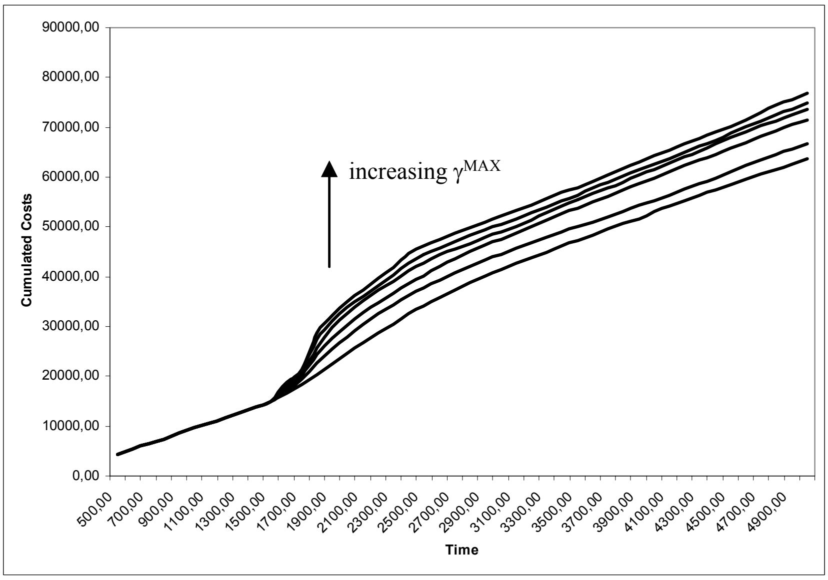


Fig. 8: Averaged cumulated costs $C(\gamma^{MAX},t)$

$C(\gamma^{MAX},5000)$ grow if $\gamma^{MAX}$ is increased, so that the most intensified interventions causes the highest additional costs.

## 6 Conclusions

We have studied the impacts of considering longer term reliability information in the short term plan update for the management of a transport system in dynamic environments. By increasing the weight of this type of information in the plan update decision making we achieve a higher reliability level compared with a pure myopic strategy.

With respect to the findings from the numerical experiments we are now prepared to formulate initial answers to the questions asked in the introduction of this article.

In pure myopic decision making the disregard of longer term goals in the particular decision problem instances leads to a complete failure with respect to the achievement of the longer term goals. These goals are not addressed at all.

The myopic goals(s), however, should be considered as the major goal to be followed in updating the existing transportation plan. A model preprocessing that fixes selected decision variables a way which helps to fulfil the longer term goals seems to be an adequate tool to integrate both short and longer term goals in the automatic decision making. The autonomous and self-adaptive adjustment of the decision variable domains seems to be adequate.

The realized cumulated costs are higher as soon as the consideration of the fulfilment of longer term goals is carried out in solving the instances of the online decision problem. However, the solution quality concerning the reliability issues is significantly improved compared to a pure myopic cost minimization. The additional amount of cumulated costs in the scenario with the model pre-processing can be understood as the costs for the additional consideration of the longer term goals.

We will continue with our research and investigate further the idea of autonomous model-adaptation in online planning approaches. Our next efforts will be spent to investigate ideas to reduce the additional costs related to the overruling of the minimal cost optimization strategies in solving the short term re-planning instances.

**Acknowledgement**. This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Center 637 "Autonomous Cooperating Logistic Processes".

## References

1. Bertsimas, D.J.; van Ryzin, G. (1989) The Dynamic Traveling Repairman Problem. MIT Sloan School Working Paper No. 3036-89-MS
2. Brotcorne, L., Laporte, G., Semet, F. (2003) Ambulance location and relocation models. European Journal of Operational Research 147, pp. 451-463.
3. Fiat, A.,Woeginger, G. (Eds.) (1998) Online Algorithms: The State of the Art. Springer.
4. Fleischmann, B., Gnutzmann, S., Sandvoß, E. (2004) Dynamic vehicle routing based on online traffic information. Transportation Science 38 (4), pp. 420-433.
5. Gayialis, S., Tatsiopoulos, I. (2004) Design of an it-driven decision support system for vehicle routing and scheduling. European Journal of Operational Research 152, pp. 382-298.
6. Gendreau, M., Potvin, J.-Y. (1998) Dynamic vehicle routing and dispatching. In: Crainic, T., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer, pp. 115-126.
7. Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, E. (1999) Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science 33 (4), pp. 381-390.

8. Ghiani, G.; Guerriero, F.; Laporte, G.; Musmanno, R. (2003) Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. European Journal of Operational Research 151, pp. 1-11.
9. Golden, B., Assad, A. (Eds.) (1988) Vehicle Routing: Methods and Studies. North-Holland.
10. Gutenschwager, K., Niklaus, C., Voß, S. (2004) Dispatching of an electric monorail system: Applying metaheuristics to an online pickup and delivery problem. Transportation Science 38 (4), pp. 434-446.
11. Ichoua, S.; Gendreau, M.; Potvin, J.-Y. (2003) Vehicle dispatching with time dependent travel times. European Journal of Operational Research 144, pp. 279-396.
12. Irani, S.; Lu, X.; Regan, A. (2004) On-line Algorithms for the dynamic traveling repair problem. Journal of Scheduling 7, pp. 243-258
13. Jaillet, P.; Odoni, A. (1988) The probabilistic vehicle routing problem. In: Golden and Assad (1988), pp. 293-318.
14. Jensen, M.T. (2001) Robust and flexible scheduling with evolutionary computation, PhD Thesis, University of Aarhus, Denmark.
15. Krumke, S. (2001) Online-Optimization – Competitive Analysis and Beyond. Habilitation Thesis, Technical University Berlin, Germany.
16. Lackner, A. (2004) Dynamische Tourenplanung mit ausgewählten Metaheuristiken. Cuvilier Verlag Göttingen.
17. Mitrović-Minić, S.; Krishnamurti, R.; Laporte, G. (2004) Double-horizon based heuristic fort he dynamic pickup and delivery problem with time windows. Transportation Research B 38, pp. 635-655.
18. Pankratz, G. (2002) Speditionalle Transportdisposition. DUV.
19. Psaraftis, H. (1988) Dynamic vehicle routing problems. In: Golden and Assad (1988), pp. 223-248.
20. Psaraftis, H. (1995) Dynamic vehicle routing: status and prospects. Annals of Operations Research 61, pp. 143-164.
21. Rudolph, G. (2000) Evolution strategies. In: Bäck, T.; Fogel, D.B.; Michalewicz, Z. (Eds.) Evolutionary Computation, IoP, pp. 81-88.
22. Sandvoss, E. (2004) Dynamische Tourenplanung auf Basis von Online-Verkehrsinformationen. Pro Business.
23. Schönberger, J. (2005) Operational Freight Carrier Planning. Springer.
24. Slater, A. (2002) Specification for a dynamic vehicle routing and scheduling system. International Journal of Transport Management 1, pp. 29-40.
25. Solomon, M. (1987) The vehicle routing and scheduling problems with time window constraints. Operations Research 35 (2), pp. 254-265.