2009

RULES FOR THE IDENTIFICATION OF PORTFOLIO-INCOMPATIBLE REQUESTS IN DYNAMIC VEHICLE ROUTING

Jörn Schönberger
University of Bremen

Herbert Kopfer
University of Bremen

RULES FOR THE IDENTIFICATION OF PORTFOLIO-INCOMPATIBLE REQUESTS IN DYNAMIC VEHICLE ROUTING

Jörn Schönberger, Herbert Kopfer[1]

Abstract
In this article, we propose and evaluate simple rules for selecting transport requests that do not fit into a request portfolio because their temporal or spatial requirements are incompatible with the requirements of other requests so that the compilation of profitable routes is compromised. We integrate these rules into an adaptive online vehicle operations planning system and analyze in numerical simulation experiments how their application has impacts on the flexibility, the stability and the profitability of the controlled transportation system and the integration of consecutively arriving requests.

1. Introduction

The identification of bundles of transport requests to be combined in profitable routes is a core decision task in operational transport process planning. Spatial, temporal and kind-of-good related information are exploited in order to build request clusters which are then completely assigned to transport resources. Often, selected requests do not fit to the others. They are located far away from all other requests or their time window requirements prevent a consolidation with other requests into the route of a vehicle. In such a situation, the outsourcing ("subcontracting") of such a request is the only opportunity to protect the overall system performance and profitability even if the request-associated costs of subcontracting are enlarged compared to the costs for fulfilling the request with an own vehicle.

Consecutively arriving customer requests cause revisions of once created processes so that the decision situation becomes even more challenging. Beside the necessity for updating the processes it is necessary to adjust (adapt) the used decision logic to the updated problem situation, if the decision problem input data (number of requests, vehicles etc.) have varied significantly. Here, the process control circuit consisting of the process and a decision model ("controlled subsystem") is coupled with a second control circuit ("controller"). The controller detects changes in the subsystem's environment and implements necessary adjustments into the controlled subsystem [7], [11]. This extension of the online decision making paradigm is called *adaptive online decision making*. In addition to solving a new instance of the maintained decision model, it is necessary to decide about the appropriate severeness of the decision model adjustment and afterwards it is necessary to

[1] University of Bremen, Wilhelm-Herbst-Straße 5, 28359 Bremen, Germany

select adequate adjustments. The first task ("severeness detection") can be modeled by control-signal functions that map performance indicators to a continuous value that represents the intensity of the severeness of the model adjustment. So far, the impacts of biasing the necessary adjustments of the controlled subsystem are not investigated.

The definition and evaluation of simple rules for the determination of suitable model adjustments are subject of this article. We want to check exemplarily whether the following research hypothesis within a given artificial transport planning system is true: *"The performance of the transport system (measured in flexibility, stability and costs) increases, if more knowledge is considered for the run-time adaptation of the decision logic"*. Clearly, we cannot prove this statement in its most generality. Therefore, we introduce a specific dynamic decision problem into Section 2 and configure an adaptive process control system in Section 3. Different rules for the run-time adaptation of the process control system which exploit different request information are proposed in Section 4. Numerical experiments are reported in Section 5.

2. Decision Scenario Description

Previous and Related Work. Recent surveys on dynamic transport process planning problems are given in [3], [7] and [13]. The generic idea of adjusting a formal decision model of a process planning agent is called *image modification* [1]. Image modification approaches for mathematical optimization models try to vary/adjust/replace a global objective function by single-usage instance-specific objective functions [4], [7] and/or try to sharpen and/or relax constraints [6]. A generic system layout for an integrated planning system with image modification has been proposed in [11] and a comparison of the two general adaptation strategies (objective function as well as constraint set adaptation) is reported in [8], [9] and [12].

Using Options in the Supply Chain Order Fulfillment Planning. The fulfillment of customer orders in a supply chain is organized as follows [7]. Customers express their demand in terms of external orders submitted to the supply chain coordinator. This coordinator receives the external orders as customer orders and takes over the responsibility for their reliable fulfillment [2], [5]. The coordinator splits each customer order into the necessary internal purchasing, production, distribution 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 responsible for the fulfillment of the specified internal requests according to their competencies in order to contribute to the fulfillment of the customer orders.

The supply chain coordinator agent receives charges paid by the customers for the fulfillment of the customer orders. From the sum of earned charges, budgets are defined that are used to cover the material flow process costs specified by the service centre agents. In order to stimulate a service centre to determine processes of highest efficiency, the difference between the budget and the process costs remains in the service centre as its gain (profit).

Contracts manifesting the relationships between the coordinator and the involved service providers are made for a longer term period and use estimated average workloads to determine the budgets, penalties and the quality of the request fulfillment like punctuality rates. However, in the daily business there are several situations in which the contracted service quality runs into danger to be compromised by an increase of absolute workload (additional requests) or relative workload (machine failures, etc.). The consecutively and unpredictably arriving requests require a revision of the so far used processes determined by the service providers. Therefore, the problem of determining the adequate processes is a dynamic decision problem.

Two fulfillment modes are available for the completion of a request. Own vehicles are deployed in the self-fulfillment mode (SF) while in the subcontracting mode (SC) external service providers are booked and paid for the fulfillment of the subcontracted requests. The main differences between the two modes are (i) their reliability and (ii) their associated costs. In the SC-mode a request is served in time in every case (assuming the availability of a suitable external logistic service provider) but in the SF-mode some requests might be late due to a large number of customer sites waiting for a visit. However, an SC-mode completion of a request is more expensive than a (delayed) SF-mode completion so that the subordinate service providing agent typically prefers the SF-mode. We investigate the simplified scenario outlined in Fig. 1. Requests emerge from transportation demand of the production stage towards the retail stage in the considered supply chain. The distribution service provider receives the requests and fulfills them, so that the required transportation of goods towards the retail stage of the supply chain is realized.

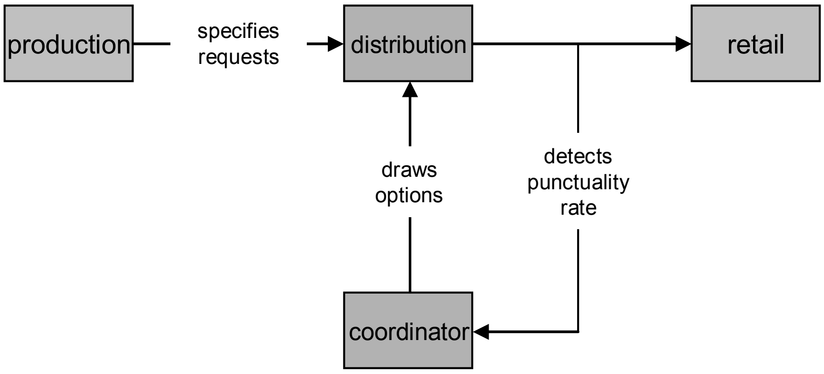

Figure 1: Investigated Scenario

In order to conserve the effectiveness and efficiency of the overall supply chain, the coordinator hedges himself by specifying options that enable him to intervene into the otherwise independent planning process of the subordinate service department. In particular, the coordinator overrules cost-based process decisions of the subordinate service department agent in order to ensure that critical requests are fulfilled with the highest priority and reliability independently of the costs. Such an intervention is necessary because service departments are not informed about the requirements of the customer demand. Only the details of the internal requests are provided. To compensate the resulting additional process costs of the service providing agent, the realization of such an option is coupled with an increase of the budget, so that the profit of the service provider is affected only partly or is even remained unchanged. The coordinator continuously observes the punctuality rate p_t of the fulfilled requests [7]. As soon as this indicator falls below the given threshold of p^{target}=80% the coordinator starts exercising the options to intervene. Here, an option associated with a request r grants the coordinator the right to pre-select the SC-mode for request r. The following two decisions must be made to implement the options.

Intensity of the intervention (how many options should be exercised?). To keep the subcontraction costs as small as possible, the number of exercised options is kept as low as possible. If p_t lies significantly above the intended target punctuality p^{target} at the re-planning time t then none of the currently unscheduled requests is enforced into the SC-mode. If the current p_t-value has fallen significantly below the target punctuality, then all recently released requests are immediately directed into the SC-mode without taking the fulfillment costs into account. In a transition phase if p_t falls down (or grows up), the number of requests N_t^{PRE} with pre-determined fulfillment mode is increased (reduced).

Selection of the surely subcontracted requests (for which requests should an option be drawn?). All requests arriving at time t are collected in the set $R^+(t)$. To complete its intervention, the coordinator selects N_t^{PRE} requests from $R^+(t)$ and stores these selected requests into the set $R(t, p_t)$. The SC-fulfillment mode is irrevocably fixed for all requests $r \in R(t, p_t)$.

Adaptive Online Deployment Model. We map the dynamic disposition task of the distribution centre agent into an online optimization model consisting of a sequence of optimization problem instances P_0, P_1, … which are solved consecutively at the dispatching times t_0, t_1, …. Each instance P_i is complete in the sense that it considers all problem data known at time t_i. A generated solution TP_i (set of processes) is executed until additional requests arrive at time t_{i+1}. A new optimization model instance $M(t_{i+1})$ is setup and solved then. The solution replaces the not yet executed process parts from TP_i by the recently generated processes collected in TP_{i+1} and adds process instructions for the additional requests. A complete and detailed discussion of the optimization model is presented in [6].

All requests for which the SC-mode has already been selected in $TP_{i-1,}$ are collected in $R^E(t_i)$. Beside the typical routing constraints the model includes the constraint (1). This constraint enforces the binary decision variable y_r into the value "1" (indicating that request r is subcontracted) and ensures that all previously externalized requests remain subcontracted. But it also ensures that all requests contained in $R(t_i, p_{t_i})$ are externalized so that the adaptive interventions are implemented into the updated processes.

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

The constraint (1) enables the adaptation of the decision model to the punctuality rate p_{t_i}. Thereby, the knowledge acquired during the online-model processing is automatically fed back into the formulation of the next decision task(s) model.

Since the transport service provider agent decides in general independently about the deployment of the available transport resources, the model does not comprise a restriction like "80% of the request stock must be in time". Furthermore, requests whose execution time is expected to be far in the future are only temporarily and tentatively scheduled. With the arrival of additional (currently unknown) requests, they are re-scheduled several times until their final completion time is fixed. The consideration of these requests in such a hard constraint is of limited worth.

3. Algorithmic Approach

Framework. The algorithmic framework integrating the coordinator's and the transport service department's decision making is shown in Fig. 2. Initially, the iteration counter i is set to 0 (a) and the first planning time is fetched (b). Next, an initial solution is generated (c) and broadcasted to the vehicles of the transport service department and to the subcontractor(s) (d). Now, the procedure is idle and waits until the current solution has been completely executed or additional requests are received (e). In the first case, the procedure stops (f) and is re-started as soon as additional requests become known. If the process execution is still in progress, then the iteration counter is increased by 1 (g) and the current system time is fetched (h). All requests just released at time t_i are put into the set $R^+(t_i)$ (i). Next, it is checked whether the consideration of the additional requests compromise the execution of the current solution (j). The procedure falls back into an idle state if the current solution is not corrupted by the additional requests. Otherwise, the current performance (punctuality rate) is calculated (k) and an error signal is derived (l). The intervention intensity is determined in dependence of the error signal value (m) and the requests which are prematurely directed into the SC fulfillment mode are selected (n). Afterwards, the new decision model is defined (o) and a high quality solution of this model is derived (p) which replaces the so far followed solution. The new solution is broadcasted to inform the field teams and the subcontractors (q). Again, the procedure falls back into the idle (waiting) state (r).

PROCEDURE process_management(Ψ, β);
(a) i:=0;
(b) t_i:= GET_CURRENT_TIME();
(c) CurrentSolution := GENERATE_INITIAL_SOLUTION();
(d) BROADCAST(CurrentSolution);
(e) **wait until** (CurrentSolution is completed) **or** (additional requests are released);
(f) **if** (CurrentSolution is completed) **then goto** (r);
(g) i:=i+1;
(h) t_i:= GET_CURRENT_TIME();
(i) $R^+(t_i)$:= GET_RELEASED_REQUEST(t_i);
(j) **if not** (SOLUTION_CORRUPTED(CurrentSolution)) **then goto** (e);
(k) p_{t_i} := GET_CURRENT_PUNCTUALITY(t_i);
(l) $e(t_i)$:= GET_CURRENT_ERRORSIGNAL(p_{t_i});
(m) $h_\beta(e(t_i))$:= GET_INTERVENTION_INTENSITY($e(t_i)$);
(n) $R(t_i,p_{t_i})$:= SPECIFY_INTERVENTION($h_\beta(e(t_i))$; $R^+(t_i)$; Ψ);
(o) $M(t_i)$:= DEFINE_MODEL(t_i, CurrentSolution, $R(t_i,p_{t_i})$);
(p) CurrentSolution := SOLVE_MODEL($M(t_i)$);
(q) BROADCAST(CurrentSolution);
(r) **Goto** (e);
(s) stop();

Figure 2: Pseudo Code of the Algorithm Framework

Adapting the Model of the Next Problem Instance (steps (k)-(o)). We only use the performance indicator p_{t_i} with the associated image set [0;1] whose current value is fetched by the function GET_CURRENT_PUNCTUALITY(t_i). The reference input $r(t_i)$ is defined by the closed interval $r(t_i):=[p^{target};1]$. This leads to the system development corridor $D(t_i):= [t_i;\infty)\times[p^{target};1]$ describing the desired future system performance and its core $C(t_i):= [t_i;\infty)\times[p^{target}+0.1;1]$. Since $p^{target}=0.8$ we get the system development corridor $[t_i;\infty)\times[0.8;1]$ and its core $[t_i;\infty)\times[0.9;1]$. As long as $p_{t_i}\geq 0.9$ the current system performance (t_i, p_{t_i}) belongs to the core $C(t_i)$. If p_{t_i} falls below 0.9 and if the distance of p_{t_i} from 0.9 increases then the system's performance gets more and more off the core $C(t_i)$ and finally leaves even the system development corridor $D(t_i)$. This leads to the following error signal (2) that is calculated by calling the function GET_CURRENT_ERRORSIGNAL(p_{t_i}).

$$e(t_i):= -\min(p_{t_i} - (p^{target}+0.1); 0) \qquad (2)$$

The error signal prematurely indicates that the performance runs into danger to leave the system development corridor as soon as the next external disturbance like a peak in the system workload occurs.

The controller transforms the previously calculated error signal $e(t_i)$ into a control value that manipulates the existing decision model afterwards. Therefore, it is a mapping h_β that assigns the error signal $e(t_i)$ to the control value $h_\beta(e(t_i))$. We define h_β as the piecewise linear function (3) which is calculated by calling GET_INTERVENTION_INTENSITY($e(t_i)$)

$$h_\beta(e(t_i))=0, \text{ if } e(t_i)\leq 0;\ h_\beta(e(t_i))=0, \text{ if } e(t_i)\geq 0.2;\ h_\beta(e(t_i))=5\beta e(t_i) \text{ in all other cases} \quad (3)$$

We interpret $h_\beta(e(t_i))$ as the percentage of the requests recently released at time t_i for which the SC-mode is chosen using an option exercised by the coordinator. The value β determines the maximal percentage of just arrived requests, for which an option is exercised. The number $N_{t_i}^{PRE}$ of affected requests is determined as specified in (4).

$$N_{t_i}^{PRE} := \left\lceil \left|R^+(t_i)\right| \cdot h_\beta(e(t_i)) \right\rceil \quad (4)$$

No request is enforced into the SC-mode (no option is exercised) if the error signal is 0. All additional requests released at t_i are enforced into the SC-mode if the error signal reaches its maximal value of 1. The percentage of enforced externalization increases smoothly with an increasing error signal.

Finally (corresponding to step (n) in the framework procedure in Fig. 2), the specification of the intervention is carried out by calling the function SPECIFY_INTERVENTION($h_\beta(e(t_i))$; $R^+(t_i)$; Ψ). The set $R(t_i, p_{t_i})$ of recently released requests, which are directed into the SC-fulfillment mode, is filled. We first arrange the n_i elements contained in $R^+(t_i)$ in a sequence SEQ($R^+(t_i)$,Ψ):=$(r_{i_1}, r_{i_2}, \ldots, r_{i_{n_i}})$ according to a request sequencing rule Ψ. Then, we consecutively insert the requests r_{i_1}, r_{i_2}, … into the set $R(t_i, p_{t_i})$ which contains exactly those requests for which the SC fulfillment mode is pre-determined. If the number of elements in $R(t_i, p_{t_i})$ has reached the number $N_{t_i}^{PRE}$ we stop with the insertion of requests into $R(t_i, p_{t_i})$. The call of the function DEFINE_MODEL(t_i, CurrentSolution, $R(t_i, p_{t_i})$) triggers the formulation of the next decision model instance $M(t_i)$. The fulfillment mode of the remaining requests can be freely determined in the process optimization.

Solving the Adjusted Deployment Model. For solving the instances of the online decision problem introduced in Section 2 we use a Memetic Algorithm realizing a hybrid search strategy consisting of a genetic search and a local 2-opt improvement procedure. Every time a new decision problem instance model has been stated the Memetic Search Algorithm is re-started by the call of the SOLVE_MODEL command (step (p) in the procedure in Fig. 2) [7].

4. Priority Rules for the Pre-Selection of Surely Subcontracted Requests

A sequencing rule Ψ determines the order SEQ($R^+(t_i)$,Ψ) of the elements contained in $R^+(t_i)$. A numerical value *sorteval(r)* is assigned to each request $r \in R^+(t_i)$. Furthermore, Ψ declares, whether these requests are sorted by increasing or decreasing evaluation values *sorteval(•)*.

A simple sequencing rule derives the *sorteval(r)* value for request r by analyzing only the specifications of this single request r. Spatial information associated with r like its location or distance to a fixed reference point or temporal information associated with r like the length of its associated time window or its release time are exploited in order to determine *sorteval(r).*

Distance-to-be-Bridged Sequencing (DBS). Requests in the middle of the operations area can more often be combined with other requests into profitable routes than requests which are far away from the centre (median) of the operations area. We first calculate the median m from all requests contained in $R^+(t_i)$. Let μ_r be the location of the site associated with the request r and let dist(m, μ_r) denote the Euclidian distance of the site of request r to the calculated median m. We define the following sorting criterion (5) and sort the requests from $R^+(t_i)$ so, that the *sorteval(•)*-values decrease. Requests situated on the periphery of the operations area are the first to be subcontracted in the expectation that they cannot be profitably combined with other requests into routes.

$$\text{sorteval}(r) := \text{dist}(m, \mu_r) \tag{5}$$

Vehicle Availability Sequencing (VAS). For each request $r \in R^+(t_i)$ the number vn_r of requests that can reach the site μ_r from their current positions before the time window of r closes is calculated. This number defines the sorting criterion (6). Then, the requests in $R^+(t_i)$ are sorted by increasing vn_r-values. Consequently, those requests which cannot be reached in time or only by few own vehicles are subcontracted preferentially. Penalty payments for late arrivals are tried to be prevented.

$$\text{sorteval}(r) := vn_r \tag{6}$$

Remaining Time Based Sequencing (RTS). At time t, RTS sorts the recently arrived requests by increasing remaining time in which the site μ_r of request r can be visited without violating the associated time window $TW(r):=[t^+_r,t^-_r]$. Therefore, we determine the sorting criterion as shown in (7). The requests in $R^+(t_i)$ are then sorted by increasing *sorteval(•)*-values.

$$\text{sorteval}(r) := t^-_r - t_i \tag{7}$$

Expenses and benefits of a single request can hardly be evaluated since the coupling effects of combining the fulfillment of several requests are very high. Isolated requests should be preferentially selected for being forwarded to a subcontractor because they corrupt the performance of the routes of the own vehicles.

Isolation Based Sequencing (IBS). In order to quantify the "degree of isolation" of the site μ_r we first calculate for each request r its distance $d_1(r)$ from the median m of the current request portfolio. After having calculated this distance for each request in $R^+(t_i)$, we calculate the normalized distance $d_1^*(r) := d_1(r) / \max\{d_1(r) | r\in R^+(t_i)\}$ for each request $r\in R^+(t_i)$. If $d_1^*(r)$ is close to 1 then μ_r is situated at the edge of the operations area which is often a first hint for isolation. To find out whether r can be combined with other requests to an efficient route, we calculate the distance mindist(r) to the nearest other request site in the complete request portfolio $R(t_i)$, that has not yet been subcontracted. It is mindist(r):= $\min\{d_2(r, r_j)+ d_3^{tw}(r, r_j) | r_j\in R(t_i), r_j \text{ not subcontracted}\}$, where $d_2(r, r_j)$ gives the travel distance between μ_r und μ_{r_j}. The term $d_3^{tw}(r, r_j)$ is used to depreciate the spatial distance in case that the time windows $TW(r):=[t_r^+,t_r^-]$ and $TW(r_j)=[t^+_{r_j}, t^-_{r_j}]$ of r and r_j interdict the combination of the two requests in one route. It is $d_3^{tw}(r,r_j):=0$, if $\min\{\| t^+_r - t^-_{r_j} \|, \| t^+_{r_j} - t^-_r \|\}\geq \text{dist}(r,r_j)$ (that is, there is enough time for a vehicle to travel from μ_r to the site of r_j or vice versa) and in all other cases it is $d_3^{tw}(r, r_j) := dist(r, r_j) - \min\{\| t_r^+ - t_{r_j}^- \|, \| t_{r_j}^+ - t_r^- \|\}$. Finally, we calculate the normalized minimal distance indicator mindist*(r)=mindist(r) / $\max\{mindist(r) | r \in R^+(t_i)\}$. The value (8) is then assigned as sorting value to the request r. If *sorteval(r)* is small (close to 0) then the site μ_r is either in the centre of the operations area and/or it is close to the sites of other requests. If a request r is situated at the edge of the operations area and not closely situated to the sites of other requests then r can be classified as isolated (*sorteval(r)* close to 1). We sort the requests in $R^+(t_i)$ by decreasing *sorteval(•)*-values and get the request selection order SEQ($R^+(t_i)$, IBS). At the beginning of this order those requests which seem to be most isolated are found and these requests are subcontracted preferentially.

$$\text{sorteval}(r) := d_1^*(r) \bullet \text{mindist}^*(r) \tag{8}$$

In order to find out whether DBS, VAS, RTS or IBS have a positive impact on the overall performance of the considered logistic system, we compare the results achieved by applying the four previously described priority rules in the simulation experiments with the reference rule RRS (Random Request Sequencing). If this rule is applied then a randomly selected value is drawn from the interval [0,1] (assuming a uniform distribution), assigned to sorteval(r) and the requests from $R^+(t_i)$ are then sorted by increasing *sorteval(•)*-values.

5. Computational Experiments

Experimental Setup. Four different imbalanced streams $i \in \{R103, R104, R107, R108\}$ of incoming transport requests [6] have been analyzed. Each stream is combined with one of the maximal intervention intensities $\beta \in \{0.2, 0.4, 0.6, 0.8\}$ and each of the resulting 16 scenarios (i, β) is executed under utilization of the request selection rules $\Psi \in \{DBS, VAS, RTS, IBS, RRS\}$, so that 16•5=80 different simulation experiments have been defined. Each single experiment has been executed with three different seedings leading to 80•3 = 240 performed simulation runs. We have calculated the averagely observed increase of the system flexibility $F^{sys}(\Psi,\beta)$ [8] with respect to the RRS results. System flexibility is expressed as the percentage of all requests that could be served within the given time window. The increase of the overall costs $C(\Psi,\beta)$ compared to the RRS results is also recorded as well as the increase of the system's arrival time nervousness $ATN^{sys}(\Psi,\beta)$ [9]. Arrival time nervousness gives the percentage of all requests released during the simulation experiments, which are not re-scheduled, e.g. for which a once fixed fulfillment time is not revised in later schedule revisions.

Presentation and Discussion of Results. Table 1 contains the averagely observed increase of the system flexibility. Generally, the application of a biased request selection rule leads to an increase of the system flexibility. The highest increases are observed for medium intervention intensities ($\beta \in \{0.4, 0.6\}$). A distance-based request selection (DBS) as well as the sorting by remaining service time (RTS) shows the best performance with respect to F^{sys}. The request selection based upon the resource availability does not lead to convincing results.

The increase of the overall request fulfillment costs is shown in Table 2. With the exception of VAS, all other rules lead to a decrease of the sum of costs. If the maximal intervention intensity is small ($\beta \in \{0.2, 0.4\}$) then the identification of isolated customer site requests and the subcontraction of these requests (IBS) works best. For larger maximal intervention intensities, the subcontraction of requests situated on the edge of the operations area shows the best performance (DBS).

Again, the explicit consideration of the spatial information about a customer site supports the improvement of the hybrid algorithm performance.

Instability (Nervousness) of schedules is a drawback of increased flexibility. Table 3 shows that also in the investigations reported here, the increase of system flexibility achieved by the deployment of DBS, RTS and IBS implies a significant increase in the arrival time nervousness (a larger percentage of request fulfillment times are revised). However, if requests are immediately outsourced because no adequate vehicle is available to serve it (VAS) then there is evidence that re-scheduling decisions in later re-planning stages are prevented.

Table 1: System flexibility increase $F^{sys}(\Psi,\beta)$

sequencing rule Ψ	maximal intervention intensity β			
	0.2	0.4	0.6	0.8
DBS	1,65%	1,73%	1,83%	1,08%
VAS	0,13%	0,00%	0,12%	-0,24%
RTS	1,39%	1,85%	1,34%	1,08%
IBS	0,76%	0,99%	0,73%	0,84%

Table 2: Total cost increase $C(\Psi,\beta)$

sequencing rule Ψ	maximal intervention intensity β			
	0.2	0.4	0.6	0.8
DBS	-0,17%	-4,63%	-7,02%	-7,07%
VAS	5,54%	2,74%	2,75%	3,09%
RTS	4,39%	-0,09%	-4,44%	-5,93%
IBS	-2,02%	-5,00%	-5,39%	-6,59%

Table 3: Arrival time nervousness increase $ATN^{sys}(\Psi,\beta)$

sequencing rule Ψ	maximal intervention intensity β			
	0.2	0.4	0.6	0.8
DBS	0,41%	0,56%	4,34%	4,05%
VAS	-2,45%	-2,23%	-1,93%	-3,29%
RTS	-0,82%	2,42%	8,19%	8,35%
IBS	-0,61%	0,93%	2,41%	1,01%

If the four proposed biased request selection rules are ranked based on their average performance for a given criteria then DBS and ISO outperform VAS and RTS. Since both rules DBS and IBS are based on the evaluation of request discrimination by customer site location information, we conclude that the request selection is positively influenced if the request selection is made by means of these attributes. Thereby, we have verified the research hypothesis given in the introduction partially, since the performance improvement depends upon the applied sequencing rule.

6. Conclusions

In this article we have proposed simple non-iterative rules for the classification of transportation requests. The results observed in computational simulation experiments show that these rules are strong enough to bias the global behavior of the investigated transportation system. If the spatial specifications of customer sites are preferentially used to value the requests, then an increase of the system responsibility as well as a reduction of the request fulfillment costs are observed. However, the arrival time nervousness of the system increases. Future research efforts will be dedicated to the integration to the currently contradicting goals of high flexibility and high stability.

Acknowledgement. This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 “Autonomous Cooperating Logistic Processes” (Subprojects B7 and B9).

Literature

[1] BIERWIRTH, C., Adaptive Search and the Management of Logistics Systems, Kluwer, 2000.

[2] CHOI, T.-M., LI, D., YAN, H., Optimal returns policy for supply chain with e-marketplace, in : International Journal of Production Economics, Vol. 88, 2004, pp. 205—227.

[3] GHIANI, G., Guerriero, F., Laporte, G., Musmanno, R., Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, in: European Journal of Operational Research, 151(1), 2003, pp.1—11.

[4] GUTENSCHWAGER, K., BÖSE, J., VOß, S., Effiziente Prozesse im Kombinierten Verkehr – Ein Neuer Lösungsansatz zur Disposition von Portalkränen, in: Logistik Management, Vol 5(1), 2003, pp. 62—73.

[5] SAUER, J., APPELRATH, H.-J., Scheduling the Supply Chain by Teams of Agents, in: Proceedings of the 36th International Conference on System Sciences, 2002.

[6] SCHÖNBERGER, J. and KOPFER, H., On Decision Model Adaptation in Online Optimization of a Transport System, in: H.-O. Günther, D.C. Mattfeld, L. Suhl (Eds.), Management logistischer Netzwerke, Physica-Verlag, 2007, pp. 361—381.

[7] SCHÖNBERGER, J. and KOPFER, H., Online Decision Making and Automatic Decision Model Adaptation, in: Computers & Operations Research, to appear.

[8] SCHÖNBERGER, J. and KOPFER, H., Transport system responsiveness improvement, in: International Journal of Physical Distribution and Logistics Management, to appear.

[9] SCHÖNBERGER, J. and KOPFER, H., Schedule Nervousness Reduction in Transport Re-Planning, in: Communications of SIWN, Vol. 3, 2008, pp.7—14

[11] SCHÖNBERGER, J. and KOPFER, H., Manipulating the Decision Behaviour of a Subordinate Service Centre by a Closed-Loop Cost Accounting Scheme Control, in: D.C. Mattfeld, H.-O. Günther, L. Suhl, S. Voß (Eds.), Informations- und Kommunikationssysteme in Supply Chain Management, Logistik und Transport, DSOR Beiträge zur Wirtschaftsinformatik, Band 5, Paderborn, 2008, pp. 3-19.

[12] SCHÖNBERGER, J., KOPFER, H., Autonomously Controlled Adaptation of Formal Decision Models – Comparison of Generic Approaches, in: H.-D. Haasis, H.-J. Kreowski, B. Scholz-Reiter (Eds.), Dynamics in Logistics, Springer, Berlin, Heidelberg, 2008, pp. 341—348.

[13] ZEIMPEKIS, V., TARANTILIS, C.D., GIAGLIS, G.M., MINIS, I., Dynamic Fleet Management, Springer, New York, 2007.