Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh

Research School for Operations Management and Logistics

Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh

Murat Firat, Gerhard J. Woeginger

Beta Working Paper series 338

BETA publicatie	WP 338 (working paper)
ISBN	
ISSN	
NUR	982
Eindhoven	November 2010

Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh

Murat Firat[*] Gerhard J. Woeginger[†]

Abstract

Hunsaker and Savelsbergh [Operations Research Letters 30, 2002] discussed feasibility testing for a dial-a-ride problem under maximum wait time and maximum ride time constraints. We show that this feasibility test can be expressed as a shortest path problem in vertex-weighted interval graphs, which leads to a simple linear time algorithm.

Keywords: Dial-a-ride; feasibility check; shortest path; difference constraint system.

1 Introduction

Dial-a-ride problems concern the dispatching of a vehicle to satisfy requests where an item (or a person) has to be picked up from a specific location and has to be delivered to some other specific location. Dial-a-ride problems arise in many practical application areas, as for instance shared taxi services, courier services, and transportation of elderly and disabled persons.

Hunsaker & Savelsbergh [4] analyzed the following feasibility question for a dial-a-ride problem arising in a taxi company: An instance specifies a sequence of $2n+1$ events that have to be served (one after the other and in the given order) by a single vehicle. The first event is the dispatch of the vehicle from a central facility. The remaining $2n$ events are grouped into a set $\mathcal{P}$ of pairs (i,j) with $i<j$. In every pair (i,j) the earlier event i is the pickup and the later event j is the delivery of some fixed item (we stress that the two events in such a pair are not necessarily consecutive in the event sequence). The problem consists in deciding whether there exist $2n+1$ time points for these $2n+1$ events subject to the following three families of constraints.

Time windows: The ith event ($1 \le i \le 2n+1$) must occur during a pre-specified time window between time points α_i and β_i with $\alpha_i \le \beta_i$.

[*] m.firat@tue.nl. Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands

[†] gwoegi@win.tue.nl. Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands

Riding times: The riding time from the ith to the $(i+1)$th event $(1 \le i \le 2n)$ is $\gamma_{i,i+1}$. For every pickup and delivery pair $(i,j) \in \mathcal{P}$, the time from pickup to delivery can be at most $\delta_{i,j} > 0$ time units.

Waiting times: At the ith pickup or delivery location $(2 \le i \le 2n+1)$, the vehicle can wait for at most ω_i times units before departing.

We note that there are three points in which our problem description deviates from the one by Hunsaker & Savelsbergh [4]. First, our riding time bounds $\delta_{i,j}$ can be arbitrary numbers, whereas the riding time bounds in [4] are proportional to the distances between pickup location and delivery location. In this respect, our model is slightly more general and contains the model in [4] as a special case. Secondly, our waiting time bounds ω_i depend on the event, whereas the waiting time bounds in [4] all are identical. Again this is a slight extension of the model in [4], which also is mentioned in the discussion section of [4]. Thirdly, the problem in [4] also incorporates item sizes and a capacity bound for the vehicle. These capacity constraints are independent of the timing and riding constraints, and they can be checked separately in $O(n)$ overall time. This independent subproblem has been discussed and settled in [4], and there is no reason for re-discussing it here.

Hunsaker & Savelsbergh [4] design a sophisticated linear time algorithm that tests for the existence of a schedule that satisfies the three constraint families listed above. Tang, Kong, Lau & Ip [5] identify a crucial gap in the algorithm of [4], and they also provide a concrete counter-example where the algorithm declares a feasible instance to be infeasible. As a partial repair [5] provides another algorithm for the problem with a weaker quadratic running time $O(n^2)$. We note that such a quadratic running time is too slow for practical applications: The feasibility test shows up as a subproblem in improvement-based algorithms, and has to be performed many times.

Contribution of this note. We formulate the dial-a-ride feasibility test of [4] as a system of linear inequalities (Sections 3 and 4), which by standard methods can be rewritten into a system of difference constraints (Section 5). By carefully analyzing the structure of these difference constraints (Section 6), we then transform the problem into a shortest path problem in a vertex-weighted interval graph. All in all, this yields the desired linear time $O(n)$ algorithm for the feasibility test.

2 Preliminaries

Suppose that $\alpha_{i+1} < \alpha_i$ for some i with $1 \le i \le 2n$. Since the vehicle cannot serve the ith event before time α_i, it cannot arrive at the $(i+1)$th location before time α_i. Hence we may update the data as $\alpha_{i+1} := \alpha_i$. All updates (for all values of i) can be performed by a single $O(n)$ time pass over the locations in increasing order of i. A symmetric argument shows that whenever $\beta_{i+1} < \beta_i$ holds for some i with $1 \le i \le 2n$, then we may update $\beta_i := \beta_{i+1}$. Furthermore, all such updates can be done during a single $O(n)$ time pass over the locations in decreasing order of i.

Therefore we assume throughout that the left and right endpoints of the time intervals form two non-decreasing sequences $\alpha_1 \le \alpha_2 \le \cdots \le \alpha_{2n+1}$ and $\beta_1 \le \beta_2 \le \cdots \le \beta_{2n+1}$.

3 Linear equations and inequalities

We formulate the dial-a-ride problem as a system of linear equations and inequalities. The ith event ($1 \le i \le 2n+1$) is described by three real variables: The arrival time A_i and the departure time D_i of the vehicle at the location of the ith event, and the time point E_i at which the actual pickup/delivery occurs. For the first event, we identify the variables E_1 and D_1 so that they coincide with the dispatch time of the vehicle from the central facility.

$$A_1 = \alpha_1 \quad \text{and} \quad E_1 = D_1 \tag{1}$$

Clearly the ith event must fit between the arrival and the departure time of the vehicle.

$$A_i \ \le \ E_i \ \le \ D_i \qquad \text{for } i = 2, \ldots, 2n+1. \tag{2}$$

Now let us express the constraints of the problem. The ith event must occur during its time window $[\alpha_i, \beta_i]$.

$$\alpha_i \ \le \ E_i \ \le \ \beta_i \qquad \text{for } i = 1, \ldots, 2n+1. \tag{3}$$

The riding time $\gamma_{i,i+1}$ from the ith to the $(i+1)$th event yields

$$A_{i+1} \ = \ D_i + \gamma_{i,i+1} \qquad \text{for } i = 1, \ldots, 2n. \tag{4}$$

For every pickup and delivery pair $(i,j) \in \mathcal{P}$, the time from pickup to delivery is constrained by

$$E_j \ \le \ E_i + \delta_{i,j} \qquad \text{for all pairs } (i,j) \in \mathcal{P}. \tag{5}$$

Finally, the waiting time constraints yield

$$D_i \ \le \ A_i + \omega_i \qquad \text{for } i = 2, \ldots, 2n+1. \tag{6}$$

The dial-a-ride instance has a feasible solution, if and only if the linear system (1)–(6) with $O(n)$ constraints has a feasible solution over the real numbers.

4 More linear inequalities

Our next step is to rewrite the system (1)–(6) into an equivalent but simpler system centered around a new set of variables: For $i = 1, \ldots, 2n+1$ we introduce the non-negative variable x_i to measure the total waiting time between time point α_1 and time point D_i (that is, the total time that the vehicle did not spent on driving before departing from the ith location). Furthermore, for $1 \leq i \leq j \leq 2n+1$ we introduce the constant Γ_i to denote the overall riding time to move through the locations $1, 2, \ldots, i$.

$$\Gamma_i \;=\; \sum_{k=1}^{i-1} \gamma_{k,k+1}.$$

Then the arrival times $A_2, \ldots, A_{2n+1}$ and the departure times $D_1, \ldots, D_{2n+1}$ can be rewritten as

$$A_i = x_{i-1} + \Gamma_i \qquad \text{and} \qquad D_i = x_i + \Gamma_i. \tag{7}$$

Each of the events $E_2, \ldots, E_{2n+1}$ is either a pickup or a delivery that is constrained by (2), (3), and (5). Hence it must occur between the lower bound $\max\{A_i, \alpha_i\}$ and the upper bound $\min\{D_i, \beta_i\}$. The resulting interval is non-empty, if and only if $A_i \leq D_i$ and $A_i \leq \beta_i$ and $\alpha_i \leq D_i$ hold. Thus Fourier-Motzkin elimination of E_i yields for $i = 2, \ldots, 2n+1$ the constraints

$$x_{i-1} \leq x_i \qquad \text{and} \qquad x_{i-1} \leq \beta_i - \Gamma_i \qquad \text{and} \qquad \alpha_i - \Gamma_i \leq x_i. \tag{8}$$

If the ith event is a pickup, then we may delay it as much as possible by setting $E_i := \min\{D_i, \beta_i\}$. If the jth event is a delivery, then we may schedule it as early as possible and set $E_j := \max\{A_j, \alpha_j\}$. Then (5) means for every pickup and delivery pair $(i, j) \in \mathcal{P}$ the four conditions that $A_j \leq D_i + \delta_{i,j}$, that $A_j \leq \beta_i + \delta_{i,j}$, that $\alpha_j \leq D_i + \delta_{i,j}$, and that $\alpha_j \leq \beta_i + \delta_{i,j}$. The last condition does not depend on any variable (and if it is violated, then the system is trivially infeasible). The other three conditions yield the following constraints.

$$x_{j-1} \;\leq\; x_i + \delta_{i,j} + \Gamma_i - \Gamma_j \qquad \text{for all } (i,j) \in \mathcal{P} \tag{9}$$

$$x_{j-1} \;\leq\; \beta_i + \delta_{i,j} - \Gamma_j \qquad \text{for all } (i,j) \in \mathcal{P} \tag{10}$$

$$\alpha_j - \delta_{i,j} - \Gamma_i \;\leq\; x_i \qquad \text{for all } (i,j) \in \mathcal{P} \tag{11}$$

The remaining constraints (1), (4), (6) are handled as follows. Constraints (1) and (3) yield

$$0 \;\leq\; x_1 \;\leq\; \beta_1 - \alpha_1. \tag{12}$$

Constraint (4) is satisfied because of (7), and becomes vacuous. Finally the waiting time constraints (6) translate into

$$x_i \;\leq\; x_{i-1} + \omega_i \qquad \text{for } i = 2, \ldots, 2n+1. \tag{13}$$

The dial-a-ride instance has a feasible solution, if and only if the linear system (8)–(13) with $O(n)$ constraints has a feasible solution over the real numbers.

5 Difference constraint systems

Every inequality in (8)–(13) is either an upper bound constraint $x_i \le U_i$, or a lower bound constraint $L_i \le x_i$, or a difference constraint $x_j - x_i \le D_{i,j}$. By applying a standard trick, we will now transform all upper and lower bound constraints into difference constraints.

For this purpose, we introduce two new variables x_0 and x_{2n+2}. Variable x_0 represents the value 0, and hence is a lower bound on all other variables. Variable x_{2n+2} represents the value $K := \beta_{2n+1} - \alpha_1$, and hence is an upper bound for all other variables. We create the two new constraints

$$x_{2n+2} - x_0 \le K \qquad \text{and} \qquad x_0 - x_{2n+2} \le -K, \tag{14}$$

which together enforce $x_{2n+2} - x_0 = K$. Every upper bound constraint $x_i \le U_i$ in (8)–(13) is replaced by a corresponding constraint

$$x_i - x_0 \ \le \ U_i. \tag{15}$$

Every lower bound constraint $L_i \le x_i$ in (8)–(13) is replaced by a corresponding constraint

$$x_{2n+2} - x_i \ \le \ K - L_i. \tag{16}$$

We will refer to (14), to the new difference constraints (15) and (16), and to the old difference constraints in (8), (9), (13) short as the difference constraint system DCS.

Lemma 1 *The following four statements are pairwise equivalent.*

(i) The original dial-a-ride instance has a feasible solution.

(ii) The system (8)–(13) has a feasible solution over the real numbers.

(iii) DCS has a feasible solution with $x_0 = 0$ and $x_{2n+2} = K$.

(iv) DCS has a feasible solution.

There is a close connection between difference constraint systems and negative-weight cycles in directed graphs; see for instance Section 24.4 of Cormen, Leiserson, Rivest & Stein [2]. We create for every variable x_i ($0 \le i \le 2n+2$) a corresponding vertex i. We create for every difference constraint $x_j - x_i \le D_{i,j}$ an arc from vertex i to vertex j with weight $D_{i,j}$. It is well-known and easy to see (see for instance Theorem 24.9 in [2]) that the underlying difference constraint system has a feasible solution, if and only if the corresponding directed graph G does not contain any negative-weight cycles.

In our case, the directed graph G has $O(n)$ vertices and $O(n)$ arcs. Hence a straightforward application of the Bellman-Ford algorithm or of the Goldberg-Radzik algorithm would yield an $O(n^2)$ time feasibility test.

6 A linear time feasibility test

To get a linear time algorithm, we look a little bit deeper into the structure of DCS and the corresponding directed graph G. A difference constraint $x_j - x_i \leq D_{i,j}$ is a *forward* constraint (and the corresponding arc $i \to j$ is a forward arc), if $i < j$ holds. Otherwise we are dealing with a *backward* constraint (and a corresponding backward arc). Now let us go through all constraints in DCS.

- The difference constraints in (8) are of the form $x_{i-1} - x_i \leq 0$. They are backward constraints, and their arc weights are always zero.
- The constraints in (9) are forward constraints. If $\delta_{i,j} < \Gamma_j - \Gamma_i$ then the system is infeasible. Hence we may assume that all corresponding arc weights are non-negative.
- The constraints (13) are forward constraints, and the corresponding arc weights ω_i are non-negative.
- In (14) we have one forward constraint with positive arc weight, and one backward constraint with negative arc weight.
- The difference constraints in (15) arise from upper bounds, and are forward constraints. We may assume that all corresponding arc weights are non-negative (since otherwise the system would be infeasible).
- The difference constraints in (16) arise from lower bounds, and are forward constraints. Again, we assume that all corresponding arc weights are non-negative (as otherwise the system would be infeasible).

Summarizing, all forward arcs have non-negative weights, and with a single exception all backward arcs have weight zero and are of the form $i \to i-1$. The only arc with negative weight is the backward arc from vertex $2n+2$ to vertex 0 in (14). Hence every cycle of negative weight must consist of this arc plus some directed path from vertex 0 to vertex $2n+2$. Recall that the DCS is infeasible, if and only if graph G contains a negative-weight cycle, which is true if and only if the shortest path from vertex 0 to vertex $2n+2$ along arcs with non-negative weights has length strictly smaller than K. This observation in combination with fast shortest path algorithms in directed graphs [3] yields a time complexity of $O(n \log n)$.

Our next goal is to establish a connection to interval graphs, which will yield a linear time complexity. With every forward arc $i \to j$ we associate the closed interval $[i, j]$.

Lemma 2 *Among all shortest paths from vertex* 0 *to vertex* $2n+2$ *(that only use arcs with non-negative weights) let* P^* *be a path with the smallest number of forward arcs. For* $k \geq 1$*, let* $i_k \to j_k$ *denote the* k*th forward arc traversed by* P^**. Then for any pair of consecutive forward arcs* $i_k \to j_k$ *and* $i_{k+1} \to j_{k+1}$*, the two associated intervals* $[i_k, j_k]$ *and* $[i_{k+1}, j_{k+1}]$ *have non-empty intersection.*

Proof. Since P^* moves from vertex j_k to vertex i_{k+1} along backward arcs, we get $i_{k+1} \leq j_k$. If the two associated intervals do not intersect, we must have $i_{k+1} < j_{k+1} < i_k < j_k$. But then P^* could simply skip the forward arc $i_k \to j_k$, and move from i_k to i_{k+1} along backwards arcs at zero cost. This would yield another shortest path with a smaller number of traversed forward arcs. ☐

Lemma 3 *Consider a sequence of forward arcs $i_k \to j_k$ $(k = 1, \ldots, p)$ with overall weight W, such that the first arc starts in $i_1 = 0$ and the last arc ends in $j_p = 2n + 2$, and such that for any two consecutive arcs in the sequence the associated intervals have non-empty intersection. Then the directed graph contains a path from vertex 0 to vertex $2n + 2$ of weight W.*

Proof. As intervals $[i_k, j_k]$ and $[i_{k+1}, j_{k+1}]$ intersect, we get $j_k \geq i_{k+1}$. Hence we can move from vertex j_k to vertex i_{k+1} by a sequence of backward arcs. ☐

Finally we construct a vertex-weighted interval graph G^*. For every forward arc with weight w, the interval graph G^* contains the associated interval with weight w. Furthermore G^* contains the degenerate intervals $[0, 0]$ and $[2n + 2, 2n + 2]$ both of weight 0. Lemmas 2 and 3 imply that the length of the shortest path from 0 to $2n+2$ in the directed graph G (measured in arc weights) equals the length of the shortest path in the interval graph G^* from $[0, 0]$ to $[2n + 2, 2n + 2]$ (measured in interval/vertex weights). Atallah, Chen & Lee [1] show that the length of the shortest path in a vertex-weighted interval graph can be computed in linear time:

Proposition 4 *(Atallah, Chen & Lee [1])*
Given a set of intervals with weights, an ordering of these intervals according to their left endpoints, and an ordering of these intervals according to their right endpoints, the single-source shortest path problem can be solved in linear time. ☐

The single-source shortest path problem consists in computing the shortest paths from a given source-interval to all other intervals, where the length of a path is the sum of all interval weights along the path.

Since the endpoints of all intervals are intervals in the range 0 to $2n + 2$, it is easy to sort these intervals according to their left or right endpoints in linear time $O(n)$; this can for instance be done by counting sort or by some variant of bucket sort (see Section 8 of Cormen, Leiserson, Rivest & Stein [2]). Altogether this yields the main result of our paper.

Theorem 5 *The feasibility test for the dial-a-ride problem of Hunsaker and Savelsbergh under time window constraints, riding time constraints, and waiting time constraints can be performed in $O(n)$ time.* ☐

7 Final remarks

Up to this point, we only discussed how to decide whether a given instance is feasible. But if an instance is feasible, then we can also explicitly construct a corresponding schedule in $O(n)$ time: We extend the interval graph G^* by adding the $2n + 1$ degenerate intervals

$[i, i]$ with $1 \leq i \leq 2n+1$ to it, where interval $[i, i]$ has weight 0 and corresponds to variable x_i in the difference constraint system DCS.

Then we use Proposition 4 to compute the shortest path lengths y_i from the source interval $[0, 0]$ to all intervals $[i, i]$ in linear time. It is easy to see that setting $x_i := y_i$ yields a feasible solution for the difference constraint system. Finally, by some straightforward backwards calculations we can determine from this the corresponding feasible solutions for the inequality systems in Sections 3 and 4.

Acknowledgement. We thank the referee and the handling associate editor for helpful comments that improved the presentation of the paper.

This research has been supported by the Netherlands Organization for Scientific Research (NWO), grant 639.033.403; by DIAMANT (an NWO mathematics cluster); by France Telecom/TUE Research agreement No. 46145963.

References

[1] M.J. Atallah, D.Z. Chen, and D.T. Lee (1995). An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications. *Algorithmica 14*, 429–441.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein (2001). *Introduction to Algorithms.* MIT Press.

[3] M.L. Fredman and R.E. Tarjan (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM 34*, 596–615.

[4] B. Hunsaker and M. Savelsbergh (2002). Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters 30*, 169–173.

[5] J. Tang, Y. Kong, H. Lau, and A.W.H. Ip (2010). A note on "Efficient feasibility testing for dial-a-ride problems". *Operations Research Letters 38*, in press.

Working Papers Beta 2009 - 2010

nr.	Year	Title	Author(s)
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan
330	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326		Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten

322	2010	[The selective use of emergency shipments for service-contract differentiation]	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	[Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse]	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	[Preventing or escaping the suppression mechanism: intervention conditions]	Nico Dellaert, Jully Jeunet.
319	2010	[Hospital admission planning to optimize major resources utilization under uncertainty]	
318		[Minimal Protocol Adaptors for Interacting Services]	R. Seguel, R. Eshuis, P. Grefen.
317	2010	[Teaching Retail Operations in Business and Engineering Schools]	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
316	2010	[Design for Availability: Creating Value for Manufacturers and Customers]	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
315	2010	[Transforming Process Models: executable rewrite rules versus a formalized Java program]	Pieter van Gorp, Rik Eshuis.
314	2010	[Getting trapped in the suppression of exploration: A simulation model]	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
313	2010	[A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows]	S. Dabia, T. van Woensel, A.G. de Kok
	2010		
312	2010	[Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures]	Osman Alp, Tarkan Tan
311	2010	[In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints]	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	[The state of the art of innovation-driven business models in the financial services industry]	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	[Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case]	R. Seguel, P. Grefen, R. Eshuis
308	2010	[Effect of carbon emission regulations on transport mode selection in supply chains]	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	[Interaction between intelligent agent strategies for real-time transportation planning]	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	[Internal Slackening Scoring Methods]	Marco Slikker, Peter Borm, René van den Brink
305	2010	[Vehicle Routing with Traffic Congestion and	A.L. Kok, E.W. Hans, J.M.J. Schutten,

		Drivers' Driving and Working Rules	W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak

284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	R. Seguel, R. Eshuis, P. Grefen
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl