MIPLIB Truckload PDPTW Instances Derived from a Real-World Drayage Case

F. Jordan Srour, Tamas Mahr, Mathijs de Weerdt, and Rob Zuidwijk

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2010-036-LIS
Publication	October 2010
Number of pages	18
Persistent paper URL	http://hdl.handle.net/1765/20883
Email address corresponding author	rzuidwijk@rsm.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus Universiteit Rotterdam P.O.Box 1738 3000 DR Rotterdam, The Netherlands Phone: + 31 10 408 1182 Fax: + 31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

ABSTRACT AND KEYWORDS	
Abstract	This paper describes five sets of 33 Mixed Integer Problem instances each, for a total of 165 instances, derived from a real-world full-truckload pick-up and delivery problem with time windows at the Port of Rotterdam. These instances represent 33 individual days of data encompassing 65 jobs and 40 trucks. We report, in this paper, on the structure of the real-world problem, the mechanism by which the real data was transformed into the test instances, the Mixed Integer Programming formulation used to solve these instances, the results obtained, and sources in the literature describing alternative uses for these instances.
Free Keywords	Mixed Integer Programming, problem instances, vehicle routing, drayage, online routing
Availability	The ERIM Report Series is distributed through the following platforms: Academic Repository at Erasmus University (DEAR), DEAR ERIM Series Portal Social Science Research Network (SSRN), SSRN ERIM Series Webpage Research Papers in Economics (REPEC), REPEC ERIM Series Webpage
Classifications	The electronic versions of the papers in the ERIM report Series contain bibliographic metadata by the following classification systems: Library of Congress Classification, (LCC) LCC Webpage Journal of Economic Literature, (JEL), JEL Webpage ACM Computing Classification System CCS Webpage Inspec Classification scheme (ICS), ICS Webpage

MIPLIB Truckload PDPTW Instances Derived from a Real-World Drayage Case

F. Jordan Srour
Faculty of Engineering and Architecture, The American University of Beirut,
Beirut, Lebanon
{srourf@hotmail.com},
Tamás Máhr
TU Delft,
P.O. Box 5031, 2600 GA Delft, The Netherlands
{tamas.mahr@gmail.com},
Mathijs de Weerdt
TU Delft,
P.O. Box 5031, 2600 GA Delft, The Netherlands
{M.M.deWeerdt@tudelft.nl},
Rob Zuidwijk
Rotterdam School of Management, Erasmus University,
Burg. Oudlaan 50, 3062 PA Rotterdam, The Netherlands
{rzuidwijk@rsm.nl}

Abstract

This paper describes five sets of 33 Mixed Integer Problem instances each, for a total of 165 instances, derived from a real-world full-truckload pick-up and delivery problem with time windows at the Port of Rotterdam. These instances represent 33 individual days of data encompassing 65 jobs and 40 trucks. We report, in this paper, on the structure of the real-world problem, the mechanism by which the real data was transformed into the test instances, the Mixed Integer Programming formulation used to solve these instances, the results obtained, and sources in the literature describing alternative uses for these instances.

Keywords: Mixed Integer Programming; problem instances; vehicle routing; drayage; online routing

1 Introduction

In an effort to appreciate the relative merits of a decentralized route planning system versus a centralized route planning system, we compared the performance of an agent-based approach to an on-line optimization approach for routing in the drayage industry. (Drayage commonly refers to the transport of containerized cargo, within a limited geographic range, to and from port or rail terminals and inland locations.) The results of that comparison can be seen in the article entitled "Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty" appearing in *Transportation Research, Part C: Emerging Technologies* (Máhr et al., 2010).

At the heart of the comparative study were five sets of truckload pick-up and delivery problems with time windows. These problem instances were used in an on-line or real-time manner in the context of the aforementioned article. However, static versions were also studied in an effort to obtain a lower bound for the on-line results. These static problem instances are interesting in their own right as they represent a set of Mixed Integer Problems derived from a real-world setting. Hence, these instances have been made publicly available via the MIPLIB (`http://miplib.zib.de/`). This report is intended to supplement the problem instances by describing the structure of the real-world problem, the mechanism by which the real data was transformed into the test instances, the Mixed Integer Programming formulation used to solve these instances, the results obtained, and sources in the literature describing the use of these instances for benchmarking on-line planning approaches.

2 Description of Real-World Setting

The data used to construct the problem instances, described in this paper, come from the drayage operations of a Dutch logistics service provider (LSP). The LSP dedicates a portion of its business to draying refrigerated ("reefer") containers from/to the Port of Rotterdam to/from various customer locations in the Netherlands. Approximately 40 trucks transport an average of 65 containers per day in this operation. Given the number and geographic range of jobs, each truck, beginning and ending at a home base location near the port, can serve approximately two, and maximally three, jobs per day.

In general, the containers arrive on container ships arranged by customers. They are off-loaded at sea terminals, where trucks must then pick them up. The containers are then transported to their destination at the customer, where they are emptied. The empty containers are later returned to a sea terminal. In reality, because reefers are considered high-value equipment, the same truck waits with the container until it is emptied and then returns it to a sea terminal. The return terminal may be the same terminal from which the container originated or it may be a different terminal. For export containers the sequence is the same, with the exception that the containers are not emptied, but loaded at the customer's location. At each location there are time windows within which trucks can make their visits. At sea terminals the time windows correspond to the opening hours of the terminal. At customer sites, the time windows are defined by the customers.

The primary objective of the LSP is to route their uniform fleet of approximately 40 trucks on the Netherlands' road network at lowest cost without violating time windows. We now describe the exact manner in which the instances were constructed from the data provided by the LSP.

3 Instance Construction

In this section, we describe how our data was inspired and fed by the operations of the Dutch LSP. Recall that the LSP is transporting comparatively high-valued reefer containers. As such, the trucks always wait at customer sites for the containers to be (un)loaded, and they never exchange containers. We therefore handle each pick-up, delivery, and return sequence as one job. Moreover, some customers have more than one job serviced in a day. Nevertheless, we handle each job separately, as if they all belonged to different customers. Each job is specified by two data vectors — one spatial and one temporal.

The spatial vector contains the location of the pick-up terminal, the customer site, and the return terminal. This data was derived from a set of operational data tables provided by the LSP. In all, we were given data from January 2002 to October 2005 as well as from January 2006 through March 2006. The tables represented jobs that were planned to be served on a given day. Unfortunately, the exact timing of the jobs each day was nearly absent from the data. Further problems were presented by some of the addresses that referred to postal boxes instead of real customer or terminal locations; therefore these points had to be pruned from the data. Nevertheless, after a preliminary review of the data, we could conclude that, on average, 65 jobs were served in a day, at customer and terminal locations associated with less than 25 distinct zipcodes. The rare timing information in the database, coupled with observations made by the human planners, suggested that the jobs were served uniformly throughout the day. Using these parameters, we extracted a random sample of appropriately defined jobs from the original data-set in order to generate a set of 33 days with 65 jobs per day using the locations in the sample. Note, we consider each day as a single instance. Thus, there are no jobs that persist in the planning system from one day (or instance) to the next. Figure 1 depicts the geography of the Netherlands and the full set of locations represented in our data.

The temporal vector is comparatively more complex — containing three data types: data on time windows, data on service times, and data on job arrival. The data on time windows includes the terminal operating time windows and the customer time window. The data on service times includes the service time required at the three job locations. Finally, the data on job arrival includes one element — the time the job is announced in the planning system. As mentioned earlier, such timing information was sparsely recorded in the data tables. Therefore this part of the job descriptions was entirely generated based on the experiences of the human planners.

To standardize the data for our experimental purposes, we specified time windows at all locations as follows: terminals are open for pick up between 6am and 6pm, and for return between 6am to 5:59am on the next day. The wide return time windows reflect the practice that trucks can bring containers to the terminals on the following day, if they were too late on the same day. These time windows are the same for all jobs. Delivery time windows, that is time windows at the customer location, are set to two hour intervals, and their start times are distributed uniformly over the working day between 8am and 5pm. Figure 2 displays the number of open time windows for all jobs at any time point of the working day. Since time windows open regularly and stay open for two hours, the number of open time windows gradually builds up reaching a maximum between 12am and 2pm. After that, the number of open time windows, and therefore the number of jobs requiring service before the end of the day decreases.

The service time data type refers to the time trucks need to complete service at the different locations. When a truck arrives at a sea terminal or a customer, it spends some time to pick up, to deliver, or to return a container. The length of this time depends on various factors. Picking up a container for example, can be delayed by customs clearing,

Figure 1: All locations in the Netherlands. Black markers indicate customer locations; grey markers indicate terminal locations; and the white marker indicates the home terminal of the LSP

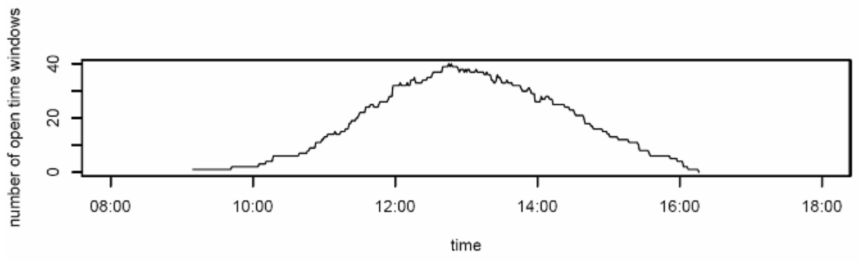

Figure 2: Number of open time windows for all jobs throughout the working day.

paperwork, or problems with putting the container on the truck. Emptying a container at the customer can be quick if the customer is ready to unload the goods, but it can be delayed if a warehouse is very busy. Similarly, when a container is returned, technical issues may delay the trucks. In discussions with the LSP, it seems that the human planners, by experience, allocate one hour for picking up, one hour for delivering, and half an hour for returning a container. As such, the instances set all service time values to these times for all jobs.

The job arrival data type refers to the time after which a container may be planned. Before this time, the container is not available for service; after this time, the container may be planned for service. In the R0_* instances, all job arrival times are set to the start of the day, 6am. In the R25_* (R50_*, R75_*, and R100_*) instances, a randomly selected set of jobs, totaling 25% (50%, 75%, and 100%, respectively) of the jobs, had their job arrival data element set to time beyond the start of the day. This later job arrival time was calculated by subtracting two hours from the time at which a truck must depart the pick-up terminal location in order to travel to the customer location, and arrive at the start of the time-window associated with the customer location of that job.

In all instances that we constructed, we added a homogeneous fleet of 40 trucks starting at a home base location close to the Port of Rotterdam. Although the number of trucks used by human planners varies each day, we chose to use 40 trucks, because this proved to be enough to solve each instance. In general, however, using more trucks would not yield different results; using fewer trucks would yield a higher rejection rate.

Within the instances, each job requires, on average, approximately 4.2 hours of loaded distance. The goal is to assign jobs to the trucks in such a way that minimizes the total routing costs. These costs consist of time traveling empty plus the penalty for rejected jobs. Jobs may be rejected when they cannot be served within the time restrictions. In our instances, the penalty for rejecting a job equals the loaded time of that job. The loaded time of a job is the time from the start of the pick-up action to the end of the return action

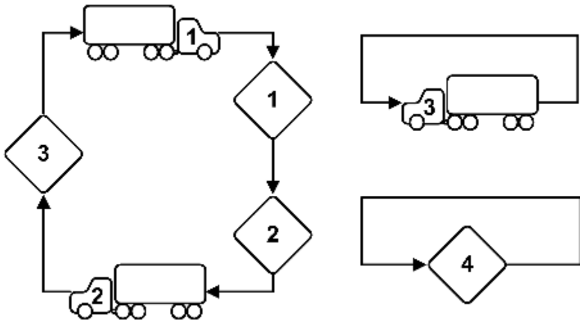

Figure 3: Cycles in the MIP solution structure.

— including all loading, unloading, and traveling time. This is an appropriate penalty for a rejected job as it represents the profit lost in not serving the job. Admittedly, rejecting a job may also yield a loss in customer good will or relations. Given the difficulty in quantifying this loss, we however choose to use the loaded time as a low estimate of the cost associated with job rejection. As our instances represent only one planning day, each, rejected jobs are simply rejected, although in practice they are reconsidered for service the next day. When the routing is optimal, and all jobs are available for planning at the start of the day, the average empty time per job is approximately 25 minutes (or 27 hours total per day).

4 MIP Formulation

The mixed integer programming formulation of this problem, as originally presented in Máhr et al. (2010), is nearly identical to that proposed by Yang et al. (1999). Before introducing the notation and mathematical formulation for this problem, we begin with a small example to illustrate exactly how Yang et al.'s MIP works to exploit the structure of this truckload pick-up and delivery problem with time windows. Imagine a scenario with three trucks and four jobs. The model of Yang et al. is constructed such that it will find a set of least cost cycles describing the order in which each truck should serve the jobs. For example, as depicted in Figure 3, the outcome may be a tour from truck 1 to job 1, then job 2, then truck 2, then job 3, then back to truck 1. This would indicate that truck 1 serves job 1 and 2, while truck 2 serves job 3. The cycle including only truck 3 indicates that truck 3 remains idle. Similarly, the cycle including only job 4 indicates that job 4 is rejected.

Given this problem description, we designate the following notation for the given information.

K	the total number of vehicles available in the fleet.
N	the total number of known demands.
d_{ij}	the travel time required to go from demand i's return terminal to the pick-up terminal of demand j. Note, if $i = j$ then the travel time d_{ii} represents the loaded distance of job i; this distance includes the time from pick-up at the originating terminal to completion of service at the return terminal.
d_{0i}^k	the travel time required to move from the location where truck k started to the pick-up terminal of demand i.
d_{iH}^k	the travel time from the return terminal of demand i to the home terminal of vehicle k.
v^k	the time vehicle k becomes available.
τ_i^-	earliest possible arrival at demand i's pick-up terminal.
τ_i^+	latest possible arrival at demand i's pick-up terminal.
M	a large number set to be $2 \cdot \max_{i,j}\{d_{ij}\}$.

Note that τ_i^- and τ_i^+ are calculated to ensure that all subsequent time windows (at the customer location and return terminal) are respected. Specifically, τ_i^- is calculated by selecting the *maximum* of 1) the pick-up terminal's opening time (6am), 2) the job's arrival time, and 3) the time obtained by taking the start time of the randomly generated delivery (or customer location) time-window, subtracting from it the travel time required between the pick-up terminal's location and the customer location plus the service time required at the pick-up terminal location. The value of τ_i^+ is similarly calculated by selecting the *minimum* of 1) the return terminal's closing time (5:59am one day later), 2) the time obtained by taking the end time of the randomly generated delivery (or customer location) time-window, subtracting from it the travel time required between the pick-up terminal's location and the customer location plus the service time required at the pick-up terminal location, and 3) the time obtained by taking the end time of the return terminal's closing time, subtracting from it all travel times (between the pick-up terminal and customer location as well as the customer location and return terminal) along with all service times (at the pick-up terminal and the customer location).

Given the problem of interest, we specify the following two variables.

x_{uv} a binary variable indicating whether arc (u, v) is used in the final routing; $u, v = 1, \ldots, K + N$.

δ_i a continuous variable designating the time of arrival at the pick-up terminal of demand i.

Using the notation described above, we formulate a MIP that explicitly permits job rejections, based on the loaded distance of a job.

$$\begin{aligned} \min \quad & \textstyle\sum_{k=1}^{K} \sum_{i=1}^{N} d_{0i}^{k} x_{k,K+i} + \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{K+i,K+j} \\ & + \textstyle\sum_{i=1}^{N} \sum_{k=1}^{K} d_{iH}^{k} x_{K+i,k} \end{aligned}$$

such that

$$\sum_{v=1}^{K+N} x_{uv} = 1 \qquad \forall u = 1, \ldots, K + N \qquad (1)$$

$$\sum_{v=1}^{K+N} x_{vu} = 1 \qquad \forall u = 1, \ldots, K + N \qquad (2)$$

$$\delta_i - \sum_{k=1}^{K} (d_{0i}^{k} + v^{k}) x_{k,K+i} \geq 0 \qquad \forall i = 1, \ldots, N \qquad (3)$$

$$\begin{aligned} & \delta_j - \delta_i - M x_{K+i,K+j} + \\ & \quad (d_{ii} + d_{ij}) x_{K+i,K+i} \\ & \quad \geq d_{ii} + d_{ij} - M \end{aligned} \qquad \forall i, j = 1, \ldots, N \qquad (4)$$

$$\tau_i^{-} \leq \delta_i \leq \tau_i^{+} \qquad \forall i = 1, \ldots, N \qquad (5)$$

$$\delta_i \in \mathbb{R}^{+} \qquad \forall i = 1, \ldots, N \qquad (6)$$

$$x_{uv} \in \{0, 1\} \qquad \forall u, v = 1, \ldots, K + N \qquad (7)$$

In words, the objective of this model is to minimize the total amount of time spent traveling without a profit generating load. Specifically, we wish to minimize the penalty incurred from rejecting jobs, time spent traveling empty to pick up a container, between containers and when returning to the home depot. This objective is subject to the following seven constraints:

(1) Each demand and vehicle node must have one and only one arc entering.

(2) Each demand and vehicle node must have one and only one arc leaving.

(3) If demand i is the first demand assigned to vehicle k, then the start time of demand i (δ_i) must be later than the available time of vehicle k plus the time required to travel from the available location of vehicle k to the pick up location of demand i.

(4) If demand i follows demand j then the start time of demand j must be later than the start time of demand i plus the time required to serve demand i plus the time required to travel between demand i and demand j; if however, demand i is rejected, then the pick up time for job i is unconstrained.

(5) The arrival time at the pick up terminal of demand i must be within the specified time windows. (Note, this constraint prevents a truck from arriving early or arriving late to a demand i.)

(6) δ_i is a positive real number.

(7) x_{uv} is binary.

Mathematically this model specification serves to find the least-cost (in terms of time) set of cycles that includes all nodes given in the set $\{1, \dots, K, K+1, \dots, K+N\}$. We define $x_{uv}, (u, v = 1, \dots, K+N)$ as a binary variable to indicate whether arc (u, v) is selected in one of the cycles. These tours require interpretation in terms of vehicle routing. This is done by noting that node $k, (1 \leq k \leq K)$ represents the vehicle k and node $K+i, (1 \leq i \leq N)$ corresponds to demand i. Thus, each tour that is formed may be seen as a sequential assignment of demands to vehicles respecting time window constraints.

5 Results

The results presented in this section are intended to provide additional information about the instances as well as the best known results. Table 1 indicates the number of rows and columns along with the variable and constraint types for each of the R* groups of instances. The variable and constraint type labels are consistent with those used in the MIPLIB (`http://miplib.zib.de/`). Tables 2-6 present the MIP objective value, the number of trucks required for the optimal solution, and the runtime associated with each instance in the groups R0, R25, R50, R75 and R100. These results were obtained using the scip solver (`http://scip.zib.de/`) on AMD Opteron(64bit) machines with two processors (cores) and the ClusterVisionOS 2.1 operating system based on Scientific Linux 4.3.

Table 1: Columns, rows, variable and constraint type for R* instances.

			Variable Type		Constraint Type			
Name	Cols	Rows	Bin	Con	PAR	M01	VLB	VUB
R100_*	11090	4630	11025	65	210	4290	65	65
R75_*	11090	4630	11025	65	210	4290	65	65
R50_*	11090	4630	11025	65	210	4290	65	65
R25_*	11090	4630	11025	65	210	4290	65	65
R0_*	11090	4630	11025	65	210	4290	65	65

Table 2: Objective value and runtimes for all R0_* instances.

Name	Obj. Val.	No. of Trucks	Runtime (seconds)
R0_1	117090.168457	28	14.83
R0_2	98740.531654	26	2644.59
R0_3	106170.685444	27	5.77
R0_4	100450.539322	26	54.52
R0_5	106938.777351	27	14.16
R0_6	105661.961803	32	10.2
R0_7	90129.760437	25	18.93
R0_8	94436.906662	27	47.42
R0_9	90469.112320	24	50.87
R0_10	89487.591820	26	3748.2
R0_11	105567.590160	28	14.69
R0_12	92214.161282	26	16.18
R0_13	94298.388119	27	15.98
R0_14	103815.505371	28	71.09
R0_15	90663.248978	26	941.54
R0_16	93747.076149	27	15.08
R0_17	101808.676586	29	13.29
R0_18	110661.940498	27	15.9
R0_19	101536.619396	29	12.54
R0_20	92051.905689	26	550.37
R0_21	88746.249447	25	68.05
R0_22	104744.389515	32	10.43
R0_23	99718.790531	30	13.33
R0_24	103892.313766	26	13.09
R0_25	108534.767303	31	8.7
R0_26	87075.419369	27	17
R0_27	100426.962662	28	112.18
R0_28	90377.904110	28	32.95
R0_29	98024.053631	30	6.34
R0_30	95118.335552	25	43.52
R0_31	96675.591145	27	18.6
R0_32	91262.649307	25	22.34
R0_33	93629.305916	27	4386.51

Table 3: Objective value and runtimes for all R25_* instances.

Name	Obj. Val.	No. of Trucks	Runtime (seconds)
R25_1	118396.8828	29	11.28
R25_2	98740.53165	26	3230.15
R25_3	106170.6854	27	13.12
R25_4	100450.5393	26	31.92
R25_5	108207.6914	27	16.48
R25_6	105661.9618	32	8.83
R25_7	90522.67456	25	14.33
R25_8	94436.90666	27	108.72
R25_9	90469.11232	24	199.65
R25_10	89487.59182	26	830.57
R25_11	105605.3905	29	10.19
R25_12	92623.36115	27	18.89
R25_13	94298.38812	27	18.04
R25_14	102997.5337	27	59.96
R25_15	90663.24898	26	98.24
R25_16	93747.07615	27	15.87
R25_17	103151.0472	30	12.26
R25_18	111525.3425	27	12.25
R25_19	98866.27653	29	13.12
R25_20	92312.30598	26	233.32
R25_21	88746.24945	25	27.67
R25_22	104744.3895	32	11.5
R25_23	101208.0756	30	11.4
R25_24	103892.3139	26	16.88
R25_25	108534.7673	31	10.64
R25_26	89065.01873	28	17.76
R25_27	101790.5912	27	313.42
R25_28	90377.90411	28	16.74
R25_29	98024.05363	30	7.11
R25_30	95118.33555	25	44.07
R25_31	96675.59115	27	18.24
R25_32	91395.84915	25	20.72
R25_33	93629.30592	27	1625.11

Table 4: Objective value and runtimes for all R50_* instances.

Name	Obj. Val.	No. of Trucks	Runtime (seconds)
R50_1	118789.7969	29	15.06
R50_2	98920.27491	26	24
R50_3	106170.6854	27	37.51
R50_4	100633.3684	26	35.23
R50_5	108207.6914	27	15.15
R50_6	105661.9618	32	10.1
R50_7	90522.67456	25	16.3
R50_8	94436.90666	27	80.61
R50_9	90469.11232	24	151.3
R50_10	89487.59182	26	492.31
R50_11	105605.3905	29	11.86
R50_12	92548.78973	27	15.47
R50_13	95201.90243	28	16.95
R50_14	102583.3625	27	102.21
R50_15	89942.56376	25	157.53
R50_16	93747.07615	27	13.76
R50_17	106214.3042	30	8.81
R50_18	112361.5689	28	15.73
R50_19	102391.3621	29	34.19
R50_20	92312.30598	26	259.72
R50_21	88929.07854	25	22.89
R50_22	104744.3895	32	9.78
R50_23	101208.0756	30	12.69
R50_24	103892.314	26	14.21
R50_25	108534.7673	31	10.17
R50_26	89065.01873	28	17.22
R50_27	103097.3055	28	391.99
R50_28	93441.16121	28	18.74
R50_29	100416.853	31	6.64
R50_30	96204.42049	25	35.56
R50_31	96675.59115	27	20.2
R50_32	91262.64931	25	22.35
R50_33	93629.30592	27	631.72

Table 5: Objective value and runtimes for all R75_* instances.

Name	Obj. Val.	No. of Trucks	Runtime (seconds)
R75_1	119646.5105	30	14.32
R75_2	98994.84615	26	48.74
R75_3	106170.6854	27	12.31
R75_4	100633.3684	26	45.52
R75_5	108638.4058	28	15.88
R75_6	105661.9618	32	10.31
R75_7	90522.67456	25	17.66
R75_8	96420.50617	27	37.98
R75_9	90469.11232	24	67.84
R75_10	89487.59182	26	27.09
R75_11	105605.3905	29	8.95
R75_12	93266.56111	27	12.36
R75_13	95201.90243	28	18.28
R75_14	103815.5054	28	19.48
R75_15	91260.24931	26	711.33
R75_16	93996.76128	27	12.35
R75_17	106861.5329	30	10.41
R75_18	112361.5689	28	12.53
R75_19	102391.3622	29	12.33
R75_20	93868.70567	27	449.09
R75_21	88929.07854	25	17.89
R75_22	104744.3895	32	9.22
R75_23	101344.2747	31	9.63
R75_24	108655.1994	27	17.24
R75_25	108534.7673	31	10.63
R75_26	89065.01873	28	16.3
R75_27	103097.3055	28	85.47
R75_28	96504.4182	28	410.77
R75_29	100416.853	31	6.53
R75_30	96425.04988	26	33.7
R75_31	96675.59115	27	16.3
R75_32	92055.33485	26	24.38
R75_33	93629.30592	27	56.41

Table 6: Objective value and runtimes for all R100_* instances.

Name	Obj. Val.	No. of Trucks	Runtime(seconds)
R100_1	119646.5105	30	12.71
R100_2	98994.84615	26	74.46
R100_3	106170.6854	27	11.67
R100_4	101276.5684	26	309.12
R100_5	108638.4058	28	17.18
R100_6	105661.9618	32	8.84
R100_7	91436.47476	26	28825.65
R100_8	95954.90604	28	25.52
R100_9	90469.11232	24	116.98
R100_10	89487.59182	26	38.65
R100_11	105605.3905	29	9.39
R100_12	93963.7607	27	13.17
R100_13	95201.90243	28	16.97
R100_14	104229.6766	28	21.03
R100_15	91260.24931	26	373.81
R100_16	96570.58994	29	5.37
R100_17	107947.618	30	8.48
R100_18	112361.5689	28	13.2
R100_19	102391.3622	29	9.02
R100_20	93868.70567	27	177.23
R100_21	89219.04945	25	31.95
R100_22	104744.3895	32	10.7
R100_23	103333.8741	32	9.94
R100_24	108655.1994	27	17.97
R100_25	108534.7673	31	10.01
R100_26	89065.01873	28	15.7
R100_27	103097.3055	28	181.9
R100_28	96504.4182	28	129.5
R100_29	104786.8243	32	5.82
R100_30	97511.13493	26	13.64
R100_31	96675.59115	27	19.17
R100_32	92055.33485	26	19.84
R100_33	93629.30592	27	25.56

6 Instances as Benchmarks for an On-Line PDPTW

The instances presented in this report and contained in the MIPLIB are, in a sense, relatively easy. This relative ease stems from two sources. The first is the efficient MIP formulation available for this problem type (i.e. drayage) and the second is the originally intended use of these instances — an on-line optimization based on real-world data (see Máhr et al. (2010)). For example, each instance in this set represents one day's worth of 65 truckload jobs that require transport within their time windows by a fleet of 40 trucks; because these specifications came from operational conditions in the real-world, every instance (day) was known to be feasible. This section elaborates on the use of Yang et al.'s MIP formulation as well as on the use of these instances and others for the evaluation of on-line algorithms.

In the original use of these instances, each day was partitioned into 30-second intervals for use in a rolling horizon, on-line optimization framework. In this way, jobs were included in the MIP for planning only as their arrival time dictated. Given the promising results regarding the use of MIP formulations in the literature, we chose such a formulation for our on-line optimization. Specifically, Yang et al. (2004) demonstrate the superiority of an exact mixed integer programming formulation of the truckload pick-up and delivery problem with time-windows, solved in a rolling horizon framework at each decision instance. They compare their re-optimization approaches to three heuristic approaches (a simple round robin assignment, an insertion heuristic, and a reordering approach). This comparison reveals that the re-optimization approaches systematically outperform the heuristic approaches by about 10%. This superior re-optimization approach, has its origins in the paper by Yang et al. (1999).

In order for on-line re-optimization to be competitive, however, the MIP must be able to run to optimality or near-optimality within each decision epoch of the rolling horizon. Yang et al. (1999) achieve this by exploiting an assignment problem structure for the routing decisions in this problem; the time-related decisions are then included only in the constraints to ensure the feasibility of the arrival time at each job (see 4). As assignment problems are comparatively easy to solve, most MIP solvers can readily generate an optimal solution in a short amount of time. Thus, for these instances, it is primarily the continuous (time-window related) constraints that influence the runtime. The instances included in the MIPLIB demonstrate that utilizing this MIP structure, outside of a rolling horizon framework with a full day's worth of data, is still efficient with an average runtime of approximately 338seconds and a standard deviation of 2302 seconds.

Jobs with variable arrival times are just one out of many possible causes for the uncertainty that occurs in the real-world. Therefore, we consider the study of on-line algorithms to be very important. In the paper by Máhr et al. (2010), mentioned previously, two other causes of uncertainty were studied in the same drayage setting: variable service times and random truck break-downs. These two alternative benchmark sets were not submitted to the MIPLIB, because the focus of the MIPLIB is on static instances, and as static instances, these are not very interesting. In particular, in contrast to the instances with variable arrival times, most of the variable service time and truck breakdown instances are infeasible in their *a priori* form. Specifically, these instances, when solved with the current MIP formulation, exhibit short computation times and solutions many job rejections.

In an on-line setting, however, these instances make more sense. Plans based on the expected service times and no truck break-downs are usually feasible, but may lead to significant time window violations when service times turn out to be much longer. These benchmark instances, in turn, serve as the basis for comparison of an on-line MIP approach and multi-agent heuristics. In general, with varying job-arrival times the on-line MIP performs better, but with other causes of uncertainty, the multi-agent heuristics are competitive, or sometimes even outperform the on-line optimization approach discussed above. For more details on these results, or on the generation of the other on-line problem instances, please refer to this paper Máhr et al. (2010) or contact one of the authors.

References

Máhr, Tamás, Jordan Srour, Mathijs de Weerdt, Rob Zuidwijk. 2010. Can agents measure up? a comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty. *Transportation Research Part C: Emerging Technologies* **18**(1) 99 – 119. URL `http://www.sciencedirect.com/science/article/B6VGJ-4WHDHPM-1/2/1fdf7dba4276dc23acfaf245efc94642`.

Yang, J., P. Jaillet, H. Mahmassani. 1999. On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record* **1667** 107–113.

Yang, J., P. Jaillet, H. Mahmassani. 2004. Real-time multi-vehicle truckload pick-up and delivery problems. *Transportation Science* **38**(2) 135–148.

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2010

Linearization and Decomposition Methods for Large Scale Stochastic Inventory Routing Problem with Service Level Constraints
Yugang Yu, Chengbin Chu, Haoxun Chen, and Feng Chu
ERS-2010-008-LIS
http://hdl.handle.net/1765/18041

Sustainable Passenger Transportation: Dynamic Ride-Sharing
Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang
ERS-2010-010-LIS
http://hdl.handle.net/1765/18429

Visualization of Ship Risk Profiles for the Shipping Industry
Sabine Knapp and Michel van de Velden
ERS-2010-013-LIS
http://hdl.handle.net/1765/19197

Intelligent Personalized Trading Agents that facilitate Real-time Decisionmaking for Auctioneers and Buyers in the Dutch Flower Auctions
Wolfgang Ketter, Eric van Heck, and Rob Zuidwijk
ERS-2010-016-LIS
http://hdl.handle.net/1765/19367

Necessary Condition Hypotheses in Operations Management
Jan Dul, Tony Hak, Gary Goertz, and Chris Voss
ERS-2010-019-LIS
http://hdl.handle.net/1765/19666

Human Factors: Spanning the Gap between OM & HRM
W. Patrick Neumann, and Jan Dul
ERS-2010-020-LIS
http://hdl.handle.net/1765/19668

AUK: a simple alternative to the AUC
Uzay Kaymak, Arie Ben-David, and Rob Potharst
ERS-2010-024-LIS
http://hdl.handle.net/1765/19678

The Value of Optimization in Dynamic Ride-Sharing: a Simulation Study in Metro Atlanta
Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang
ERS-2010-034-LIS
http://hdl.handle.net/1765/20456

MIPLIB Truckload PDPTW Instances Derived from a Real-World Drayage Case
F. Jordan Srour, Tamas Mahr, Mathijs de Weerdt, and Rob Zuidwijk
ERS-2010-036-LIS
http://hdl.handle.net/1765/20883

---

* A complete overview of the ERIM Report Series Research in Management:
https://ep.eur.nl/handle/1765/1

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship