The Truck Scheduling Problem at Crossdocking Terminals
Exclusive versus Mixed Mode

L. Berghman[1], C. Briand[2,3], R. Leus[4] and P. Lopez[2]
[1]*Université de Toulouse - Toulouse Business School, 20 BD Lascrosses BP 7010, 31068 Toulouse Cedex 7, France*
[2]*CNRS, LAAS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France*
[3]*Université de Toulouse, UPS, LAAS, F-31400 Toulouse, France*
[4]*ORSTAT, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium*

Keywords: Crossdocking, Truck Scheduling, Parallel Machine Scheduling, Integer Programming.

Abstract: In this paper we study the scheduling of the docking operations of trucks at a warehouse; each truck is either empty and needs to be loaded, or full and has to be unloaded (but not both). We focus on crossdocking, which is a recent warehouse concept that favors the transfers of as many incoming products as possible directly to outgoing trailers, without intermediate storage in the warehouse. We propose a time-indexed integer programming formulation for scheduling the loading and unloading of the trucks at the docks, and we distinguish between a so-called "mixed mode", in which some or all of the docks can be used both for loading as well as unloading, and an "exclusive mode", in which each dock is dedicated to only one of the two types of operations. Computational experiments are provided to compare the efficiency of the two modes.

1 INTRODUCTION

Crossdocking is a warehouse management concept in which items delivered to a warehouse by inbound trucks are immediately sorted out, reorganized based on customer demands and loaded into outbound trucks for delivery to customers, without requiring excessive inventory at the warehouse ((van Belle et al., 2012)). If any item is held in storage, it is usually for a brief period of time that is generally less than 24 hours. Advantages of crossdocking can accrue from faster deliveries, lower inventory costs, and a reduction of the warehouse space requirement ((Apte and Viswanathan, 2000; Boysen, 2010)). Compared to traditional warehousing, the storage as well as the length of the stay of a product in the warehouse is limited, which requires an appropriate coordination of inbound and outbound trucks ((Boysen et al., 2010; Yu and Egbelu, 2008)).

The truck scheduling problem, which decides on the succession of truck processing at the dock doors, is especially important to ensure a rapid turnover and on-time deliveries. The problem studied concerns the operational level: trucks are allocated to the different docks so as to minimize the storage usage during the product transfer. The internal organization of the warehouse (scanning, sorting, transporting) is not explicitly taken into consideration. We also do not model the resources that may be needed to load or unload the trucks, which implies the assumption that these resources are available in sufficient quantities to ensure the correct execution of an arbitrary docking schedule.

There exist two different service modes, which depend on the degree of freedom in assigning inbound and outbound trucks to docks. In the *exclusive mode*, each dock is exclusively dedicated either to inbound or to outbound operations. It is a widespread guideline in real-world terminals to ease product flows and supervision ((Boysen and Fliedner, 2010)). Typically, one side of the terminal is dedicated to inbound and the other to outbound operations. In the *mixed mode*, on the other hand, an intermixed sequence of inbound and outbound trucks to be processed per dock is allowed. As stated in (Carlo and Bozer, 2011), in a typical crossdock application, once a dock is classified as an inbound (or an outbound) dock, it remains that way until the docks are reclassified. Their experiments show, however, that grouping the inbound docks together and the outbound docks together is generally not a good configuration to use when the decision maker wants to minimize the travel distance of the forklifts, which follow a rectilinear travel path between the doors inside the warehouse. Remark that

Berghman L., Briand C., Leus R. and Lopez P..
The Truck Scheduling Problem at Crossdocking Terminals - Exclusive versus Mixed Mode.
DOI: 10.5220/0005205102470253
In *Proceedings of the International Conference on Operations Research and Enterprise Systems* (ICORES-2015), pages 247-253
ISBN: 978-989-758-075-8

the exclusive mode resembles a hybrid flow shop: the inbound and outbound docks are the first and the second stage, respectively. The mixed mode, on the other hand, resembles a parallel machine scheduling problem with precedence constraints: both the inbound and the outbound trucks are scheduled on the same set of identical machines.

The purpose of this paper is to propose a mathematical formulation of the truck scheduling problem at crossdocking terminals operating in mixed mode and to show its interest by comparison with the exclusive mode. It is structured as follows. Section 2 gives a brief overview of the relevant literature on the problem under study. A detailed problem description can be found in Section 3. A time-indexed formulation is presented in Section 4. The results of the computational experiments can be found in Section 5. Finally, some conclusions round off the paper in Section 6.

2 LITERATURE REVIEW

The concept of crossdocking has received a lot of attention in recent literature: cases with one receiving and one shipping door are most frequently studied. A comprehensive overview of different variations and the available literature can be found in (Boysen and Fliedner, 2010), (van Belle et al., 2012) and (Maknoon, 2013).

(Alpan et al., 2011a) and (Alpan et al., 2011b), on the one hand, consider a multiple-door crossdock environment with exclusive mode of service, where preemption of loading operations is allowed. Each outbound truck serves a single destination; each inbound truck can contain products for several destinations. There are several ways to treat the product flows: (i) products can be transshipped directly from an inbound to an outbound truck if one is available; (ii) they can be temporarily stored to be loaded later on; or (iii) an outbound truck can be replaced to facilitate direct loading. Contrary to our model, the objective here is not time-related; the sum of the inventory holding cost (per unit product) and the truck replacement cost is minimized. As the sequence of the inbound trucks is known, the problem consists in scheduling the outbound trucks. (Alpan et al., 2011b) try to find optimal or near-optimal scheduling policies using dynamic programming while (Alpan et al., 2011a) present several heuristics based on constraining the solution space that is generated by the dynamic programming model of (Alpan et al., 2011b). Numerical experiments show that the heuristics can find near-optimal solutions much faster.

(Miao et al., 2009), on the other hand, consider the truck-dock assignment problem with operational time constraint for the mixed mode. For each pair of inbound and outbound trucks, the number of pallets that has to be transferred from the inbound to the outbound truck is defined. Once more, the objective is not time-related; the authors aim to minimize the number of unfulfilled shipments and the total shipment costs at the same time. The problem is formulated as an integer programming model and two metaheuristics are proposed: tabu search and a genetic algorithm. It turns out that for medium-size and large-size instances, the metaheuristic approaches are preferred in order to get quick and good solutions.

Some articles in literature model the truck scheduling problem at crossdocking terminals with exclusive mode as a machine scheduling problem. In (Chen and Lee, 2009) and (Chen and Song, 2009), the crossdocking environment is treated as a two-stage flow shop, but only instances with a small number of docks are considered. The number and types of products to be loaded (unloaded) per outbound (inbound) truck are not defined a priori, but each job in the second stage (outbound; loading) can be processed only after the processing of some jobs in the first stage (inbound; unloading). (Chen and Lee, 2009) show that the problem is NP-hard for the two-machine case where the objective is to minimize the makespan. Furthermore, they present a polynomial approximation algorithm and a branch-and-bound algorithm. (Chen and Song, 2009) consider the hybrid case where at least one stage contains more than one machine; they present a mixed integer programming model for small-scale instances and different heuristics for moderate and large-scale instances.

(Li et al., 2004) use JIT scheduling to solve the problem of scheduling loading and unloading activities when the goal is to complete processing each container exactly at its due date. Each incoming container has a release time and a due date and each outgoing container has a due date. The crossdock can be divided into an import area and an export area. Products have known destinations before they enter the crossdock, such that precedence relationships arise. They present an integer programming model, as well as two heuristics for this NP-hard problem. The first uses squeaky wheel optimization ((Joslin and Clements, 1999)) embedded in a genetic algorithm and the second uses linear programming within a genetic algorithm. (Álvarez Pérez et al., 2009) consider the same problem and present a solution approach based on a combination of two metaheuristics, reactive GRASP and tabu search. They conclude that their algorithm is an excellent alternative to the approach of (Li et al., 2004). Note that JIT scheduling as considered in the two above papers amounts to minimizing the difference in completion time between each pair of tasks that is involved in a precedence constraint. The details of the problems studied in the latter references are such that the proposed algorithms are not suitable for our setting.

3 DETAILED PROBLEM STATEMENT

We examine a crossdocking warehouse where incoming trucks $i \in I$ need to be unloaded and outgoing trucks $o \in O$ need to be loaded (where I is the set containing all inbound trucks while O is the set containing all outbound trucks). The warehouse features n docks that can be used both for loading and unloading (mixed mode). The processing time of truck $j \in I \cup O$ equals p_j. This processing time includes the loading or unloading but also the transportation of goods inside the crossdock and other handling operations between dock doors. It is assumed that there is sufficient workforce to load/unload all docked trucks at the same time. Hence, a truck assigned to a dock does not wait for the availability of a material handler.

The products on the trucks are packed on unit-size pallets, which move collectively as a unit: re-packing inside the terminal is to be avoided. Each pallet on an inbound truck i needs to be loaded on an outbound truck o, which gives rise to a precedence constraint $(i,o) \in P \subset I \times O$, with P the set containing all couples of inbound trucks i and outbound trucks o that share a precedence constraint. Each truck j has a release time r_j (planned arrival time) and a deadline $\tilde{d}_j$ (its latest departure time).

Products can be transshipped directly from an inbound to an outbound truck if the outbound truck is placed at a dock. Otherwise, the products are temporarily stored and will be loaded later on. Each couple $(i,o) \in P$ has a weight w_{io}, representing the number of pallets that go from inbound truck i to outbound truck o. The objective is to minimize the weighted sum of sojourn times of the pallets stocked in the warehouse. According to (Boysen and Fliedner, 2010), this is a valuable objective because the crossdocking concept relies on a rapid turnover of shipments. It also reduces the danger of late shipments: the number of products in the storage area can only be decreased by loading them on outbound trucks to leave the terminal as early as possible. Moreover, a lower stock size also reduces the material handling effort inside the terminal. Remark that the time spent by a pallet in the storage area is equal to the flow time of the pallet: the difference between the start of loading the outbound trailer and the start of unloading the inbound trailer.

Our problem can be modeled as a parallel machine scheduling problem with release dates, deadlines, and precedence constraints, denoted by $Pm|r_i, \tilde{d}_i, prec|-$. As this problem is a generalization of the $1|r_j, \tilde{d}_j|-$ problem which is NP-complete ((Lenstra et al., 1977)), even finding a feasible solution for the problem is NP-complete.

For all trucks $j \in I \cup O$, let s_j be the starting time of the handling of truck j. A conceptual problem statement with these variables is the following:

$$\min \quad z = \sum_{(i,o)\in P} w_{io}(s_o - s_i) \tag{1}$$

subject to

$$s_j \geq r_j \qquad \forall j \in I \cup O \tag{2}$$
$$s_j + p_j \leq \tilde{d}_j \qquad \forall j \in I \cup O \tag{3}$$
$$s_o - s_i \geq 0 \qquad \forall (i,o) \in P \tag{4}$$
$$|A_\tau| \leq n \qquad \forall \tau \in \mathcal{T} \tag{5}$$

with $A_\tau = \{j \in I \cup O \,|\, s_j < \tau \leq s_j + p_j\}$ the set containing all tasks being executed during time period τ and $\mathcal{T}$ the set containing all time periods considered (time horizon). The objective function (1) minimizes the total weighted usage of the storage area. Constraints (2) and (3) impose the time windows for all trucks. Constraints (4) ensure that, if there exists a precedence constraint between inbound truck i and outbound truck o, then o cannot be processed before i. Finally, constraints (5) enforce the capacity of the docks.

Remark that if we replace constraint (5) by two capacity constraints (one for the inbound docks with right-hand side n_i equal to the number of inbound docks and one for the outbound docks with right-hand side n_o equal to the number of outbound docks), we obtain a formulation for the exclusive mode. This can be easily done for the time-indexed formulation presented in the next section as well.

4 TIME-INDEXED FORMULATION

A time-indexed formulation discretizes the continuous time space into periods $\tau \in \mathcal{T}$ of a fixed length. Let period τ be the interval $[t-1, t[$. It is well known that time-indexed formulations perform well for scheduling problems because the linear programming relaxations provide strong lower bounds ((Dyer and Wolsey, 1990)). For this reason, we will test the integer programming formulation below, which is called F1 in the sequel.

For all inbound trucks $i \in I$ and for all time periods $\tau \in \mathcal{T}_i$, we have

$$x_{i\tau} = \begin{cases} 1 & \text{if the unloading of inbound truck } i \text{ is} \\ & \text{started during time period } \tau, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathcal{T}_i = \{r_i+1, r_i+2, \ldots, \tilde{d}_i - p_i + 1\}$, the relevant time window for inbound truck i. Additionally, for all outbound trucks $o \in O$ and for all time periods $\tau \in \mathcal{T}_o$, we have

$$y_{o\tau} = \begin{cases} 1 & \text{if the loading of outbound truck } o \text{ is} \\ & \text{started during time period } \tau, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathcal{T}_o = \{r_o+1, r_o+2, \ldots, \tilde{d}_o - p_o + 1\}$, the relevant time window for outbound truck o.

A time-indexed formulation for the considered truck scheduling problem is the following:

$$\min \quad z = \sum_{(i,o)\in P} \sum_{\tau\in\mathcal{T}} w_{io}\tau(y_{o\tau} - x_{i\tau}) \tag{6}$$

subject to

$$\sum_{\tau\in\mathcal{T}_i} x_{i\tau} = 1 \qquad \forall i \in I \tag{7}$$

$$\sum_{\tau\in\mathcal{T}_o} y_{o\tau} = 1 \qquad \forall o \in O \tag{8}$$

$$\sum_{\tau\in\mathcal{T}} \tau(x_{i\tau} - y_{o\tau}) \le 0 \qquad \forall (i,o) \in P \tag{9}$$

$$\sum_{i\in I} \sum_{u=\tau-p_i+1}^{\tau} x_{iu} + \sum_{o\in O} \sum_{u=\tau-p_o+1}^{\tau} y_{ou} \le n \qquad \forall \tau \in \mathcal{T} \tag{10}$$

$$x_{i\tau} \in \{0,1\} \qquad \forall i \in I; \forall \tau \in \mathcal{T}_i \tag{11}$$

$$y_{o\tau} \in \{0,1\} \qquad \forall o \in O; \forall \tau \in \mathcal{T}_o \tag{12}$$

The objective function (6) minimizes the total weighted usage of the storage area. Constraints (7) and (8) demand each truck to be assigned to exactly one gate. Constraints (9) ensure that if there exists a precedence constraint between inbound truck i and outbound truck o, then o cannot be processed before i. Constraints (10) enforce the capacity of the docks.

An alternative precedence constraint is the following:

$$\sum_{u=1}^{\tau} x_{iu} - y_{o\tau} \ge 0 \qquad \forall (i,o) \in P; \forall \tau \in \mathcal{T} \tag{13}$$

Informally, this constraint states that in fractional solutions, the loading task can only be started up to the fraction to which the unloading task has been started. (Christofides et al., 1987) call this constraint *disaggregated*. The formulation obtained by replacing constraint (9) in formulation F1 by (13) will be referred to as F2. When we take a look at the polyhedron that contains all feasible solutions for the LP-relaxations, F2 is theoretically stronger since each feasible solution for the LP relaxation of formulation F2 is also a feasible solution to the LP relaxation of formulation F1. (Laborie and Nuijten, 2008) observe, however, that the additional CPU time needed to solve the larger linear program can counterbalance the significant improvement of the bound. Both formulations will be tested empirically.

Although in mixed mode all gates can serve both to unload incoming trailers and to load outgoing trailers, it might not be needed that every gate has this *double purpose*. It is possible that in an optimal schedule, at some gates only incoming trailers are unloaded and at other gates only outgoing trailers are loaded. Indeed, since switching completely to mixed mode might impact significantly the company organization, both because of the placing of the docks and because of the internal transportation within the warehouse, it is worth determining the gain obtained when switching only a small number of docks from exclusive to mixed mode. We remark that when a warehouse is expanding, the additional docks can be mixed docks, or changing only a limited number of docks does not necessarily change the internal organization of the warehouse in a drastic way. To determine the minimal number of gates that has to be double purpose so that the optimal objective value is kept, we can work as follows. We refer to n_i (respectively n_o) as the number of gates that are used for unloading (loading) purposes. Moreover, let δ_i (δ_o) be the number of gates that we allow to unload (load) incoming (outgoing) trailers, on top of n_i (n_o), during certain time periods and define δ_i^*, δ_o^* as the optimal values for these variables. A schematic representation is given in Figure 1. In a first stage, we solve the formulation F1, which gives the optimal objective value z^*. In a second stage, we solve a second time-indexed formulation: we minimize $\delta = \delta_i + \delta_o$ subject to (7)–(12) and we add the following constraints:

$$\sum_{(i,o)\in P} \sum_{\tau\in\mathcal{T}} w_{io}\tau(y_{o\tau} - x_{i\tau}) \le z^* \tag{14}$$

$$\sum_{i\in I} \sum_{u=\tau-p_i+1}^{\tau} x_{iu} \le n_i + \delta_i \qquad \forall \tau \in \mathcal{T} \tag{15}$$

$$\sum_{o\in O} \sum_{u=\tau-p_o+1}^{\tau} y_{ou} \le n_o + \delta_o \qquad \forall \tau \in \mathcal{T} \tag{16}$$

$$n_i + n_o = n \tag{17}$$

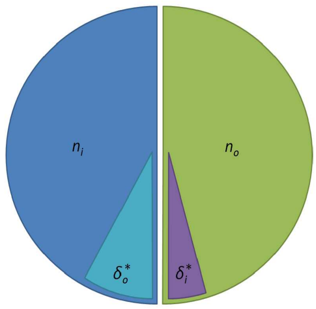

Figure 1: Schematic representation.

The values of δ_i and δ_o are calculated in constraints (15) and (16), respectively. Constraints 10 ensuring in any case that the sum of used gates never exceeds the total gate capacity. By minimizing their sum, we minimize the number of gates that are used both for loading as well as for unloading.

5 COMPUTATIONAL RESULTS

To the best of our knowledge, the problem studied in this paper was never studied as such. Thus, we create new instances in line with (Chen and Song, 2009) and (Li et al., 2004) in the following way.

Remark that we rounded all fractional values to obtain integer data. The number of gates is $n \in \{10,20,30\}$. The number of inbound trucks is $|I| \in \{3n,4n,5n\}$ and the number of outbound trucks is $|O| \in \{0.8|I|,|I|,1.2|I|\}$. Since the time needed to unload one pallet equals one time unit, the time needed to unload a trailer i, equals the number of pallets to be unloaded. The processing time p_i is uniformly distributed in $[a,30]$ with $a \in \{10,20,30\}$. For each inbound truck i, the number of outbound trucks in which goods of truck i will be loaded is $nb_i \in \{1,\ldots,\frac{p_i}{\gamma}\}$. The number of precedence constraints is determined by γ: the larger γ, the more precedence constraints. These outbound trucks are chosen randomly. The number of pallets that will be charged from this inbound truck i to one of the corresponding outbound trucks o is $nb_{io} \in \{0.8\frac{p_i}{nb_i},\frac{p_i}{nb_i},1.2\frac{p_i}{nb_i}\}$. There exists a precedence constraint $(i,o) \in P$ between an inbound trailer i and an outbound trailer o when at least one pallet is associated with both inbound trailer i and outbound trailer o. The weight w_{io} of a precedence constraint $(i,o) \in P$ is equal to the number of pallets that need to be unloaded from inbound trailer i and loaded to outbound trailer o afterwards. The time needed to load an outbound trailer o is equal to the number of pallets that will be loaded in this trailer. The ready times r_i for the inbound trailers are uniformly distributed in $[0,\alpha\frac{\sum p_i}{n}]$ with $\alpha \in \{0.3,0.6,0.9\}$. The deadlines for the outbound trailers are in $\tilde{d}_o \in [\phi\Omega_o,\beta\Omega_o]$ with $\Omega_o = \max_{(i,o)\in P}\{r_i+p_o\}$. The length of the time horizon is $|T| = \max_{o\in O}\{\tilde{d}_o\}$. The ready times r_o for the outbound trailers are uniformly distributed in $[r_o^{\max},\tilde{d}_o - p_o]$ with $r_o^{\max} = \max_{(i,o)\in P}\{r_i\}$. The deadlines for the inbound trailers $\tilde{d}_i$ are uniformly distributed in $[1.5(r_i+p_i),\tilde{d}_i^{\max}]$ with $\tilde{d}_i^{\max} = \min\{\min_{(i,o)\in P}\{\tilde{d}_o - p_o\}+p_i,\max_{(i,o)\in P}\{\tilde{d}_o\}\}$.

Remark that γ, ϕ and β are parameters. The tightness of the time windows is determined by ϕ and β: the smaller the difference between β and ϕ, the tighter the time windows. We will assign different values to both γ and β to obtain different datasets.

All models are encoded in C using the Microsoft Visual Studio programming environment, and executed on a PC computer with an Intel Core i3-2350M CPU 2.30-GHz processor and 2 GB RAM, equipped with Windows 7. ILOG CPLEX 12.4 is used to solve the models.

As preliminary experiments, we created 567 instances : we fixed the parameters $\gamma = 3$, $\phi = 1.5$ and $\beta = 5$ and we created three instances for each combination of the other parameters. An instance is named after its parameters : "n_$|I|$_$|O|$_a_α_index". We have implemented both constraints (9) and (13) as precedence constraints and we have remarked that the formulation performs better with constraint (9). The following results refer to this formulation. Detailed results can be found in Table 1: for each set of instances, the average number of instances that were proved to be infeasible, the average number of instances for which a feasible solution was found that was not proven to be optimal and the average number of instances for which an optimal solution was found are mentioned both for exclusive and for mixed mode. Overall, for the exclusive mode, 22% of the instances were proved to be infeasible, while none of them were proved infeasible for the mixed mode. This is a first evidence that companies can expect more flexibility when switching from exclusive to mixed mode. For the exclusive mode, a feasible solution was found within a time limit of five minutes for 69% of the instances; for 2% of all instances, this solution was proven to be an optimal solution. For the mixed mode, a feasible solution was found for 93% of the instances, and for 3% of all instances this solution was optimal. When we only look at the instances for which an optimal solution was found, the average computation time to find these optimal solutions is 70 seconds for the exclusive mode and 23 seconds for the mixed mode. To give an idea about the quality of the solutions founds, we mention that the av-

Table 1: Computational results.

		exclusive mode			mixed mode		
n	$\|I\|$	infeasible	feasible	optimal	infeasible	feasible	optimal
10	30	23.81%	65.08%	11.11%	0.00%	84.13%	15.87%
10	40	23.81%	74.60%	1.59%	0.00%	96.83%	3.17%
10	50	26.98%	63.49%	0.00%	0.00%	98.41%	0.00%
20	60	12.70%	82.54%	3.17%	0.00%	96.83%	3.17%
20	80	22.22%	73.02%	0.00%	0.00%	98.41%	0.00%
20	100	23.81%	61.90%	0.00%	0.00%	90.48%	0.00%
30	90	15.87%	77.78%	0.00%	0.00%	95.24%	4.76%
30	120	19.05%	65.08%	0.00%	0.00%	95.24%	0.00%
30	150	28.57%	53.97%	0.00%	0.00%	79.69%	0.00%
total		21.87%	68.61%	1.76%	0.00%	92.95%	3.00%

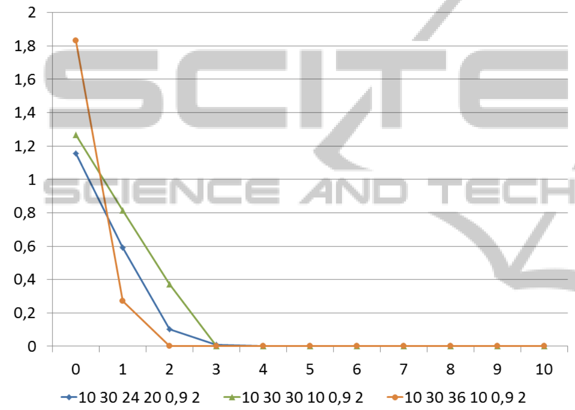

Figure 2.

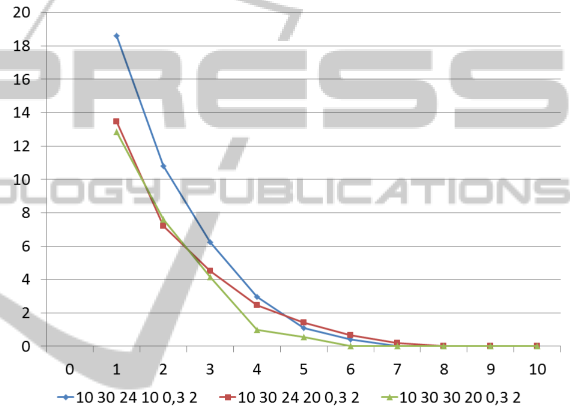

Figure 3.

erage GAP between the best solution found within 5 minutes and the optimal solution of the linear relaxation of the formulation is 13,28%. The average GAP with respect to a Lagrangian relaxation that we implemented is 6,73%. When we compare the instances for which a feasible solution was found both for the exclusive and mixed modes, we calculated an improvement of 8% of the objective value with a mixed mode. This is a second evidence that companies might take profits from switching to a mixed organization.

We minimized the number of double purpose gates δ for the 17 instances for which we found an optimal solution for the mixed mode. For 6 of these instances, we found a feasible solution that was not guaranteed to be optimal; the average number of double purpose gates is 49%. For the other instances, we found an optimal solution. The average number of double purpose gates is 27% and the average computation time is 50 seconds.

In Figures 2 and 3, we illustrate the results obtained with three instances having 10 docks. On the horizontal axis, we display the number of docks $\delta = \delta_i + \delta_o$ (out of 10) switched in mixed mode. On the vertical axis, we indicate the GAP $= 100 * (z - z^*)/z^*$ between the obtained solution z and the optimal solution z^* when all docks are in mixed mode. In Figure 2, we can see that for instances with a rather small difference between exclusive and mixed mode (some 2%), only changing one, two, or three docks (out of 10) to mixed mode is enough to obtain the optimal solution obtained when all docks are in mixed mode. Figure 3 shows instances for which the exclusive mode is infeasible. Having only one dock in mixed mode allows finding a feasible solution, and better solutions are obtained when the number of mixed gates increases.

6 CONCLUSIONS

We have presented a time-indexed (integer programming) formulation for the truck scheduling problem at crossdocking terminals. We have experimentally compared the mixed mode strategy with the exclusive one. As might be expected, the results confirm that it is easier to find a feasible solution, or even an optimal one, when handling terminals operating with a mixed mode. Moreover, our experiments provide insight into the number of gates to be changed from exclusive to mixed in order to guarantee the best performance.

For future research, it may be interesting to investigate the special case of the problem with $p_i = p$. The complexity of $Pm|r_i, \tilde{d_i}, p_i = p|\sum w_i C_i$ is open ((Kravchenko and Werner, 2011)) and this problem is a special case of our problem with $p_i = p$: take $|I| = 1$ with $i \in I$, define $\tilde{d_i} = r_i + p_i$ such that $s_i = r_i$ and define for all $o \in O$, $r'_o = \max\{r_o; r_i + p_i\}$ and $w_o = w_{io}$.

Another interesting problem is an extension in which trailers are allowed to remain at the gate longer than strictly needed for loading or unloading. In this way, the number of direct transfers from inbound to outbound trailers can be augmented and consequently, the usage of the storage area can be decreased.

ACKNOWLEDGEMENTS

This research was part of ROCKS project supported by CNRS/INS2I (Centre National de la Recherche Scientifique/Institut des Sciences de l'Information et de leurs Interactions), PICS action n 6421.

REFERENCES

Alpan, G., Ladier, A.-L., Larbi, R., and Penz, B. (2011a). Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. *Computers & Industrial Engineering*, 61(2):402–408.

Alpan, G., Larbi, R., and Penz, B. (2011b). A bounded dynamic programming approach to schedule operations in a cross docking platform. *Computers & Industrial Engineering*, 60(3):385–396.

Álvarez Pérez, G., González-Velarde, J., and Fowler, J. (2009). Crossdocking – Just in time scheduling: an alternative solution approach. *Journal of the Operational Research Society*, 60:554–564.

Apte, U. and Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics: Research and Applications*, 3(3):291–302.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37:32–41.

Boysen, N. and Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38:413–422.

Boysen, N., Fliedner, M., and Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*, 32:135–161.

Carlo, H. and Bozer, Y. (2011). Analysis of optimum shape and door assignment problems in rectangular unit-load crossdocks. *International Journal of Logistics: Research and Applications*, 14(3):149–163.

Chen, F. and Lee, C.-Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193:59–72.

Chen, F. and Song, K. (2009). Minimizing makespan on two-stage hybrid cross docking schedule problem. *Computers & Operations Research*, 36:2066–2073.

Christofides, N., Alvarez-Valdés, R., and Tamarit, J. (1987). Project scheduling with resource constraints: A branch-and-bound approach. *European Journal of Operational Research*, 29(3):262–273.

Dyer, M. E. and Wolsey, L. A. (1990). Formulating the single machine sequencing problem with release dates as a mixed integer problem. *Discrete Applied Mathematics*, 26:255–270.

Joslin, D. E. and Clements, D. P. (1999). Squeaky wheel optimization. *Journal of Artificial Intelligence Research*, 10(1):353–373.

Kravchenko, S. and Werner, F. (2011). Parallel machine problems with equal processing times: a survey. *Journal of Scheduling*, 14(5):435–444.

Laborie, P. and Nuijten, W. (2008). Constrained programming formulations and propagation algorithms. In Artigues, C., Demassey, S., and Néron, E., editors, *Resource-Constrained Project Scheduling*. Control Systems, Robotics and Manufacturing Series. ISTE Ltd. Chapter 4.

Lenstra, J., Kan, A. R., and Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362.

Li, Y., Lim, A., and Rodrigues, B. (2004). Crossdocking – JIT scheduling with time windows. *Journal of the Operational Research Society*, 55:1342–1351.

Maknoon, M. Y. (2013). *Scheduling material handling in cross-docking terminals*. PhD thesis, École polytechnique de Montréal.

Miao, Z., Lim, A., and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research*, 192:105–115.

van Belle, J., Valckenaers, P., and Cattrysse, D. (2012). Cross docking: State of the art. *Omega*, 40(6):827–846.

Yu, W. and Egbelu, P. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *International Journal of Production Economics*, 184:377–396.