TWO-DEDICATED-PARALLEL-MACHINE SCHEDULING APPROACH FOR A TWO-DOCK TRUCK SCHEDULING PROBLEM IN A CROSS-DOCKING CENTRE

Thiago Henrique Nogueira
Universidade Federal de Viçosa, Engenharia de Produção
Rodovia MG-230 Km 7, Rio Paranaíba MG
thiagoh.nogueira@ufv.br

Eliaquim Pereira Guimares Neto
Universidade Federal de Viçosa, Engenharia de Produção
Rodovia MG-230 Km 7, Rio Paranaíba MG
eliaquim.neto@ufv.br

Gustavo Ferreira Morais
Universidade Federal de Viçosa, Engenharia de Produção
Rodovia MG-230 Km 7, Rio Paranaíba MG
gustavo.f.morais@ufv.br

Martín Gómez Ravetti
Universidade Federal de Minas Gerais, Engenharia de Produção
Av. Antnio Carlos, 6627 - Pampulha - Belo Horizonte - MG
martin@ufmg.br

ABSTRACT

Cross-docking is a logistics strategy that minimizes the storage and picking functions of a warehouse. It is a result of freights that are unload from inbound trucks and directly load into outbound trucks, with little or no storage in between. The success of this technique relies on an efficient transshipment operation. This work undertakes a study of a two-dock truck scheduling problem in a cross-docking center as it is modeled as a two-dedicated-machine scheduling problem with implicit precedence constraint between stages. The objective is to minimize the makespan. A completion time and precedence formulation is proposed to model the problem as well as a heuristic for generating bounds. Results show that the mathematical model based on this new approach has efficient computational times. Similarly, the proposed heuristic outperformed current results in the literature for small, moderate and large size instances.

KEYWORDS. Truck scheduling, Cross-docking, Logistics.

1. Introduction

The environment of today's market, marked by increasingly fierce competition, the globalization of the economy and an accelerated technological revolution, has led companies to aim improvements in their systems, production, logistics and distribution. Therefore, greater efficiency and better services, and consequently a favorable position in the market are more likely to be achieved.

Recently, costumers have demanded better services, which translates into more accurate and timely shipments. Instead of waiting a week to get a product, most consumers expect express deliveries. Thus, managers have been under constant pressure to find alternatives and efficient ways to achieve high customer satisfaction. According to the [Apte and Viswanathan, 2000], as companies increasingly seek to their supply chains for new cost reduction opportunities, improved efficiencies, and customer satisfaction, many are finding that cross-docking meets all three goals.

A Cross-docking Center (CDC) is a logistic technique widespread throughout the world. Several well-known companies such as retail chains (Wal-Mart [Stalk et al., 1991]), mailing companies (UUPS [Forger, 1995]), automobile manufacturers (Toyota [Witt, 1998]) and less-than-truckload logistics providers ([Gue, 1999], [Kim et al., 2008]) have gained considerable competitive advantages. The main idea behind cross-docking is to receive products from different suppliers or manufacturers and consolidate them with for common final delivery destinations. In comparison to traditional warehouses, a cross-dock is managed with minimal handling and with little or no storage between unloading and loading of the goods. This practice can serve different goals: the consolidation of shipments, a shorter delivery lead time, the reduction of costs, etc.

A schematic representation of the process that takes place at cross docking terminals is illustrated in the Figure 1. Firstly, incoming trucks arrive at the yard of the cross dock, if the number of trucks is greater than the number of docks, some of them will have to wait in a queue on the yard until assignment, otherwise they can be directly assigned to a receiving door. Secondly, after being docked, goods of the inbound trailer are unloaded, scanned, sorted, moved across the dock and loaded onto outbound trucks for an immediate delivery elsewhere in the distribution chain. Once an outbound (inbound) truck is completely loaded (unloaded) the trailer is removed from the dock, replaced by another trailer and the course of action repeats.

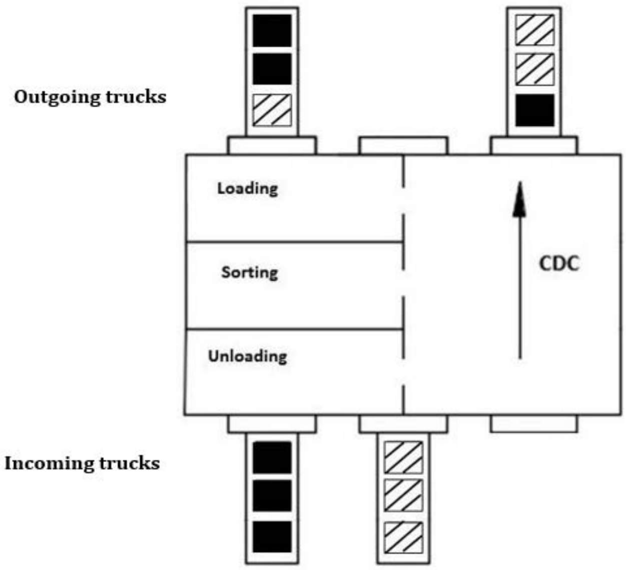

Figure 1: Schematic representation of a CDC.

In a conventional distribution center, five operations are usually carried out when managing products: receiving, sorting, storing, retrieving and shipping operations. All of which can be done consecutively or in a special order. As a result of applying a cross-docking system, the cost of both storage and products retrieval can be considerably reduced by consolidating the inbound and outbound trailer flows. Cross docks are perceived beneficial, as they reduce costs (warehousing costs, inventory-holding costs, handling costs, labor costs), shorter delivery lead-times (from supplier to customer, increase of throughput), improve customer service and customer satisfaction, reduce storage space, faster inventory turnover, fewer overstocks, reduce risk of loss and damage, etc. However, efficient transshipment processes and a careful planning of operations becomes indispensable within a CDC, where inbound and outbound flows need to be synchronized to keep the terminal storage as low as possible and on-time deliveries are ensured. Many articles in the literature develop computerized scheduling procedures, which have achieved good results (See examples in Table 1).

In this work, we propose a mixed integer linear programming formulation with implicit precedence constraint between stages for the two-dock truck scheduling problem in a cross-docking center. Our approach considers a novel perspective on the problem. Until now the problem has been modeled considering a flowshop environment, as in [Chen and Lee, 2009] or [Cota et al., 2016], in our work we model the problem as a parallel machine case with implicit cross-docking constraints, minimizing the makespan. The problem is denoted as $P2|M_j, CD|C_{max}$ and as $P2|CD|C_{max}$ is strongly NP-hard (See Pinedo [2008] and Chen and Lee [2009]), it is not difficult to see that our problem is also strongly NP-hard. Computational experiments show that the proposed model has efficient computational times when solving small problems.

A constructive heuristic based in parallel machine environment is also proposed and tested. The heuristic is able to efficiently generate lower and upper bounds simultaneously. Results are compared to the best heuristic proposed by Chen and Lee [2009], which is based on Johnson's algorithm and better solutions were consistently obtained for all instances tested. The article is organized as follows, an overview of related works is presented in Section 2. The proposed model is formulated in Section 3. A constructive heuristic is described in Section 4. Computational experiments are reported in Section 5. Finally, discussions and conclusions are drawn in Section 6.

2. Literature Review

Many decisions have to be made by cross-docking practitioners during the planning and operational phases of CDC's. The impact of these decisions on the efficiency of the operational phase is so significant that they have to be carefully taken. In the literature, several decision problems related to cross-docking are studied. The majority of these problems are concerned about decisions with effects on either long-term scenarios (strategic or tactical) or short-term scenarios (operational). The works of [Boysen, 2010] and [Van Belle et al., 2012] discuss several works of the existing literature on cross-docking problems and its own peculiarities. These authors define the following decision problems faced during the life cycle of a CDC: location of cross-docks; layout design; cross-docking networks; dock door assignment; vehicle routing; truck scheduling; temporary storage; other issues.

In accordance with [Van Belle et al., 2012], the truck scheduling problem considers the dock doors as resources (used by the trucks) that have to be scheduled over time. The problem decides on the succession of inbound and outbound trucks at the dock doors, where and when should the trucks be processed.

In this context, 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. Due to its high real-world significance, several truck scheduling procedures have been introduced during recent years, which all treat specific cross dock settings. For that reason, we highlight in 1, some outstanding works dealing with truck scheduling problems in a CDC. As proposed by [Boysen, 2010], a classification scheme for deterministic truck scheduling problems is presented for each work as well as its contribution.

Table 1: Previous specific research works for the truck scheduling problems. Based on [Boysen and Fliedner, 2010].

Publication	Notation	Complexity	Contribution
[Miao et al., 2009]	$[M\|limit, t_{io}\|\star]$	NP-hard	MM, HM, P
[Chen and Lee, 2009]	$[E2\|t_j = 0\|C_{max}]$	NP-hard	B, ES, P
[Chen and Lee, 2009]	$[E2\|t_j = 0\|C_{max}]$	NP-hard	P
[Chen and Song, 2009]	$[E\|t_j = 0\|C_{max}]$	NP-hard	MM, HS, B
[Boysen, 2010]	$[E\|p_j = p, no-wait, t_j = 0\|\sum T_o]$	Open	MM, HM, ES
[Boysen and Fliedner, 2010]	$[E\|t_{io}, f_{ix}\|\sum W_s U_s]$	NP-hard	MM, P
[Boysen and Fliedner, 2010]	$[E\|t_i = 0, f_{ix}\|\sum W_s U_s]$	NP-hard	P
[Boysen et al., 2010]	$[E2\|p_j = p, change\|C_{max}]$	NP-hard	MM, HS, HI, ES, P
[McWilliams, 2010]	$[E\|no-wait\|*]$	Open	HM
[Vahdani and Zandieh, 2010]	$[E2\|change\|C_{max}]$	NP-hard	MM, HM
[Melo and Araujo, 2010]	$[E\|t_j = 0\|C_{max}]$	NP-hard	HM
[Arabani et al., 2011]	$[E2\|change\|C_{max}]$	Open	HM
[Larbi et al., 2011]	$[E2\|pmtn\|*]$	Np-hard	MM, ES
[Alpan et al., 2011]	$[E\|pmtn\|*]$	Open	MM, ES
[Lira, 2013]	$[E\|t_j = 0\|C_{max}]$	NP-hard	HM
[Lima, 2014]	$[E2\|t_j = 0\|\sum C_j^2]$	NP-hard	MM, HS, HM
[Fonseca, 2015]	$[E2\|t_j = 0\|C_{max}]$	NP-hard	MM, ES, H
[Cota et al., 2016]	$[E\|t_j = 0\|C_{max}]$	NP-hard	MM, HS

The notation used in column "Contribution" is stated as: MM (mathematical model), HI (heuristic improvement procedure), HM (meta-heuristic), B (bound computation), HS (start heuristic for initial solution), ES (exact solution procedure), P (properties [e.g., complexity] of problem). In the Boysen's tupel notation, column "Notation", the fields are door environment, operational characteristics and the objective respectively.

Regarding strategic decisions, some of the articles in Table 1 consider a CDC composed of two machines (docks), one inbound and one outbound door, such as [Chen and Lee, 2009], [Boysen et al., 2010], [Vahdani and Zandieh, 2010], [Arabani et al., 2011], [Larbi et al., 2011], [Lima, 2014], [Fonseca, 2015]. Among these authors, only [Chen and Lee, 2009] and [Fonseca, 2015] aim to minimize the makespan, considering dedicated machines as well as no transshipment time within the CDC. Furthermore, these works study the truck scheduling problem in an analogous manner to the so-called two-machine flow-shop problem ($F2|CD|C_{max}$), considering precedence constraints between inbound and outbound, and preemption is not allowed. This problem is proved to be NP-hard in strong sense by [Chen and Lee, 2009].

The work [Fonseca, 2015] deals with the problem implementing a Lagrangian Relaxation by a Volume Algorithm with four constructive polynomial heuristics, which are incorporated to simultaneously generate upper and lower bounds. Although this approach solve the problem for instances of 84 outbound jobs with computational time reaching up to 13 seconds, the demonstrated results were not expressive as the medium GAP is approximately 40%. In [Chen and Lee, 2009], the authors have developed a branch and bound algorithm that uses polynomial algorithms to generate bounds. Computational results show that the algorithm can solve instances up to 60 trucks in a reasonable amount of time.

Our article deals with the operational decision problem of scheduling trucks on a single inbound and outbound docks. Although we consider the same settings and objective as in [Chen and Lee, 2009], this work considers the problem as a parallel-machine problem with precedence constraints, $P2|M_j, CD|C_{max}$. The mathematical model, as well as the heuristic, take advantage of the parallel environment. Thus, we aim to compare the strength of our heuristic and model with the best heuristic proposed by [Chen and Lee, 2009].

3. Mathematical Model

The objective of this section is to give a formal description of $P2|M_j, CD|C_{max}$ by proposing a completion time and precedence (CTP) formulation with implicit precedence constraint between stages. We use a close definition to [Chen and Lee, 2009], although our problem is considered as a parallel machine environment.

Sets:

M_j: not overlap dedicated machine sets, J_1 and J_2,

J_1: set of inbound trucks, $J_1 = \{1, 2, \ldots, n\}$,
J_2: set of outbound trucks, $J_2 = \{n+1, \ldots, n+m\}$,
J: set of all jobs, $J = \{J_1 \cup J_2\}$.

Input parameters:
n: number of jobs at unloading stage,
m: number of jobs at loading stage,
p_i: processing time of job $i \in J$,
P_j: set of precedent subsets jobs $i \in J_1$ corresponding to job $j \in J_2$ ($i \in P_j$),
S_i: set of successor subsets jobs $j \in J_2$ corresponding to job $i \in J_1$ ($j \in S_i$),
M: big number, $M = \sum_i^J p_i$.

Specific sets:
A_1: $\{\forall(i,j) \in J_1 \times J_1, i \neq j\}$,
A_2: $\{\forall(i,j) \in J_2 \times J_2, i \neq j\}$,
A: $A_1 \cup A_2 \cup \{\forall(i,j) \in J_1 \times J_2, i \neq j \text{ and } i \in P_j\}$. The set A eliminates the necessity of precedence relationships constraint set in the mathematical formulation.

Decision variables:
C_{max}: makespan,
C_j: completion time of job j,
y_{ij}: 1, if job j is processed after job i; 0, otherwise. For all y_{ij} with $i \in J_1, j \in J_2$ and $i \in S_j$, y_{ij} is fixed as 1.

Follows the completion time and precedence mixed integer linear programming model (CTP):

$$\min C_{max} \tag{1}$$

subject to

$$C_j \geq C_i + p_j - M(1 - y_{ij}), \forall(i,j) \in A, \tag{2}$$

$$y_{ij} + y_{ji} = 1, \forall(i,j) \in A_1 \cup A_2, \tag{3}$$

$$C_j \geq p_j, \forall j \in J, \tag{4}$$

$$C_{max} \geq C_j, \forall j \in J_2, \tag{5}$$

$$y_{ij} \in \{0,1\}, \forall i,j \in J, i \neq j, \tag{6}$$

$$C_j \geq 0, \forall j \in J, \tag{7}$$

$$C_{max} \geq 0. \tag{8}$$

The objective is to minimize the scheduling makespan (1). The constraint set (2) ensures that the completion time of job j happens only after the completion time of job i plus the processing time of job j when j is processed after i. The precedence constraint is implicitly presented in the definition of set A, as it ensures the schedule of jobs in the second stage only after the completion of their predecessors at the first stage. The constraint set (3) describes precedence relations between each pair of jobs i and j. Constraint set (4) ensures that completion time of job j is greater than or equal to its release date. Constraint set (5) defines the makespan, by computing the maximum completion time. Constraint sets (6), (7) and (8) specify the domains of each decision variable.

4. Construtive Heuristic MP1

In this work, we consider a CDC composed of two machines (docks), one inbound and one outbound door. However, all jobs in the second stage can only start being processed after all its predecessors are completed.

The MP1 heuristic starts by considering that all jobs in the second stage will have as release date the sum of the processing time of its precedent set, $r_j = \sum_{l \in S_j} p_l$. With this consideration, the scheduling problem on the second stage is modeled as $1|r|C_{max}$ and can be solved in polynomial time through the ERD rule (Earliest Release Date) [Pinedo, 2012]. By applying ERD, we obtain the sequence U_2 for the second stage, and a valid lower bound for the makespan.

Based in Property I developed by [Chen and Lee, 2009], given a specific sequence of jobs in the second stage, U_2, we can obtain a processing sequence in the first stage, U_1, that makes U_2 an optimal sequence of jobs in the second dock. U_1 is obtained by following the order of its successors: $P_1, P_2\ P_1, P_3|P_1 \bigcup P_2, \ldots, P_m\ (\bigcup_{j=1}^{m-1} P_j), j \in J_2$. To improve the solution, MP1 uses the sequence U_2 as input to the heuristic NEH (such as proposed by [Nawaz et al., 1983]). Finally, a local search procedure, based on swap and insertion moves (see [den Besten and Stützle, 2001; De Paula et al., 2007]), is applied to U_2 as refinement. It must be highlight that the local search procedure is not a polynomial-time algorithm. However, the computational results are efficient.

The heuristic receives as input data (J_1, S_1, J_2, P_2, p), and is detailed as follow:

- Step 1 : Compute the release date of each job in the second stage as the sum of its predecessors.
- Step 2 : Solve the problem $1|r|C_{max}$ on the second stage by applying the ERD rule. This is a valid lower bound.
- Step 3: Use U_2 as input to construct feasible solutions through the NEH heuristic, generating U_2'.
- Step 4 : Refine U_2' by local search procedure, obtaining U_2'' as result. This makespan is a valid Upper Bound.

Lemma 1. *The MP1 can be computed in* $O(max\{nm, mlog(m), m^2\})$.
PROOF. In Step 1 the release dates are calculated for each job at second stage ($O(nm)$). The Step 2 applies the ERD rule for the jobs of the second stage, and its complexity is $O(mlog(m))$. Step 3 construct feasible solutions through the NEH heuristic ($O(m^2)$). Finally, the Step 4 consists in a local search procedure ($O(m^2)$). Thus, the computational complexity of MP1 is $O(max\{nm, mlog(m), m^2\})$ assuming that the number of jobs is greater than the number of machines.

Theorem 1. $C^{UB}_{MP1} - C^* \leq \sum_{i \in J_1} p_i - P^{min}$, *where* C^* *denotes the optimal makespan,* C^{UB}_{MP1} *a feasible solution obtained by MP1 heuristic, and* $P^{min} = \min_{j \in J_2}\{\sum_{i \in J_1, i \in P_j} p_i\}$.
PROOF. Let C^{LB}_{MP1} be the optimal makespan to the auxiliary problem $1|r|C_{max}$. For any feasible solution to $P2|M_j, CD|C_{max}$, we can always construct a feasible solution for the corresponding auxiliary $1|r|C_{max}$ with job sequence exactly the same as that in the second stage of $P2|M_j, CD|C_{max}$, with smaller or equal makespan.

On one hand, with P^{min} and with the optimal makespan value, we can define:

$$C^* \geq P^{min} + \sum_{j \in J_2} p_j. \tag{9}$$

The makespan obtained by the heuristic MP1, in the worst scenario, is obtained as the sum of the processing times for both stages. Thus the following inequality is also valid,

$$C^{UB}_{MP1} \leq \sum_{j \in J_1} p_j + \sum_{j \in J_2} p_j. \tag{10}$$

Finally, subtracting C^* from C^{UB}_{MP1}, the following inequality is obtained,

$$C^{UB}_{MP1} - C^* \leq \sum_{j \in J_1} p_j - P^{min}. \tag{11}$$

□

5. Computational Experiments

To investigate the performance of the proposed model (CTP) and the constructive heuristic (MP1), artificial instances were generated varying the number of jobs and machines according to [Chen and Lee, 2009]. The heuristics are coded in C++ and the model is solved using AMPL and CPLEX 12.6 with default settings. Furthermore, the results are compared with those obtained by constructive heuristic developed in Chen and Lee [2009]. The tests are performed on a Linux Mint Edition with a double 2.0 GHz processor and 4GB memory.

5.1. Instances Generation

The instances used in this work were generated by [Fonseca, 2015], following the description found in [Chen and Lee, 2009]. The Table 2 depicts a summary of generated instances and its respective characteristics. Processing times of jobs present in the sets J_1 and J_2 are randomly defined, as well as its predecessors, in accordance with their uniform distribution shown in the table. The values that represent n and m were established a priory. For each size n, five instances with distinct size m are defined. Finally, for each pair of size n and m, 10 different problems are generated and therefore 50 different instances by row of the Table 2.

Table 2: Summary of generated instances for testing from Fonseca [2015], which is divided into groups 1 and 2, and subgroups, in accordance to the number of $jobs$ in the machine 1 ($Jobs\ M1(n)$). Furthermore, this table informs the number of jobs in the machine 2 ($Jobs\ M2(m)$), the maximum number of predecessors of $jobs\ j \in J_2(NP)$ and the processing time of $jobs\ j \in J\ (TP)$.

Group	Jobs M1(n)	Jobs M2(m)	NP	TP
1	5	3-4-5-6-7	U(1,4)	U(1,10)
	10	6-8-10-12-14	U(1,9)	U(1,10)
	20	12-16-20-24-28	U(1,19)	U(1,10)
	40	24-32-40-48-56	U(1,39)	U(1,10)
	60	36-48-60-72-84	U(1,59)	U(1,10)
2	5	3-4-5-6-7	U(1,4)	U(10,100)
	10	6-8-10-12-14	U(1,9)	U(10,100)
	20	12-16-20-24-28	U(1,19)	U(10,100)
	40	24-32-40-48-56	U(1,39)	U(10,100)
	60	36-48-60-72-84	U(1,59)	U(10,100)

5.2. Computational Results

In Tables 3 and 4, we report computational results when solving small, moderate and large instances with CTP and (CPT_Rl) (linear relaxation) models, the constructive heuristics MP1 and JB. It is worth highlighting that the constructive heuristic JB is proposed by [Chen and Lee, 2009] and we aimed to compare its strength with that (MP1) proposed in this work. The heuristic JB is proposed in two steps. Firstly, for an instance of $F2|CD|C_{max}$, the authors construct an instance of $F2||C_{max}$ with n jobs. The first stage is maintained unchangeable and the second stage is converted into n jobs. Secondly, Johnson's algorithm is applied in order to obtain a sequence in the first stage, generating a lower bound. From the schedule in the first stage, jobs $j \in J_2$ at second stage are scheduled as soon as possible, respecting the completion time of its predecessors. We run the above algorithm for the primary problem and its reverse problem and select the best solution. As a result, an Upper Bound for the problem is generated (See [Chen and Lee, 2009] for a more detailed explanation). In a similar way to the heuristics MP1, the NEH algorithm and Local search is used to refine JB results. For each case presented, the columns n and m show respectively the number of jobs in the first and second stage. Furthermore, best and average results over 10 instances are reported. In the same way, in each combination of number of jobs the column T(s) refers to CPU times in seconds to solve to optimality, the column TR(s) for the CPU times in seconds to solve the linear relaxation of the CTP, and the column Gap(%) is computed as $\frac{\text{(Upper Bound - Lower Bound)}}{\text{Upper Bound}}$.

The computer runs are limited to one hour of CPU time (3,600 s). To analyze the differences between the formulations, it is carried out a comparison of the optimality Gap, the linear relaxation Gap and CPU times. The linear relaxation Gap is defined as the relative difference between the best integer solution found for each instance and the linear programming relaxation value, (LP). In addition, the Gap of the constructive heuristics is calculated with their own bounds. Finally, after comparing and analyzing the heuristics JB and MP1 the best Gap value is highlighted in Tables 3 and 4.

The experiments showed that the CTP model proposed is efficient to solve to optimality instances with $n = 5$ until $n = 10$ $jobs$, as the best Gap values for each instance is mostly zero. It happened for both short processing times [1, 10] and large processing times [10, 100]. In addition, most instances are solved in a very low computational time. The CTP model is able to optimally solve instances with up to n = 10 and m = 14 considering the groups of instances 1 and 2, in an acceptable time. Likewise, as the number of jobs in both stages increase, n = 20 and m = 12, the Gap for CTP reaches weaker values using 3600 seconds. Considering the largest instance size, n = 60 and m = 84, the Gap is 93% in a computational time of 3,600s, for groups 1 and 2.

From the computational results, it is clear that (CTP_Rl) model is able to rapidly solve all instances but presenting poor GAP values. The relaxed problem starts with a Gap of approximately 30% for small instances and goes up to 96% in for large instances when considering the group of instances 1. Whereas, it goes from a Gap of approximately 35% to 97% when considering the group 2. This highlights the weakness of the relaxed problem (CTP_Rl).

The increase in the number of jobs results in a significant increase in the runtime of the models. This fact justifies the proposal of constructive heuristics to solve problems with larger instances. Here, the proposed MP1 is compared with the heuristic developed by [Chen and Lee, 2009], denominated as JB. The total average Gap for all the generated instances is calculated for both heuristics. The GAP average for group 1 is around 20% for JB and around 18% for MP1, whereas for group 2, it showed around 20% for JB and 19% for MP1. The heuristic here proposed, MP1, appeared to be the best option twice as more than JB, presenting the best results in approximately 69% of the cases while JB in approximately 31%.

Based on Tables 3 and 4, the heuristic MP1 showed better gap results than JB in most instance sizes and job combinations. Firstly, for $n = m$, MP1 gives the best performance in 82% of the instances, while the heuristic JB, reaches best Gaps in only 20% of the cases. Secondly, it is noticeable that when considering $n < m$, MP1 is the best in 85% of the instances, and JB in 18% of them. Finally, when $n > m$, MP1 and JB almost presented the same average results. Summarizing, the heuristic MP1 has its best performances when $n = m$ and $n < m$ for small, moderate and large instances, showing to be far better than JB. However, when $n > m$, the heuristic MP1 decreases its performance, although still outperforming JB.

6. Conclusion

In this work, we aim to find better practices to solve the two-dock truck scheduling problem in cross-docking facilities. The problem is modeled as a two-dedicated-machine scheduling problem with implicit precedence constraint between stages. A completion time and precedence (CTP) formulation is proposed and analyzed. The CTP relaxation (CTP_Rl) showed poor bounds, but its relaxation is able to generate solutions in less than a second. Furthermore, the CTP formulation managed to solve only a few instances at optimality.

Table 3: Computational results for the mathematical model and constructive heuristics, for TP(1,10)

Group 1 - Processing times [1, 10]										
n	m		CTP		CTP_RI		JB		MP1	
			GAP	T(s)	GAP	T(s)	GAP	T(s)	GAP	T(s)
5	3	Best	0,0%	0,02	29,4%	0,00	**0,0%**	0,00	**0,0%**	0,00
		Average	0,0%	0,03	43,0%	0,00	**6,8%**	0,00	11,4%	0,00
	4	Best	0,0%	0,03	45,5%	0,00	**0,0%**	0,00	2,6%	0,00
		Average	0,0%	0,04	53,3%	0,00	**7,7%**	0,00	13,7%	0,00
	5	Best	0,0%	0,02	41,9%	0,00	4,5%	0,00	**0,0%**	0,00
		Average	0,0%	0,06	52,8%	0,00	**9,8%**	0,00	12,2%	0,00
	6	Best	0,0%	0,06	48,1%	0,00	4,8%	0,00	**0,0%**	0,00
		Average	0,0%	0,13	58,1%	0,00	16,6%	0,00	**12,4%**	0,00
	7	Best	0,0%	0,12	53,8%	0,00	**0,0%**	0,00	**0,0%**	0,00
		Average	0,0%	720,20	61,1%	0,00	14,1%	0,00	**5,7%**	0,00
10	6	Best	0,0%	3600,00	51,0%	0,00	**0,0%**	0,00	**0,0%**	0,00
		Average	0,0%	3600,00	67,1%	0,00	11,8%	0,00	**9,2%**	0,00
	8	Best	0,0%	1622,97	64,4%	0,00	2,7%	0,00	**0,0%**	0,00
		Average	0,0%	3139,16	70,9%	0,00	13,3%	0,00	**13,2%**	0,00
	10	Best	0,0%	1031,68	67,8%	0,00	1,4%	0,00	**0,0%**	0,00
		Average	2,9%	2649,57	73,9%	0,00	15,9%	0,00	**10,0%**	0,00
	12	Best	0,0%	2479,81	73,0%	0,00	3,9%	0,00	**0,0%**	0,00
		Average	14,0%	3442,12	76,9%	0,01	14,0%	0,00	**8,6%**	0,00
	14	Best	8,0%	3600,00	75,0%	0,01	1,3%	0,00	**0,0%**	0,00
		Average	28,7%	3600,00	79,5%	0,01	10,8%	0,00	**5,6%**	0,00
20	12	Best	48,0%	3600,00	73,4%	0,01	**10,0%**	0,00	11,9%	0,00
		Average	54,3%	3600,00	80,7%	0,01	18,1%	0,00	**18,0%**	0,00
	16	Best	53,0%	3600,00	75,5%	0,01	16,4%	0,00	**11,7%**	0,00
		Average	58,9%	3600,00	84,0%	0,01	23,0%	0,00	**18,7%**	0,00
	20	Best	60,0%	3600,00	83,9%	0,01	14,3%	0,00	**9,3%**	0,00
		Average	65,9%	3600,00	86,4%	0,01	24,2%	0,00	**19,9%**	0,00
	24	Best	65,0%	3600,00	87,3%	0,02	2,4%	0,00	**1,3%**	0,00
		Average	69,3%	3600,00	88,5%	0,02	21,2%	0,00	**17,1%**	0,00
	28	Best	69,0%	3600,00	88,4%	0,02	1,6%	0,00	**1,1%**	0,00
		Average	72,2%	3600,00	89,6%	0,02	16,7%	0,00	**13,9%**	0,00
40	24	Best	80,0%	3600,00	83,6%	0,04	**15,7%**	0,00	17,8%	0,00
		Average	82,5%	3600,00	91,5%	0,04	**20,7%**	0,00	22,5%	0,00
	32	Best	83,0%	3600,00	92,1%	0,05	**21,3%**	0,00	22,3%	0,00
		Average	85,1%	3600,00	93,4%	0,06	**26,5%**	0,00	**26,5%**	0,00
	40	Best	86,0%	3600,00	92,4%	0,07	**26,1%**	0,00	26,8%	0,00
		Average	86,9%	3600,00	93,9%	0,08	31,1%	0,00	**28,9%**	0,00
	48	Best	87,0%	3600,00	94,1%	0,10	23,9%	0,00	**22,0%**	0,00
		Average	88,6%	3600,00	94,6%	0,10	29,6%	0,00	**27,7%**	0,00
	56	Best	89,0%	3600,00	94,8%	0,12	16,8%	0,00	**16,4%**	0,00
		Average	89,5%	3600,00	95,0%	0,13	23,3%	0,00	**22,2%**	0,00
60	36	Best	88,0%	3600,00	93,3%	0,10	**16,9%**	0,00	18,7%	0,00
		Average	90,0%	3600,00	95,0%	0,11	**22,8%**	0,00	23,1%	0,00
	48	Best	90,0%	3600,00	93,5%	0,14	26,0%	0,00	**25,7%**	0,00
		Average	91,2%	3600,00	95,3%	0,15	29,9%	0,00	**28,8%**	0,00
	60	Best	91,0%	3600,00	94,8%	0,18	**26,6%**	0,00	27,8%	0,00
		Average	91,9%	3600,00	96,0%	0,20	33,5%	0,00	**30,8%**	0,00
	72	Best	93,0%	3600,00	96,1%	0,23	**25,7%**	0,00	26,0%	0,00
		Average	93,2%	3600,00	96,4%	0,25	30,6%	0,00	**29,3%**	0,00
	84	Best	93,0%	3600,00	96,6%	0,28	**20,3%**	0,00	21,1%	0,00
		Average	93,7%	3600,00	96,8%	0,34	26,0%	0,00	**25,4%**	0,00
Total Average			48,8%	2749,72	78,1%	0,06	19,92%	0,00	**18,19%**	0,00

Table 4: Computational results for the mathematical model and constructive heuristics, for TP(10,100)

Group 2 - Processing time [10, 100]										
n	m		CTP		CTP_RI		JB		MP1	
			GAP	T(s)	GAP	T(s)	GAP	T(s)	GAP	T(s)
5	3	Best	0,0%	0,03	34,7%	0,00	**0,0%**	0,00	**0,0%**	0,00
		Average	0,0%	0,04	48,0%	0,00	**2,4%**	0,00	18,1%	0,00
	4	Best	0,0%	0,03	45,0%	0,00	**0,0%**	0,00	2,9%	0,00
		Average	0,0%	0,05	54,6%	0,00	**7,5%**	0,00	13,1%	0,00
	5	Best	0,0%	0,05	42,7%	0,00	3,8%	0,00	**0,0%**	0,00
		Average	0,0%	0,07	53,7%	0,00	9,9%	0,00	**9,7%**	0,00
	6	Best	0,0%	0,06	48,6%	0,00	5,7%	0,00	**0,0%**	0,00
		Average	0,0%	0,14	59,7%	0,00	16,1%	0,00	**10,5%**	0,00
	7	Best	0,0%	0,09	53,3%	0,00	**0,0%**	0,00	**0,0%**	0,00
		Average	0,0%	0,46	62,4%	0,00	13,9%	0,00	**5,5%**	0,00
10	6	Best	0,0%	4,31	53,6%	0,00	**0,7%**	0,00	1,0%	0,00
		Average	0,0%	170,00	68,1%	0,00	11,3%	0,00	**9,9%**	0,00
	8	Best	0,0%	31,21	63,3%	0,00	2,3%	0,00	**0,0%**	0,00
		Average	1,7%	1270,29	70,3%	0,00	17,0%	0,00	**11,0%**	0,00
	10	Best	0,0%	526,07	73,7%	0,00	2,2%	0,00	**1,0%**	0,00
		Average	5,0%	2408,28	76,2%	0,00	18,4%	0,00	**11,1%**	0,00
	12	Best	9,0%	3600,00	72,6%	0,00	3,1%	0,00	**0,0%**	0,00
		Average	19,3%	3600,00	77,3%	0,01	13,9%	0,00	**8,9%**	0,00
	14	Best	15,0%	3600,00	75,8%	0,01	2,0%	0,00	**0,0%**	0,00
		Average	34,1%	3600,00	80,3%	0,01	11,4%	0,00	**6,4%**	0,00
20	12	Best	53,0%	3600,00	72,5%	0,01	**10,2%**	0,00	12,5%	0,00
		Average	57,1%	3600,00	80,9%	0,01	**17,9%**	0,00	18,7%	0,00
	16	Best	53,0%	3600,00	76,2%	0,01	17,4%	0,00	**12,0%**	0,00
		Average	61,2%	3600,00	84,5%	0,01	23,3%	0,00	**19,6%**	0,00
	20	Best	58,0%	3600,00	83,6%	0,01	15,5%	0,00	**10,8%**	0,00
		Average	66,1%	3600,00	86,8%	0,01	24,8%	0,00	**19,9%**	0,00
	24	Best	65,0%	3600,00	87,7%	0,02	1,5%	0,00	**0,6%**	0,00
		Average	69,6%	3600,00	88,8%	0,02	21,7%	0,00	**17,1%**	0,00
	28	Best	70,0%	3600,00	89,1%	0,02	3,1%	0,00	**0,9%**	0,00
		Average	74,0%	3600,00	90,0%	0,02	17,4%	0,00	**15,3%**	0,00
40	24	Best	79,0%	3600,00	87,5%	0,04	**2,3%**	0,00	30,1%	0,00
		Average	81,3%	3600,00	91,5%	0,04	**8,6%**	0,00	39,9%	0,00
	32	Best	84,0%	3600,00	92,3%	0,05	**22,4%**	0,00	23,3%	0,00
		Average	85,5%	3600,00	93,3%	0,06	28,5%	0,00	**26,6%**	0,00
	40	Best	86,0%	3600,00	92,6%	0,07	27,4%	0,00	**26,6%**	0,00
		Average	86,9%	3600,00	94,1%	0,08	32,4%	0,00	**28,7%**	0,00
	48	Best	88,0%	3600,00	94,3%	0,08	23,0%	0,00	**21,4%**	0,00
		Average	88,6%	3600,00	94,7%	0,10	29,1%	0,00	**26,9%**	0,00
	56	Best	89,0%	3600,00	94,9%	0,12	18,4%	0,00	**18,0%**	0,00
		Average	90,0%	3600,00	95,1%	0,13	23,3%	0,00	**22,4%**	0,00
60	36	Best	89,0%	3600,00	93,6%	0,11	**17,8%**	0,00	19,8%	0,00
		Average	89,8%	3600,00	95,0%	0,12	**22,7%**	0,00	23,3%	0,00
	48	Best	90,0%	3600,00	93,7%	0,13	26,5%	0,00	**25,9%**	0,00
		Average	91,1%	3600,00	95,3%	0,15	29,6%	0,00	**29,2%**	0,00
	60	Best	91,0%	3600,00	94,8%	0,18	**28,0%**	0,00	28,9%	0,00
		Average	92,3%	3600,00	96,0%	0,20	33,6%	0,00	**30,6%**	0,00
	72	Best	93,0%	3600,00	96,1%	0,22	**26,0%**	0,00	26,4%	0,00
		Average	93,2%	3600,00	96,4%	0,26	30,5%	0,00	**29,2%**	0,00
	84	Best	93,0%	3600,00	96,6%	0,29	**20,2%**	0,00	21,1%	0,00
		Average	93,9%	3600,00	96,8%	0,33	25,9%	0,00	**25,1%**	0,00
Total Average			49,7%	2536,22	78,8%	0,06	19,65%	0,00	**19,06%**	0,00

Regarding the heuristic comparison, the proposed constructive heuristic, MP1, outperformed the JB heuristic developed by [Chen and Lee, 2009]. As showed in the results, the heuristic MP1 gives the best gap results for all instance groups and job combinations ($n < m, n > m$ and $n = m$).

The hypothesis tested in this work is basically the usage of a parallel machine setting to model the truck scheduling problem in a cross-docking facility, instead of using a flow-shop scenario as suggested in [Chen and Lee, 2009] and in many other works in the literature. Results helped us to infer that algorithms and models based on parallel machines tend to perform better than those based on flow-shop environments. We believe that the development of tools and models based on the proposed paradigm will lead to efficient algorithms and better results.

Acknowledgments

The authors acknowledge support from CNPq and FAPEMIG, Brazil.

References

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

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

Arabani, A. B., Ghomi, S. F., and Zandieh, M. (2011). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert systems with Applications*, 38(3):1964–1979.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1):32–41.

Boysen, N. and Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38(6):413–422.

Boysen, N., Fliedner, M., and Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR spectrum*, 32(1):135–161.

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(1):59–72.

Chen, F. and Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research*, 36(6):2066–2073.

Cota, P. M., Gimenez, B. M., Araújo, D. P., Nogueira, T. H., de Souza, M. C., and Ravetti, M. G. (2016). Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. *Computers & Industrial Engineering*, 95:135–143.

De Paula, M. R., Ravetti, M. G., Mateus, G. R., and Pardalos, P. M. (2007). Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics*, 18(2):101–115.

den Besten, M. and Stützle, T. (2001). Neighborhoods revisited: An experimental investigation into the effectiveness of variable neighborhood descent for scheduling. In *MIC2001–4th Metaheuristics International Conference*, p. 545–549.

Fonseca, G. B. (2015). O problema de sequenciamento de caminhes em um centro de crossdocking com duas maquinas.

Forger, G. (1995). Ups starts worlds premiere cross-docking operation. *Modern material handling*, 36(8):36–38.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*, 33(4):419–428.

Kim, C., Yang, K. H., and Kim, J. (2008). A strategy for third-party logistics systems: a case analysis using the blue ocean strategy. *Omega*, 36(4):522–534.

Larbi, R., Alpan, G., Baptiste, P., and Penz, B. (2011). Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers & Operations Research*, 38(6): 889–900.

Lima, M. F. (2014). O problema de sequenciamento de caminhes numa estacao de crossdocking com duas maquinas: Formulacao indexada no tempo, relaxacao lagrangeana e geracao de colunas.

Lira, E. G. (2013). Um algoritmo iterated greedy para o problema de sequenciamento de caminhes em centros de cross docking.

McWilliams, D. L. (2010). Iterative improvement to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 59(1):136–144.

Melo, B. M. R. and Araujo, D. P. M. (2010). Heuristicas construtivas para o sequenciamento de caminhoes em centros de cross docking.

Miao, Z., Lim, A., and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European journal of operational research*, 192(1):105–115.

Nawaz, M., Enscore, E. E., and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95.

Pinedo, M. L. (2008). *Scheduling: Theory, Algorithms, and Systems*. Springer.

Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media.

Stalk, G., Evans, P., and Shulman, L. E. (1991). Competing on capabilities: the new rules of corporate strategy. *Harvard business review*, 70(2):57–69.

Vahdani, B. and Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust metaheuristics. *Computers & Industrial Engineering*, 58(1):12–24.

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

Witt, C. E. (1998). Crossdocking: Concepts demand choice. *Material Handling Engineering*, 53 (7):44–49.