

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

# Hybrid Methods for Lot Sizing on Parallel Machines

Diego Jacinto Fiorotto Silvio Alexandre de Araujo Raf Jans

November 2014

**CIRRELT-2014-56** 

Bureaux de Montréal : Université de Montréal Pavillon André-Aisenstadt C.P. 6128, succursale Centre-ville Montréal (Québec) Canada H3C 3J7 Téléphone : 514 343-7575 Télécopie : 514 343-7121 Bureaux de Québec : Université Laval Pavillon Palasis-Prince 2325, de la Terrasse, bureau 2642 Québec (Québec) Canada G1V 0A6 Téléphone: 418 656-2073 Télécopie : 418 656-2624

www.cirrelt.ca





ÉTS UQÀ

UQÂM HEC MONTREAL



# Hybrid Methods for Lot Sizing on Parallel Machines

Diego Jacinto Fiorotto<sup>1,2</sup>, Silvio Alexandre de Araujo<sup>2</sup>, Raf Jans<sup>1,\*</sup>

- <sup>1</sup> Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
- <sup>2</sup> Departamento de Matemática Aplicada, Universidade Estadual Paulista, 15054-000 São José do Rio Preto, SP, Brazil

**Abstract.** We consider the capacitated lot sizing problem with multiple items, setup time and unrelated parallel machines, and apply Dantzig-Wolfe decomposition to a strong reformulation of the problem. Unlike in the traditional approach where the linking constraints are the capacity constraints, we use the flow constraints, i.e. the demand constraints, as linking constraints. The aim of this approach is to obtain high quality lower bounds. We solve the master problem applying two solution methods that combine Lagrangian relaxation and Dantzig-Wolfe decomposition in a hybrid form. A primal heuristic, based on transfers of production quantities, is used to generate feasible solutions. Computational experiments using data sets from the literature are presented and show that the hybrid methods produce lower bounds of excellent quality and competitive upper bounds, when compared with the bounds produced by other methods from the literature and by a high-performance MIP software.

**Keywords**. Lot sizing, parallel machines, reformulation, hybrid methods, Dantzig-Wolfe decomposition.

**Acknowledgements.** This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (Process numbers 2010/16727-9, 2013/00965-6 and 2011/22647-0) and by the Natural Sciences and Engineering Research Council of Canada (grant 342182-09). This support is gratefully acknowledged.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

<sup>\*</sup> Corresponding author: Raf.Jans@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec Bibliothèque et Archives Canada, 2014

<sup>©</sup> Fiorotto, de Araujo, Jans and CIRRELT, 2014

# 1 Introduction

This article deals with a lot sizing problem that consists basically of determining the size of production lots, i.e. the amounts of each item to be produced, in each of the periods in the planning horizon in a way that minimizes total production costs, respects resource availability and meets the demand of the items. The problem studied here involves the production of multiple items in a single stage. The production sector consists of unrelated parallel machines with limited capacity. The items can be produced on any of the machines and, several different items can be produced on the same machine in the same time period (large bucket model). At the start of the production of each type of item, there is a setup time and a setup cost for the machine being used and, the setup is sequence-independent.

The paper has the following contributions. First, we propose a way to obtain lower bounds that are stronger than the ones obtained by the traditional per-item Dantzig-Wolfe decomposition. Second, we extend two hybrid algorithms that combine Lagrangian relaxation and Dantzig-Wolfe decomposition and apply them to obtain the stronger lower bounds for the problem with unrelated parallel machines. Third, we improve the Lagrangian heuristic proposed by Fiorotto and Araujo (2014) to obtain better upper bounds. Finally, computational experiments are performed to show the quality of the upper bounds and lower bounds compared to others methods from the literature. A comparison shows that the new hybrid method together with the improved heuristic provides generally better gaps.

The paper is organized as follows. In Section 2, we provide a literature review on lot sizing problems on parallel machines and Dantzig-Wolfe decomposition. In Section 3, the classical formulation of the problem is presented along with the proposed reformulation. In Section 4, we present the techniques of Lagrangian relaxation and Dantzig-Wolfe decomposition applied to the lot sizing problem on parallel machines. Section 5 describes the proposed algorithms to calculate these lower bounds. In section 6, the Lagrangian heuristic used is summarized and, in Section 7, the computational results are presented. Finally in Section 8, we present our conclusions.

# 2 Literature Review

In this section, we will first discuss papers related to lot sizing problems on parallel machines and subsequently we discuss relevant papers involving Dantzig-Wolfe decomposition and column generation applied to lot sizing problems.

#### 2.1 Literature Review on Parallel Machine Lot Sizing

In practical production planning problems, parallel machines often need to be taken into account. Areas of production that consider parallel machines are the pharmaceutical industry (De Matta and Guignard, 1995), plastic sheet production (Mergaux and van Wassenhove, 1984), tile production (De Matta and Guignard, 1994), the tire industry (Jans and Degraeve, 2004b), bottling of liquids and others (Carreno, 1990) and packaging (Marinelli, 2007).

Considering the problem with identical parallel machines, Lasdon and Terjung (1971) propose a heuristic for a lot sizing and scheduling problem with no machine setup time. Carreno (1990) proposes a heuristic for the Economic Lot Scheduling Problem (ELSP), i.e. with a constant demand rate, with setup times for parallel machines and solves problems with one hundred items and ten machines in fast computational times. Jans (2009) proposes new constraints to break the symmetry that is present due to the identical machines and tests his approach using a network reformulation for the problem. Tempelmeier and Buschkuhl (2009) consider the multi-stage problem with setup carry-over (a setup is maintained between adjacent periods) and develop a Lagrangian heuristic.

For the unrelated parallel machines case, Toledo and Armentano (2006) relax the capacity constraints and propose a Lagrangian heuristic to solve the problem. An initial solution is obtained by minimizing the Lagrangian problem, which is normally infeasible. In an attempt to make it feasible production is shifted between periods and machines, moving the production that exceeds the capacity and looking for feasible solutions that minimize the cost. Fiorotto and de Araujo (2014) study the same problem. The authors use a strong reformulation of the problem and instead of the capacity constraints, they relax the demand constraints using Lagrangian relaxation. They also propose a heuristic to find feasible solutions and compare their results with Toledo and Armentano (2006).

Multi-stage problems with unrelated parallel machines were studied in Ozdamar and Birbil (1998), who present a generic model in which the multi-stage case can be considered. Three hybrid heuristics are developed, in which a tabu search algorithm is used to make the problem feasible and improve the solutions. Stadtler (2003) and Helber and Sahling (2010) also analyze the multi-stage problem. Stadtler (2003) proposes a period decomposition heuristic and, to solve each subproblem, a reformulation based on the facility location problem is used. Helber and Sahling (2010) propose a fix-and-optimize approach and obtain better results than those obtained by Stadtler (2003).

Some research in the literature deals with the problem of parallel machines and sequencedependent setup costs and times. Some of the research papers consider small bucket models, where only one item can be produced per machine per period. Salomon et al. (1991) study the Discrete Lot Sizing and Scheduling Problem (DLSP) with parallel machines, and analyze the complexity for the cases of identical and non-identical machines. The authors also present some solution algorithms. Kang et al. (1999) propose a method based on column generation and branch-and-bound. Belvaux and Wolsey (2000) describe a generic model and an optimization system that is capable of solving a wide range of lot sizing problems including special cases with different items and parallel machines. Meyr (2002) present a general model that consists of an extension of the General Lot Sizing and Scheduling Problem (GLSP) model for the case in which both setup cost and time are sequence-dependent. Fandel and Stammen-Hegener (2006) also present a model based on the GLSP model and consider the multi-stage case. Marinelli (2007) proposes a solution approach for a real capacitated lot sizing and scheduling problem with parallel machines and shared buffers, arising in a packaging company producing yoghurt. Finally, Meyr and Mann (2013) propose a heuristic for the lot sizing and scheduling problem on parallel machines. Different decomposition approaches are proposed and compared with results from the literature.

# 2.2 Literature Review on Dantzig-Wolfe Decomposition

Dantzig-Wolfe decomposition and column generation has been used to find good quality lower bounds for lot sizing problems. Before the seminal paper of Dantzig and Wolfe (1960), Manne (1958) had already implicitly applied the ideas of decomposition for the lot sizing problem with dynamic demand considering several items and capacity constraints. Manne proposed a linear programming model that explicitly considers all possible production schedules. Lambrecht and Vanderveken (1979), Bitran and Matsuo (1986) and Degraeve and Jans (2007) further discuss the formulation proposed by Manne (1958).

Degraeve and Jans (2007) show that the decomposition proposed by Manne, while valid to calculate a strong lower bound, has a structural deficiency when it aims to solve the problem with integrality constraints. The set of feasible solutions for Manne's formulation with integrality constraints, is only a subset of feasible solutions for the original integer problem. The main reason for this deficiency is that the solution for the subproblems, i.e. a new column, consists of both setup and production variables and in Manne's formulation the decisions of the setup automatically determine the production quantity decisions according to the Wagner-Whitin property. However, it is very likely that the optimal solution for the capacitated problem will not have this property.

Dzielinski and Gomory (1965) use column generation to handle the formulation with the large number of variables proposed by Manne (1958). Indeed, Manne's formulation is the full master problem obtained when one applies Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) to a formulation with a smaller number of variables. Dzielinski and Gomory (1965) also note that the subproblems that must be solved to generate columns are equivalent to the

problem studied by Wagner and Whitin (1958).

Lasdon and Terjung (1971) develop a column generation approach to handle large problems. Algorithms of this type are also addressed by Bahl (1983), Cattrysse et al. (1990), Salomon et al. (1993) and Huisman et al. (2005).

Hindi (1995) presents a heuristic including variable redefinition and column generation. To calculate the upper bound he solves a minimum cost flow problem. Hindi (1996) combines the ideas of linear relaxation, column generation, minimum cost flow network and Tabu search in a hybrid algorithm. Haase (2005) also solves the lot sizing problem by column generation and finds improved lower bounds.

Considering that both Lagrangian relaxation and Dantzig-Wolfe decomposition have advantages and disadvantages, Huisman et al. (2005) discuss two different ways to combine these two methods in hybrid algorithms to solve the linear relaxation of the master problem. In the first approach they apply Lagrangian relaxation to solve the master problem, i.e., no simplex method is used. In the second approach, they use the simplex method to generate the optimal dual variables of the master problem and the Lagrangian relaxation approach to generate columns. In this latter approach the Lagrangian relaxation is applied to the compact formulation. The ideas are illustrated using the lot sizing problem.

Pimentel et al. (2010) consider the lot sizing problem with setup time and apply the Dantzig-Wolfe decomposition to the classical formulation in two different ways: item decomposition and period decomposition. Furthermore, a third decomposition is presented which applies decomposition per item and period simultaneously. The authors conclude that this last approach provides better lower bounds than those obtained by the other decompositions. They implemented three branch-and-price algorithms to solve the three decomposition models.

de Araujo et al. (2013) present a transformed reformulation and valid inequalities that speed up column generation and Lagrangian relaxation for the capacitated lot sizing problem with setup times (CLST) and show theoretically how both ideas are related to dual space reduction techniques. Finally, the authors propose a combination of the two methods proposed by Huisman et al. (2005). This approach obtains good computational results and avoids the need of a linear programming optimization package.

The aim of this paper is to find good lower and upper bounds for the single stage problem with unrelated parallel machines extending the ideas proposed in Huisman et al. (2005), de Araujo et al. (2013) and Fiorotto and de Araujo (2014). Furthermore, different from most other papers that use the traditional decomposition method, which is per item, we apply decomposition per period and machine.

# 3 Problem Formulations

# 3.1 Classical Formulation

We first present the classical formulation of the lot sizing problem on unrelated parallel machines. This formulation is based on the formulation of Trigeiro et al. (1989) for the single machine problem, and has been studied in Toledo and Armentano (2006).

For the mathematical formulation of the problem, we consider the following data:

 $I = \{1, ..., n\}$ : set of items;

 $J = \{1, ..., r\}$ : set of machines;

 $T = \{1, ..., m\}$ : set of periods;

 $d_{it}$ : demand of item *i* in period *t*;

 $sd_{it\tau}$ : the sum of the demand for item *i*, from period *t* until period  $\tau$  ( $\tau \ge t$ );

 $hc_{it}$ : unit inventory cost of item *i* in period *t*;

 $sc_{ijt}$ : setup cost for item *i* on machine *j* in period *t*;

 $vc_{ijt}$ : production cost of item *i* on machine *j* in period *t*;

 $fc_i$ : unit cost of initial inventory for item i;

 $st_{ijt}$ : setup time for item *i* on machine *j* in period *t*;

 $vt_{ijt}$ : production time of item *i* on machine *j* in period *t*;

 $Cap_{jt}$ : capacity (in units of time) of machine j in period t.

The decision variables are then defined as follows:

 $x_{ijt}$ : number of units produced of item *i* on machine *j* in period *t*;

 $y_{ijt}$ : binary variable, indicating the production or not of item i on machine

j in period t;

 $s_{it}$ : quantity of inventory of item *i* at the end of period *t*;

 $s_{i0}$ : initial quantity of inventory for item *i*.

Mathematical formulation:

$$Min\sum_{i=1}^{n} fc_{i}s_{i0} + \sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{t=1}^{m} (sc_{ijt}y_{ijt} + vc_{ijt}x_{ijt}) + \sum_{i=1}^{n}\sum_{t=1}^{m} hc_{it}s_{it}$$
(1)

Subject to:

$$s_{i,t-1} + \sum_{i=1}^{r} x_{ijt} = d_{it} + s_{it} \qquad \forall i \in I, t \in T$$

$$(2)$$

$$x_{ijt} \le \min\{(Cap_{jt} - st_{ijt})/vt_{ijt}, sd_{itm}\}y_{ijt} \qquad \forall i \in I, j \in J, t \in T$$
(3)

$$\sum_{i=1} (st_{ijt}y_{ijt} + vt_{ijt}x_{ijt}) \le Cap_{jt} \qquad \forall j \in J, t \in T$$
(4)

$$y_{ijt} \in \{0,1\}, \ x_{ijt} \ge 0, \ s_{it} \ge 0, \ s_{i0} \ge 0, \ s_{im} = 0 \ \forall i \in I, j \in J, t \in T$$
 (5)

The objective function (1) minimizes the total setup, production, inventory and initial inventory costs. The constraints (2) guarantee the inventory balance in each period. To avoid infeasible problems, the model considers the possibility of initial inventory. However the cost  $fc_i$  for this inventory is very large. Next are the machine setup constraints (3) and the capacity limits (4). In order to make the formulation stronger, we limit the production for each item in constraints (3) by both the remaining demand and the maximum possible production with the available capacity. Finally, constraints (5) define the variables domains.

#### 3.2 Reformulation

Next we present a reformulation of the model (1)-(5) using the variable redefinition approach proposed by Eppen and Martin (1987), producing a formulation based on the shortest path problem. Each node on the graph represents a period, including a dummy period m+1. There is an arc between each pair of nodes and the arc between nodes t and q (q > t) represents the option of producing the sum of the demands between period t and period q-1 during period t. The cost of each arc corresponds to the total production and inventory cost associated with the variable. The objective is to find the shortest path from 1 to m+1.

For the reformulation the following parameters are defined:

 $cv_{ijtk}$ : cost of production and inventory holding to produce item *i*, on machine *j* in period *t* meeting the demand for periods *t* to *k*:

$$cv_{ijtk} = vc_{ijt}sd_{itk} + \sum_{s=t+1}^{k} \sum_{u=t}^{s-1} hc_{iu}d_{is};$$

 $ci_{ii}$ : cost of initial inventory of item *i* meeting demand for the periods 1 to period *t*:

$$ci_{it} = fc_i sd_{i1t} + \sum_{s=2}^{t} \sum_{u=1}^{s-1} hc_{iu} d_{is}.$$

There are also the following new variables for the model:

- $zv_{ijtk}$ : fraction of the production plan for item *i* on machine *j*, in which the production in period *t* meets the demand for the period *t* to period *k*;
- $w_{it}$ : fraction of the initial inventory plan for item *i* in which the demand is met for the first *t* periods.
- The reformulation based on the shortest path problem is as follows:

$$Min\sum_{i=1}^{n}\sum_{t=1}^{m}ci_{it}w_{it} + \sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{t=1}^{m}sc_{ijt}y_{ijt} + \sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{t=1}^{m}\sum_{k=t}^{m}cv_{ijtk}zv_{ijtk}$$
(6)

Subject to :

$$1 = \sum_{k=1}^{m} w_{ik} + \sum_{j=1}^{r} \sum_{k=1}^{m} z v_{ij1k} \qquad \forall i \in I$$
 (7)

$$w_{i,t-1} + \sum_{j=1}^{r} \sum_{k=1}^{t-1} z v_{ijk,t-1} = \sum_{j=1}^{r} \sum_{k=t}^{m} z v_{ijtk} \qquad \forall i \in I, t \in T \setminus \{1\}$$
(8)

$$\sum_{k=t}^{m} zv_{ijtk} \le y_{ijt} \qquad \forall i \in I, j \in J, t \in T$$
(9)

$$\sum_{i=1}^{n} st_{ijt}y_{ijt} + \sum_{i=1}^{n} \sum_{k=t}^{m} vt_{ijt}sd_{itk}zv_{ijtk} \le Cap_{jt} \qquad \forall j \in J, t \in T$$
(10)

$$y_{ijt} \in \{0,1\}, \quad w_{it} \ge 0 \qquad \qquad \forall i \in I, j \in J, t \in T$$

$$(11)$$

$$zv_{ijtk} \ge 0 \qquad \qquad \forall i \in I, j \in J, t \in T \ \forall k \in T, k \ge t$$
(12)

The objective function (6) minimizes the sum of the setup, production and inventory costs including initial inventory costs. The constraints (7) and (8) define the flow balance constraints for the shortest path network. For each item, one flow unit is sent through the network, imposing that the demand for each product is met on time. Constraints (9) are the setup forcing constraints. The capacity constraints (10) limit the total setup and production times to the available capacity in each period and for each machine. Constraints (11) and (12) define the variable domains.

Note that this model is an adaptation of the shortest path reformulation that was originally proposed by Eppen and Martin (1987) for the case without capacity constraints. Due to the capacity constraints, the variables  $w_{it}$  and  $zv_{ijtk}$  can be fractional. The interpretation of a fractional value, for instance 0.3 for the variable  $zv_{ijtk}$  is as follows: produce in period t, on machine j, 30% of the total demand from period t until period k.

# 4 Dantzig-Wolfe Decomposition and Lagrangian Relaxation Applied to the Lot Sizing Problem on Parallel Machines

In this section, we analyze the theoretical principles of Dantzig-Wolfe decomposition and Lagrangian relaxation applied to the lot sizing problem on parallel machines. The ideas proposed by Jans and Degraeve (2004a), Huisman et al. (2005) and de Araujo et al. (2013) are used and were extended for the problem being addressed.

In what follows next, we consider the LP relaxation of the Dantzig-Wolfe reformulation.

### 4.1 Dantzig-Wolfe Decomposition

The decomposition that is commonly used for the lot sizing problem, has as base the formulation (1)-(5). The capacity constraints (4) are the linking constraints and the setup and demand constraints plus the integrality conditions are put into subproblems. Thus, the problem is decomposed into lot sizing problems per item. The approach that will be used in this paper, takes as base the formulation (6)-(12) and the linking constraints will be the flow constraints (7) and (8). The problem is decomposed into independent subproblems per period and per machine containing the capacity and setup constraints plus the integrality conditions. Thus, the extreme points represent production plans for each period and machine. The columns are hence production plans indicating for a specific period and machine, which items are produced and in which quantities. These production plans are feasible with respect to the capacity constraints.

Formally, let  $S_{tj}$  be the set of all extreme point production plans for period t and machine j. Thus,  $zt_{tjq}$  is the new variable representing production plan q for period t on machine j. The resulting relaxed master problem then looks as follows:

$$v_{DWMP} = Min \sum_{i=1}^{n} \sum_{t=1}^{m} ci_{it} w_{it} + \sum_{t=1}^{m} \sum_{j=1}^{r} \sum_{q \in S_{tj}} ct_{tjq} zt_{tjq}$$
(13)

Subject to :

$$1 \le \sum_{k=1}^{m} (w_{ik} + \sum_{j=1}^{r} \sum_{q \in S_{1j}} a_{ij1kq} z t_{1jq}) \quad \forall i \in I, \ (\pi_{i1})$$
(14)

$$w_{i,t-1} + \sum_{k=1}^{t-1} \sum_{j=1}^{r} \sum_{q \in S_{kj}} a_{ijk,t-1,q} z t_{kjq}$$
  
$$\leq \sum_{k=t}^{m} \sum_{j=1}^{r} \sum_{q \in S_{tj}} a_{ijtkq} z t_{tjq} \quad \forall i \in I, t \in T \setminus \{1\}, \ (\pi_{it})$$
(15)

$$\sum_{q \in S_{tj}} zt_{tjq} = 1 \qquad \forall t \in T, j \in J, \ (\mu_{tj})$$
(16)

$$zt_{tjq} \ge 0, w_{it} \ge 0 \tag{17}$$

The objective function (13) minimizes the total initial inventory cost and the cost of the production plans chosen for each period and machine. The constraints (14) and (15) are the flow constraints and correspond to the constraints (7) and (8) with a "smaller than or equal" sign instead of an equality. It means that in each node the sum of the outgoing arcs must be larger than or equal to the sum of the incoming arcs. We use this form because we will solve this master problem with Lagrangian relaxation, and these constraints will be dualized in the objective function with positive dual multipliers  $p_{it}$ . Therefore, the inequality interpretation makes it clearer to determine the sign of these relaxed constraints in the Lagrangian objective function. Note that we can make this interpretation since the cost coefficients  $c_{it}$ ,  $s_{c_{ijt}}$  and  $c_{v_{ijtk}}$  in the objective function (6) are positive. The constraints (16) are the convexity constraints, which guarantee the choice of a convex combination of the extreme points. Note that the flow constraints have dual variables called  $\pi_{it}$  and  $\mu_{tj}$ , respectively.

Due to the huge number of variables a column generation procedure is usually used to solve the master problem. The main idea of this solution method is to start the master problem with some columns (problem called restricted master (RMP)) and progress iteratively generating (with assistance of the subproblems) only the necessary columns until the solution of the restricted master is equal to the solution of the original master problem.

The parameters  $a_{ijtkq}$  and  $ct_{tjq}$  of the variables  $zt_{tjq}$  are defined by the solution of the subproblems. In the subproblems, the objective function minimizes the reduced costs. After rearranging the terms of the objective function, the subproblem for a specific period t and machine j is:

$$z_{SP_{tj}}(\pi,\mu) = Min \sum_{i=1}^{n} sc_{ijt}y_{ijt} + \sum_{i=1}^{n} \sum_{k=t}^{m-1} (cv_{ijtk} - \pi_{it} + \pi_{i,k+1})zv_{ijtk} + \sum_{i=1}^{n} (cv_{ijtm} - \pi_{it})zv_{ijtm} - \mu_{tj}$$
(18)

Subject to :

$$\sum_{i=1}^{n} st_{ijt}y_{ijt} + \sum_{i=1}^{n} \sum_{k=t}^{m} vt_{ijt}sd_{itk}zv_{ijtk} \le Cap_{jt}$$

$$\tag{19}$$

$$\sum_{k=t}^{m} zv_{ijtk} \le y_{ijt} \qquad \qquad \forall i \in I \tag{20}$$

$$y_{ijt} \in \{0,1\}, \quad zv_{ijtk} \ge 0 \qquad \qquad \forall i \in I, k \in T, k \ge t \tag{21}$$

The constraints of the subproblem (for period t and machine j) are the capacity (19) and setup (20) constraints plus the integrality constraint (21). These subproblems can be solved by the branch-and-bound method proposed by Jans and Degraeve (2004a).

Let  $(y_{ijt}^*, zv_{ijtk}^*)$  be the optimal solution for a subproblem. A new column is added to the restricted master problem, only if the optimal objective function value of the subproblem  $(z_{SP_{tj}}(\pi, \mu))$  is less than zero. A new column  $q \in S_{tj}$  will have the following parameters:

$$a_{ijtkq} = zv_{ijtk}^* \qquad \forall i \in I, \forall k \in T, k \ge t$$
(22)

$$ct_{tjq} = \sum_{i=1}^{n} sc_{ijt} y_{ijt}^{*} + \sum_{i=1}^{n} \sum_{k=t}^{m} cv_{ijtk} zv_{ijtk}^{*}$$
(23)

Finally, note that the master problem (13)-(17) is a linear programming problem and can be solved using the simplex method within the column generation procedure.

#### 4.2 Lagrangian Relaxation Applied to the Compact Formulation

Throughout this section, we discuss the Lagrangian relaxation applied to the compact formulation (6)-(12) as developed in Fiorotto and de Araujo (2014) for the lot sizing problem on parallel machines. We will call this procedure by LR/CF. In this Lagrangian relaxation, the constraints (7) and (8) are dualized in the objective function (6) with Lagrangian multipliers  $p_{it}$ . We define p as the vector of all  $p_{it}$  values. After reorganizing the terms of the objective function, the Lagrangian problem becomes:

$$v_{LR}(p) = Min \sum_{i=1}^{n} \sum_{j=1}^{r} \sum_{t=1}^{m} sc_{ijt}y_{ijt} + \sum_{i=1}^{n} \sum_{t=1}^{m-1} (ci_{it} - p_{i,1} + p_{i,t+1})w_{it} + \sum_{i=1}^{n} (ci_{im} - p_{i,1})w_{im} + \sum_{i=1}^{n} \sum_{j=1}^{r} \sum_{t=1}^{m} \sum_{k=t}^{m-1} (cv_{ijtk} - p_{it} + p_{i,k+1})zv_{ijtk} + \sum_{i=1}^{n} \sum_{j=1}^{r} \sum_{t=1}^{m} (cv_{ijtm} - p_{it})zv_{ijtm} + \sum_{i=1}^{n} p_{i1}$$
(24)

Subject to : (9)-(12)

The Lagrangian problem can be decomposed into independent subproblems for each period t and each machine j:

$$z_{LR_{tj}}(p) = Min \sum_{i=1}^{n} sc_{ijt} y_{ijt} + \sum_{i=1}^{n} \sum_{k=t}^{m-1} (cv_{ijtk} - p_{it} + p_{i,k+1}) zv_{ijtk} + \sum_{i=1}^{n} (cv_{ijtm} - p_{it}) zv_{ijtm}$$
(25)

Subject to : (19)-(21)

These subproblems are exactly the same as the subproblem (18)-(21) resulting from the Dantzig-Wolfe decomposition, except for a constant in the objective function. Indeed, this is always the case if the dualized constraints in the Lagrangian relaxation and the linking constraints of the Dantzig-Wolfe decomposition are the same (Huisman et al., 2005).

The Lagrangian multipliers  $p_{it}$  are updated by the subgradient optimization method (Camerini et al., 1975). The value of the objective function for the solution of the relaxed problem  $(v_{LR}(p))$ produces a lower bound for the original problem and the Lagrangian Dual problem gives the maximum lower bound  $v_{DL} = max_p v_{LR}(p)$ .

Note that the  $w_{it}$  variables are present in the overall Lagrange objective function (24), but not in the objective function of the subproblems (25) because they do not appear in the constraints of the subproblem. To minimize the overall Lagrangian objective function (24) we can easily calculate the optimal value of these variables according to the dual prices.

# 5 Hybrid Methods Applied to the Lot Sizing Problem on Parallel Machines

Although the possibility to combine column generation and Lagrangian relaxation has been known for long time, it has only recently been exploited in algorithms. Several papers from the literature have shown that the combination of these two techniques is a promising tool in the resolution of integer programming problems (Huisman et al., 2005).

It is known that the dual solutions of the master problem obtained when using column generation have a primordial role in the quality of this method. In the standard solution approach using the simplex method for solving the master problem, the optimal dual solution of the restricted master problem is used in the pricing problem. However, it has been pointed out in the literature that optimal dual solutions generated by the simplex algorithm typically cause an unstable behavior of the method. This occurs because these solutions are extreme point solutions and oscillate sharply from one iteration to another, delaying the progress of the method. To counter this behavior, strategies for stabilizing the dual solutions have been proposed in the literature, leading to more efficient variations of the column generation method (see for example, Ben Amor et al. (2007) and Gondzio et al. (2013)).

Following this trend, two different approaches that combine Lagrangian relaxation and Dantzig-Wolfe decomposition are explored in this section.

# 5.1 Lagrangian Relaxation Applied to the Extended Formulation (LR/EF)

Instead of using the simplex algorithm to solve the restricted master problem in order to obtain bounds and the optimal dual values of the extended formulation (13)-(17) (master problem) it is possible to use the Lagrangian relaxation applied to this formulation to approximate these values. The linking constraints of the master problem (flow constraints, i.e., (14)-(15)) are transferred to the objective function with the respective dual multipliers. The solutions of the Lagrangian subproblems provide a lower bound to the optimal (LP) relaxation value of the restricted master problem and to get good lower bounds the dual Lagrangian problem must be solved. Cattrysse et al. (1993), Jans and Degraeve (2004a) and de Araujo et al. (2013) apply this technique to solving variants of the capacitated lot sizing problem with a single machine.

Formally, to solve approximately the linear programming problem (13)-(17), the constraints (14)-(15) are dualized in the objective function (13).

$$v_{LRDW}(p) = Min \sum_{i=1}^{n} \sum_{t=1}^{m} ci_{it}w_{it} + \sum_{t=1}^{m} \sum_{j=1}^{r} \sum_{q \in S_{tj}} ct_{tjq} zt_{tjq}$$
$$- \sum_{i=1}^{n} p_{i1} (\sum_{k=1}^{m} w_{ik} + \sum_{k=1}^{m} \sum_{j=1}^{r} \sum_{q \in S_{1j}} a_{ij1kq} zt_{1jq} - 1)$$
$$- \sum_{i=1}^{n} \sum_{t=2}^{m} p_{it} (\sum_{k=t}^{m} \sum_{j=1}^{r} \sum_{q \in S_{tj}} a_{ijtkq} zt_{tjq} - w_{i,t-1} - \sum_{k=1}^{t-1} \sum_{j=1}^{r} \sum_{q \in S_{kj}} a_{ijk,t-1,q} zt_{kjq})$$
(26)

Hybrid Methods for Lot Sizing on Parallel Machines

Subject to :

$$\sum_{q \in S_{tj}} zt_{tjq} = 1 \qquad \forall t \in T, j \in J, \quad (u_{tj}) \qquad (27)$$
$$zt_{tjq} \ge 0, w_{it} \ge 0 \qquad (28)$$

The problem (26)-(28) can be easily solved by inspection. After each iteration of the subgradient optimization method, the multipliers  $p_{it}$  are approximations of the optimal dual variables  $(\pi_{it})$ . With this approximation for the vector  $\pi_{it}$  it is possible to calculate an approximation for the vector  $\mu_{tj}$  that represents the optimal multiplier for the convexity constraints (16) (Huisman et al., 2005):

$$u_{tj} = \min_{q \in S_{tj}} (ct_{tjq} - \sum_{i=1}^{n} \sum_{k=t}^{m} p_{it} a_{ijtkq} + \sum_{i=1}^{n} \sum_{k=t}^{m-1} p_{i,k+1} a_{ijtkq}) \quad \forall j \in J,$$

$$\forall t \in T \setminus \{m\}$$
(29)

$$u_{mj} = \min_{q \in S_{mj}} (ct_{mjq} - \sum_{i=1}^{n} p_{im} a_{ijmmq}) \qquad \forall j \in J$$
(30)

The approximated Lagrangian multipliers can be used in the subproblems (18)-(21) to generate new columns that are added to the restricted master problem and in the next step the optimal dual variables  $\pi_{it}$  and  $\mu_{tj}$  for the updated restricted master are approximated again by the Lagrangian relaxation.

For each approximation of the solution of the restricted master problem, a lower bound for the master problem can be computed replacing the optimal dual variables by the approximated ones, i.e.:

$$\sum_{t=1}^{m} \sum_{j=1}^{r} z_{SP_{tj}}(p, u) + v_{LRDW}(p) \le v_{DWMP}$$
(31)

This can be proven by starting from the Lagrangian relaxation  $v_{LR}(p)$  (24), which gives a valid lower bound for any p:

$$\begin{aligned} v_{LR}(p) &= \sum_{t=1}^{m} \sum_{j=1}^{r} z_{LR_{tj}}(p) - \sum_{t=1}^{m} \sum_{j=1}^{r} u_{tj} + \sum_{t=1}^{m} \sum_{j=1}^{r} u_{tj} + \sum_{i=1}^{n} \sum_{t=1}^{n} ci_{it} w_{it} \\ &- \sum_{i=1}^{n} p_{i1} (\sum_{k=1}^{m} w_{ik} - 1) - \sum_{i=1}^{n} \sum_{t=2}^{m} p_{it} (-w_{i,t-1}) \\ &= \sum_{t=1}^{m} \sum_{j=1}^{r} z_{SP_{tj}}(p, u) + \sum_{t=1}^{m} \sum_{j=1}^{r} u_{tj} + \sum_{i=1}^{n} \sum_{t=1}^{m} ci_{it} w_{it} \\ &- \sum_{i=1}^{n} p_{i1} (\sum_{k=1}^{m} w_{ik} - 1) - \sum_{i=1}^{n} \sum_{t=2}^{m} p_{it} (-w_{i,t-1}) \\ &= \sum_{t=1}^{m} \sum_{j=1}^{r} z_{SP_{tj}}(p, u) + \sum_{t=1}^{m-1} \sum_{j=1}^{r} min_{q\in S_{tj}} (ct_{tjq} - \sum_{i=1}^{n} \sum_{k=t}^{m} p_{it}a_{ijtkq} \\ &+ \sum_{i=1}^{n} \sum_{k=t}^{m-1} p_{i,k+1}a_{ijtkq}) + \sum_{j=1}^{r} min_{q\in S_{mj}} (ct_{mjq} - \sum_{i=1}^{n} p_{im}a_{ijmmq}) \\ &+ \sum_{i=1}^{n} \sum_{t=1}^{m} ci_{it}w_{it} - \sum_{i=1}^{n} p_{i1} (\sum_{k=1}^{m} w_{ik} - 1) - \sum_{i=1}^{n} \sum_{t=2}^{m} p_{it} (-w_{i,t-1}) \\ &= \sum_{t=1}^{m} \sum_{i=1}^{r} z_{SP_{tj}}(p, u) + v_{LRDW}(p) \end{aligned}$$

The advantage of approximating the optimal dual variables by the Lagrangian relaxation, is that in the case of alternative dual solutions, column generation algorithms tend to converge more quickly using dual variables produced by interior point methods than with extreme point dual variables calculated by the simplex method (Bixby et al., 1992; Barnhart et al., 1998). From this perspective, the Lagrangian multipliers can provide a better representation and speed up the convergence. Computational experiments performed in Jans and Degraeve (2004a) indicate that the use of the Lagrangian multipliers indeed speeds up the convergence and reduces the problem of degeneration. The Lagrangian relaxation also has the additional advantages that during the subgradient method, feasible solutions are possibly generated. Furthermore, the subgradient update is quick and easy to implement and finally, this procedure eliminates the need for a commercial LP optimizer.

Algorithm 1 shows the application of Lagrangian relaxation to the extended formulation in order to obtain a lower and upper bound:

Algorithm 1: Lagrangian Relaxation Applied to the Extended Formulation Input: Initial RMP; multipliers; max. iteration.

Output: Lower bound (LB); upper bound (UB).

- 1 Let  $LB = -\infty$ ,  $UB = +\infty$ ,  $p_{it} = u_{tj} = 0$ ;
- 2 While  $(z_{SP_{ti}}(p, u) < 0)$  do:

- 3 Lagrangian step counter := 0;
- 4 Do while (Lagrangian step counter < max. iteration)
- 5 Solve the problem  $v_{LRDW}(p)$  (26)-(28);
- 6 Update the multipliers  $p_{it}$  with the subgradient method;
- 7 Lagrangian step counter := Lagrangian step counter +1;
- 8 end(do while);
- 9 Use best approximation for the Lagrangian multipliers of  $\pi_{it}$  and calculate an approximation of  $\mu_{ti}$   $(u_{ti})$  with the formula (29)-(30);
- 10 Solve the subproblems (18)-(21) with the approximate dual prices  $p_{it}$  and  $u_{tj}$ ;
- 11 Apply a feasibility heuristic (see Section 6);
- 12 Update the bounds LB and UB;
- 13 If (subproblem value < 0) then add columns;
- 14 end (while)

# 5.2 Lagrangian relaxation Applied to the Extended and Compact Formulations (LR/EF/CF)

Following the ideas proposed in de Araujo et al. (2013), this approach is based on the observation that when the Lagrangian relaxation is obtained by dualizing exactly those constraints that are the linking constraints in the Dantzig-Wolfe decomposition, the same subproblem results. Indeed, the subproblem to calculate the minimum reduced cost in the Dantzig-Wolfe decomposition given by (18)-(21) and the Lagrangian subproblem (decomposition per periods and machines of the Lagrangian problem presented in Section 4.2) are de same except for a constant in the objective function. Consequently, the solutions generated by the Lagrangian relaxation (on the compact formulation) can be used to add new columns in the restricted master problem. In LR/EF/CF, we combine the approach of Section 5.1 with the ideas discussed above in this section.

The main idea of this combined method (LR/EF/CF) is to use the Lagrangian relaxation on the extended formulation (master problem) to approximate the dual variables of the RMP(as explained in Section 5.1), and afterwards use these variables as initial dual variables in a column generation procedure based on Lagrangian relaxation of the compact formulation (original formulation).

This method is represented in the Algorithm 2. After generating some initial columns, the Lagrangian relaxation on the extended formulation problem (26)-(28) is solved a first time with

the subgradient optimization method that provides an approximation  $p_{it}$  and  $u_{tj}$  for  $\pi_{it}$  and  $\mu_{tj}$ . Then, the subproblems (18)-(22) are solved. If all the reduced cost to generate columns are positive the procedure is stopped and the lower bound can be found (using the formula (31)). Otherwise, it moves to Lagrangian relaxation of compact formulation. The initial Lagrangian multipliers are equal to the current dual prices found by the solution of the problem (18)-(22)and the Lagrangian inner loop starts. The Lagrangian problem (24) is solved and the result provides a lower bound. The inner loop of Lagrangian iterations continues (and we check if we reached the maximum value). In each step new multipliers  $p_{it}$  are obtained with the subgradient optimization method and the Lagrangian problem is solved with the current dual prices and a new lower bound is obtained. If this value is better than the current lower bound, we update the current lower bound. Then, for each period t we check if we can find a column with negative a reduced cost. This provides a new column because the Lagrangian subproblem (25) is identical to the subproblem to generate columns (18)-(21). The reduced cost of each column should be checked using the dual prices of the last time that the restricted master problem was solved. After a fixed number of Lagrangian inner loops, if none of the generated column enters in the restricted master problem, the best approximation according to the predefined parameters, for the optimal solution is found. Otherwise, the columns with negative reduced costs are added to the master problem (if these columns are not yet present in the master problem). Next, we optimize again the restricted master problem with the new added columns using Lagrangian relaxation. After a predefined number of iterations of the subgradient optimization method we return to Lagrangian relaxation of the compact formulation using the new dual prices generated by the approximate solution of the master problem. The procedure is stopped when no columns price out.

#### Algorithm 2: Lagrangian Relaxation Applied to the Extended and

#### **Compact Formulation**

Input: Initial PMR; multipliers; max. iteration; max. it\_Lagrangian Output: Lower bound (LB); upper bound (UB)..

- 1 Let  $LB = -\infty$ ,  $UB = +\infty$ ,  $p_{it} = u_{tj} = 0$ ;
- 2 While  $(z_{SP_{t_i}}(p, u) < 0)$  do:
- 3 Lagrangian step counter := 0;
- 4 Do while (Lagrangian step counter < max. iteration)
- 5 Solve the problem  $v_{LRDW}(p)$  (26)-(28);
- 6 Update the multipliers  $p_{it}$  with the subgradient method;
- 7 Lagrangian step counter := Lagrangian step counter +1;
- 8 end(do while);
- 9 Use best approximation for the Lagrangian multipliers of  $\pi_{it}$  and

calculate an approximation of  $\mu_{tj}$  ( $u_{tj}$ ) with the formula (29)-(30);

- 10 Solve the subproblems (18)-(21) with the approximate dual prices  $p_{it}$  and  $u_{tj}$ ;
- 11 Apply a feasibility heuristic (see Section 6) and update the upper bound (UB);
- 12 If  $(z_{SP_{ti}}(p, u) < 0)$  Then

```
13 it\_Lagrangian = 0;
```

```
14 Do while (it\_Lagrangian \le max. it\_Lagrangian)
```

```
15 Calculate the Lagrangian bound (v_{LR}(p)) (24) and update the lower bound (LB);
```

16 Update the Lagrangian multipliers  $p_{it}$  using subgradient;

$$it_Lagrangian := it_Lagrangian + 1;$$

- 18 Add columns if  $z_{LRtj}(p)(25) < 0$  and not added yet;
- 18 end (do while);
- 20 end (if);
- 21 end (while);

# 6 Primal Heuristic

# 6.1 General Overview

To obtain a feasible solution (upper bound) we extended the feasibility heuristic proposed by Fiorotto and de Araujo (2014) by adding an initialization and an improvement stage and by making several changes. The initial solution is given by solving the Lagrangian relaxation applied to the flow constraints (7) and (8). The solutions of the subproblems are then grouped and, generally, the resulting solution is not a feasible solution for the problem (6)-(12), due to the fact that the flow constraints were not taken into consideration. In other words, the solution probably does not satisfy exactly the demand for all items and periods.

If the initial solution does not satisfy the demand constraints, the feasibility heuristic (containing a backward stage, a forward stage and an improvement stage) is applied over all the periods, making changes to the production plan, producing and, if necessary, removing excess production in its attempt to make the solution a feasible one. Note that feasibility heuristics based on production transfer have been applied to make the capacity constraints feasible (for instance, Trigeiro et al. (1989) and Toledo and Armentano (2006)). In general, the feasibility heuristic algorithm can be described as follows:

# Primal Heuristic Algorithm

• Stage 1: Backward Stage (free up capacity)

For each item, check if total production is in excess of total demand. To do this find the total inventory quantity for each period and check if the inventory of the last period is greater than zero. If it is, remove excess production.

# • Stage 2: Forward Stage

Starting with the first period and moving to the last, we check for each item if the demand is satisfied. If it is not, try to make the solution feasible in the following order:

- 2.1: Produce the quantity still needed with machines already set up (and checking the machines in increasing order of production cost) in the current period and if not possible, in previous periods;
- 2.2: Produce the remaining quantity by performing a new machine setup checking the machines in increasing order of average unit production cost in the current period or if not possible, in previous periods;
- 2.3: If, after all the above attempts, demand is still not met for a given item (i') in period t, we check among previously-made-feasible items for one which has inventory left at the end of period t and remove the corresponding excess production, thus freeing up capacity. Try to satisfy the demand for the given item (i'), by using this capacity.
- Stage 3: Improvement Stage

After apply the Backward and Forward Stage (B&F) for all Lagrangian iterations that improved the lower bound, we obtain (in most cases) a feasible production plan, i.e., a setup plan and according production quantities. Since these production quantities are determined by the B&F heuristic, they are not necessarily the optimal production plans for the given setup plan. Then, for the best found solution, we check if with the same setup schedule, we can find better production quantities than suggested by the B&F heuristic. We fix the setup variables to their current value, as given by the B&F heuristic solution, and solve the remaining LP with LINDO to determine the optimal according production quantities.

Following, each stage is described in detail.

# 6.2 Initialization

We would like to determine for each item which is the cheapest machine to produce on. This is not trivial, since we have to take into account both the unit production cost and the setup cost. The total cost, however, depends on the proposed production plan. In order to establish a ranking, we calculate an approximate average unit cost per item and per machine, assuming that we use the Economic Order Quantity (EOQ) as production quantity (Andriolo et al., 2014). The EOQ balances the holding and setup cost in a setting with constant demand. Therefore, we first calculate the average demand per period for each item:

$$AVD(i) = \frac{\sum_{t=1}^{m} d_{it}}{m}, \ i = 1, 2, ..., n$$

After this, we calculate the EOQ for each item and machine, i.e.,  $EOQ(i, j) = \sqrt{\frac{2 \times AVD(i) \times sc_{ijt}}{hc_{it}}}$  $\forall i \in I, j \in J$ . Note that in the data sets that we will use  $sc_{ijt}$ ,  $hc_{it}$ ,  $st_{ijt}$ ,  $vt_{ijt}$ , and  $Cap_{jt}$ are time invariant. To deal with the discrete time horizon, we set EOQ(i, j) := AVD(i) if  $EOQ(i, j) \le AVD(i).$ 

We need to consider two separate cases to calculate the approximate average unit cost  $(AC_{EOO})$  for each item and machine:

I) If  $EOQ(i, j) \leq \frac{(Cap_{jt} - st_{ijt})}{vt_{ijt}}$ , then  $AC_{EOQ}(i, j) = \frac{sc_{ijt}}{EOQ(i, j)} + vc_{ijt}$ ; II) However, if  $EOQ(i, j) > \frac{(Cap_{jt} - st_{ijt})}{vt_{ijt}}$ , then  $Q := \frac{(Cap_{jt} - st_{ijt})}{vt_{ijt}}$  and  $AC_{EOQ}(i, j) = \frac{sc_{ijt}}{Q} + \frac{c_{ijt}}{Q}$  $vc_{ijt};$ 

The idea behind the formula is to estimate the average production cost for an item on each machine taking into account both the unit production cost and the setup cost.

The values for  $AC_{EOQ}(i, j)$  establishes the production priority order of the machines for each item.

#### 6.3 **Backward Stage**

This first stage to make a solution feasible consists of freeing up capacity if we have excess production, and to do this, the production quantities for each item are analyzed and removed if necessary. Initially we calculate the inventory of each item at the end of each period:

$$\Delta(i,t) := \sum_{l=1}^{t} \sum_{j=1}^{r} x_{ijl} - \sum_{l=1}^{t} d_{il} \quad \forall i \in I, t \in T$$
  
If  $\Delta(i,t) < 0$  Then  $\Delta(i,t) := 0$ 

If the inventory at the end of the horizon is strictly positive, i.e., if  $\Delta(i,m) > 0$  for some i = 1, 2, ..., n then the excess production of this item must be eliminated. The aim is to remove the excess inventory at the end of the horizon by reducing the production quantities in the earliest possible period, without creating any (additional) backlog. The reason we look for the earliest period is that freeing up capacity in earlier periods provides more flexibility than freeing up capacity in later periods.

To do this, starting with the last period (m) and following the sequence t = m, m - 1, ..., 1, find the first period with zero inventory, indicated by t'. Then we calculate the total quantity that should be removed in the period (t' + 1). To do this, we determine the  $min\{\Delta(i, t)\}$  in the interval t' + 1, t' + 2, ..., m. Let  $t^*$  be the period for which we have this minimum.

Denoting this minimum by  $\Delta(i, t^*)$ , remove this amount from the production quantity in the period (t'+1). Look for a machine (j) with the greatest  $(AC_{EOQ})$  and then:

If  $min\{\Delta(i, t^*), x_{ij,t'+1}\} = x_{ij,t'+1}$  remove the total amount produced and consequently the setup for this machine and period. Thus, the remaining capacity is updated:

$$Cap'_{j,t'+1} := Cap'_{j,t'+1} + x_{ij,t'+1} \times vt_{ij,t'+1} + st_{ij,t'+1}$$

The values of  $\Delta(i, t^*)$ , the inventory level and the variables are also updated by:

$$\Delta(i,t) := \Delta(i,t) - x_{ij,t'+1}, t = t'+1, \dots, m;$$
$$x_{ij,t'+1} := 0; y_{ij,t'+1} := 0$$

Then move on to the other machines to remove any remaining excess production.

However, if  $Min\{x_{ij,t'+1}, \Delta(i, t^*)\} = \Delta(i, t^*)$ , remove the total amount that can be removed in this period and recalculate the inventory level and production quantity:

$$\begin{split} Cap'_{j,t'+1} &:= Cap'_{j,t'+1} + \Delta(i,t^*) \times vt_{ij,t'+1}; \ x_{ij,t'+1} := x_{ij,t'+1} - \Delta(i,t^*); \\ \Delta(i,t) &:= \Delta(i,t) - \Delta(i,t^*), t = t'+1, ..., m; \end{split}$$

If after this step the inventory in the last period is equal to zero, we have eliminated the production excess of this item. However, if the inventory of the last period is still greater than zero we do the same analysis again, i.e., starting with the last period (m), we find the first period (t) with zero inventory.

In the Figure 1, we provide an example with seven periods. In part (a) we see that there is excess inventory at the end of the horizon. Note that starting with the last period, the first period with zero inventory (t') is the period 3 and the minimum inventory  $\Delta(i, t^*)$  is 1 in period 5  $(t^*)$ . In part (b) we see the inventory quantity for all periods after removing 1 unit from production in period 4. Note that the inventory of the last period is still greater than zero, therefore we have to do the same analysis again.



Figure 1: Example of the backward stage

#### 6.4 Forward Stage

When the process of eliminating the excess production is over, we start to check if the demand is met and try to satisfy demand for pairs of items and periods for which demand is currently not met. For this, the analysis will be done from period 1 through to the last period (t = 1, 2, ..., m).

First, we determine the order in which the items are analyzed according to the opportunity cost (OC(i)) of each item using the decreasing order. We calculate  $OC(i) = AC_{EOQ}(i, j') - AC_{EOQ}(i, \bar{j})$ , where j' and  $\bar{j}$  are the most expensive and cheapest machine respectively for item i.

To check if demand is met, an array is needed that counts the total production for each item up to the given period. Initially  $\Omega(i) = 0 \quad \forall i \in I$ . Starting from period t = 1, go through all the items (according to the order calculated) calculating  $\Omega(i) := \Omega(i) + \sum_{j=1}^{r} x_{ijt}$  and do the following analysis.

If  $\Omega(i) \ge d_{it}$  the demand for (i, t) is met. Simply remove this specific demand from the total production, in other words,  $\Omega(i) := \Omega(i) - d_{it}$  and move on to analyze the next item. Observe that in this case we have  $\Omega(i) \ge 0$ .

If  $\Omega(i) < d_{it}$ , demand is not met for this pair (i, t), then the feasibility process needs to be started. The quantity of items that need to be produced is easily calculated:

$$\Phi = d_{it} - \Omega(i)$$

Once this amount has been calculated, we have three attempts to make the solutions feasible (stages 2.1, 2.2 and 2.3 in the overview of the heuristic in Section 6.1).

#### • Attempt 1 (Stage 2.1)

Start at the current period t and go back period by period to the first period ( $\tau = t, t-1, ...1$ ) checking first if the machines already setup for this item ( $y_{ij\tau} = 1$ ) have sufficient capacity to produce the quantity that is still needed to satisfy demand. We check over all periods  $\tau = t, t - 1, ...1$  and within each time period we search over all machines (on which a setup is done) according to the increasing order of the value of unit production cost. Note that here we use the unit production cost instead  $AC_{EOQ}$  because the setup is already done.

Therefore, if  $y_{ij\tau} = 1$ , that is, if the machine is already set up, we calculate the maximum amount that can be produced with the remaining capacity, and compare this to the amount we still need to produce. We calculate the minimum of these two values:

$$Min\{\Phi, \frac{Cap'_{j\tau}}{vt_{ij\tau}}\}$$

Then, if  $Min\{\Phi, \frac{Cap'_{j\tau}}{vt_{ij\tau}}\} = \Phi$  the machine can produce the required amount to satisfy demand for this pair (i,t). Therefore, the values are updated by:

$$Cap'_{j\tau} := Cap'_{j\tau} - \Phi \times vt_{ij\tau}; \ x_{ij\tau} := x_{ij\tau} + \Phi; \ \Phi := 0; \ \Omega(i) := 0$$

However, if  $Min\{\Phi, \frac{Cap'_{j\tau}}{vt_{ij\tau}}\} = \frac{Cap'_{j\tau}}{vt_{ij\tau}}$ , there is still a quantity which needs to be produced. So we update the values:

$$Cap'_{j\tau} := 0; \ x_{ij\tau} := x_{ij\tau} + \frac{Cap'_{j\tau}}{vt_{ij\tau}}; \ \Phi := \Phi - \frac{Cap'_{j\tau}}{vt_{ij\tau}}$$

Next we check the next machine in the list or - if we have checked all the machines - move back to the previous period and repeat the same analysis.

#### • Attempt 2 (Stage 2.2)

If after checking all previous periods and machines with a setup, there is still some demand which is not met, we check the possibility of adding a setup to another machine but only if it can produce all of the remaining quantity required to satisfy demand. Again starting from  $\tau = t$  to  $\tau = 1$  and checking the machines on which no setup is done for item *i* (according to the order calculated by  $AC_{EOQ}$ ) if there is a period  $\tau$  and a machine *j* such that  $Cap'_{j\tau} \ge \Phi \times vt_{ij\tau} + st_{ij\tau}$ then adjust the production accordingly:

$$x_{ij\tau} := x_{ij\tau} + \Phi; \ y_{ij\tau} := 1; \ Cap'_{i\tau} := Cap'_{i\tau} - (\Phi \times vt_{ij\tau}) - st_{ij\tau}; \ \Omega(i) := 0$$

However, if there is still no single machine with enough capacity to produce the total missing demand, we repeat the loop and open up the possibility of setting up several machines. At this stage, the criteria used to determine the order of which machine to set up is again the order calculated according to the average unit production cost  $(AC_{EOQ})$ .

#### • Attempt 3 (Stage 2.3)

If after all the previous attempts, it is still not possible to meet the demand for a given item (i) in a given period (t), look among previous items that have already been made feasible for

one that has inventory at the end of period t, in other words, one for which the value of omega is positive. If it is found, make the necessary adjustments, i.e., remove the excess production of the found item (freeing up capacity) and try to produce the required amount of item (i). Thus, we search in the following order  $\tau = t, t - 1, ...1$  and within each time period we search over all items according to the decreasing order calculated by  $AC_{EOQ}$  and then:

If  $\Omega(i') > 0$  and  $y_{i'j\tau} = 1$ , in other words, an item with a quantity in inventory and a machine and period already setup for this item was found, do the following analysis:

If  $\Omega(i') \ge x_{i'j\tau}$  then, as well as freeing up capacity, a setup of the machine can be removed i.e.:

$$Cap'_{j\tau} := Cap'_{j\tau} + x_{i'j\tau} \times vt_{i'j\tau} + st_{i'j\tau}; \ y_{i'j\tau} := 0; \ \Omega(i') := \Omega(i') - x_{i'j\tau}; \ x_{i'j\tau} := 0$$

However, if  $\Omega(i') < x_{i'j\tau}$ , then just free up the capacity:

$$Cap'_{j\tau} := Cap'_{j\tau} + x_{i'j\tau} \times vt_{i'j\tau}; \quad x_{i'j\tau} := x_{i'j\tau} - \Omega(i'); \quad \Omega(i') := 0$$

With the capacity freed, calculate the total quantity of items (i) that can be produced and check again to see if it is possible to produce the required amount to meet demand in the following way:

If the machine is already set up for the item i, then calculate:

$$Min\{\Phi, \frac{Cap'_{j\tau}}{vt_{ij\tau}}\}$$

However, if there is no setup for item i, calculate:

$$Min\{\Phi, \frac{(Cap'_{j\tau} - st_{ij\tau})}{vt_{ij\tau}}\}$$

Then, if the minimum is  $\Phi$ , a quantity sufficient to satisfy the demand for this pair  $(i, \tau)$  can be produced and so we adjust the production amount accordingly:

$$Cap'_{j\tau} := Cap'_{j\tau} - \Phi \times vt_{ij\tau} - st_{ij\tau}; \ x_{ij\tau} := x_{ij\tau} + \Phi, \ y_{ij\tau} = 1$$

If the machine was already set up, we do not need to subtract the setup time.

However, if the minimum is  $\frac{Cap'_{j\tau}}{vt_{ij\tau}}$  or  $\frac{(Cap'_{j\tau}-st_{ij\tau})}{vt_{ij\tau}}$ , there is still a quantity left to be produced and so we add the quantity that can be produced, for example, if the machine is already set up for item *i*:

$$Cap'_{j\tau} := Cap'_{j\tau} - \frac{Cap'_{j\tau}}{vt_{ij\tau}} \times vt_{ij\tau}; \ x_{ij\tau} := x_{ij\tau} + \frac{Cap'_{j\tau}}{vt_{ij\tau}}$$

If the machine is not set up for item *i* we have to use  $\frac{(Cap'_{j\tau} - st_{ij\tau})}{vt_{ij\tau}}$  instead  $\frac{Cap'_{j\tau}}{vt_{ij\tau}}$ . After that, we move on to the next item (already made feasible) with a quantity in inventory and check the same conditions.

Once these feasibility stages have been performed for all items, move on to the next period of the forward stage and start the analysis again, for each item from first to last, trying to make the solution feasible. Repeat this procedure until the last period of the forward stage.

The overall pseudo code for this stage is as follows. To simplify the reading we will denote by i := 1, ..., n the decreasing order of the items according to the opportunity cost, j := 1, ..., rthe increasing order of the machines according to the value of production cost and k := 1, ..., rthe increasing order of the machines according to the value of  $AC_{EOQ}$ . All the values of  $\Omega(i)$ are initialized at zero.

for 
$$t := 1, ..., n$$
 do  
for  $i := 1, ..., n$  do  
for  $j := 1, ..., r$  do  
 $\Omega(i) := \Omega(i) + x_{ijt}$   
end  
if  $(\Omega(i) > d_{it})$  then  
 $\Omega(i) := \Omega(i) - d_{it}$   
else  
 $\Phi := d_{it} - \Omega(i)$   
for  $\tau := t, t - 1, ..., 1$   
for  $j := 1, ..., r$   
if  $(y_{ij\tau} = 1 \text{ and } \Phi > 0)$  then  
Calculate the whole amount that can be produced on this  
machine in this period, updating of the necessary parame-  
ters;  
end  
end  
for  $\tau := t, t - 1, ..., 1$   
for  $k := 1, ..., r$   
if  $(y_{ik\tau} = 0 \text{ and } \Phi > 0)$  then  
First try to find a machine that can produce all quantity to  
meet demand. If we can not find, we go back to this loop  
and open the possibility to make more than one setup,

updating of the necessary parameters;

```
end
         end
       end
       for \tau := t, t - 1, ..., 1
         for q := 1, ..., i - 1
            for k := r, ..., 1
              if (\Phi > 0, \Omega(q) > 0 and y_{qk\tau} = 1) then
                 Remove the excess inventory and try to produce the re-
                 quired amount of item i, updating of the parameters;
              end
            end
         end
       end
    end (if)
  end (for i)
end (for j)
```

Finally, at the end of the forward stage, we calculate the objective function value of the solution, if the solution becomes feasible, i.e. if all the demand is met.

## 6.5 Improvement Stage

This last stage consists of trying to improve the objective function value of the solution. We fix the best setup plan found and solve the remaining LP problem using the LINDO package solver, i.e., if we denote by  $y_{ijt}^*$  the value of the setup variables found by the heuristic, we have to solve the following problem:

$$Min\sum_{i=1}^{n}\sum_{t=1}^{m}ci_{it}w_{it} + \sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{t=1}^{m}\sum_{k=t}^{m}cv_{ijtk}zv_{ijtk}$$
(32)

2

Subject to :

$$1 = \sum_{k=1}^{m} w_{ik} + \sum_{j=1}^{r} \sum_{k=1}^{m} z v_{ij1k} \qquad \forall i \in I$$
(33)

$$w_{i,t-1} + \sum_{j=1}^{r} \sum_{k=1}^{t-1} z v_{ijk,t-1} = \sum_{j=1}^{r} \sum_{k=t}^{m} z v_{ijtk} \qquad \forall i \in I, t \in T/\{1\}$$
(34)

$$\sum_{k=t}^{m} zv_{ijtk} \le y_{ijt}^* \qquad \forall i \in I, j \in J, t \in T$$
(35)

$$\sum_{i=1}^{n} \sum_{k=t}^{m} v t_{ijt} s d_{itk} z v_{ijtk} \le Cap_{jt} - \sum_{i=1}^{n} s t_{ijt} y_{ijt}^* \qquad \forall j \in J, t \in T$$
(36)

$$zv_{ijtk} \ge 0, \quad w_{it} \ge 0 \qquad \forall i \in I, j \in J, t \in T \ \forall k \in T, k \ge t$$

$$(37)$$

Finally, we remove possible setups that are not being used. We check for all the constraints (35) if there is a setup  $(y_{ijt}^* = 1)$  and  $\sum_{k=t}^{m} zv_{ijtk} = 0$ , and remove the setup if this is the case.

Note that the heuristic is applied to the solutions obtained by the Lagrangian subproblems. Consequently, the quality of the solution of the heuristic is strongly linked to quality of the production plans determined by these subproblems.

# 7 Computational Results

The algorithms described in the previous sections were tested on a total of 2160 instances proposed in Toledo and Armentano (2006). The 2160 instances are divided into 8 different types of classes that are generated with high and low values for the setup costs (HS or LS), setup times (HT or LT) and with normal and tight capacity (NC or TC). Then, the class NCLSLT, for example, refers to instances with normal capacity, low setup costs and low setup times. The notation for the other classes follows the same reasoning. For each combination of the following three parameters, ten instances were generated:

The parameters were generated in intervals [a, b] with a uniform distribution called U[a, b]:

- production cost  $(vc_{ij})$  U[1.5,2.5];
- setup cost  $(sc_{ij})$  U[5,95];
- inventory cost  $(hc_i)$  U[0.2,0.4];
- production time  $(vt_{ij})$  U[1,5];
- setup time  $(st_{ij})$  U[10,50];

• demand  $(d_{it})$  U[0,180];

To generate instances with high setup costs, the setup costs were multiplied by 10. In the same way, to generate instances with high setup times the setup times were multiplied by 1.5.

To generate the normal capacity (NC), Toledo and Armentano (2006) use as a base a lot-for-lot policy and afterwards, an adjustment is made to reduce the capacity in order to generate instances which use about 80% of the capacity. The tight capacity (TC) is obtained by multiplying this capacity by 0.9. Further details can be found in the paper Toledo and Armentano (2006).

We compare the hybrid methods described in Section 5 (LR/EF and LR/EF/CF) with other existing methods: 1) the Lagrangian heuristic proposed in Toledo and Armentano (2006) (per item decomposition), here denoted by TA, 2) the Lagrangian heuristic applied to the compact formulation (6)-(12) as proposed in Fiorotto and de Araujo (2014) (using the period and machine decomposition) denoted by LR/CF and 3) the CPLEX 12.6 software package, applied to the formulation (6)-(12). The tests were done on a personal computer Intel Core-I5, 2.27GHz with 6Gb of RAM and the Windows 7 operating system.

The parameters used in the computational tests for the method LR/EF, described in Section 5.1, which solves the master problem with Lagrangian relaxation are as follows: 900 iterations of the subgradient optimization method; initialization of the dual prices to zero; size of the initial step of the subgradient method equal to 1 which is decreased by multiplying the latest step size by a factor of 0.7 if the Lagrangian bound does not improve in the last 10 iterations. For the method LR/EF/CF, described in Section 5.2, we use the same parameter setting described in the above method to approximate the solution of the master problem with the Lagrangian relaxation. In addition, the columns are generated with Lagrangian relaxation and in this case we use the following settings: 200 iterations of the subgradient optimization method; initialization of the dual prices according to those obtained in the solution of the master problem; size of the initial step equal to 1 which is decreased by multiplying the latest step size by a factor of 0.6 if the Lagrangian bound does not improve in the last iteration. After 5 iterations of the column generation procedure, the number of iterations of the subgradient optimization method is reduced from 200 to 1, this basically means that we just do regular column generation for that point on, without subgradient optimization. For the method LR/CFproposed in Fiorotto and de Araujo (2014): they start the subgradient optimization method fixing the dual variables to zero; the size of the initial step is equal to 1 and decreases by multiplying by 0.6 if the Lagrangian solution is not improved in the last 50 iterations; 2500 iterations are made in total. For the method TA proposed in Toledo and Armentano (2006): the size of the initial step is equal 1.75 and decreased by 2 if the Lagrangian solution is not improved in the last 25 iterations; 150 iterations are made in total.

In the tables in which we compare the gaps, the upper bounds of the two methods proposed in this paper are obtained by the heuristic described in Section 6. Note that the best results are highlighted in bold numbers.

Table 1 shows the overall performance of the several methods (computational times in seconds and the gaps) aggregated over the 8 combinations of the following three factors: capacity, setup cost and setup time. Each row of the table represent thus the average of 270 instances generated for each class. Moreover, for CPLEX we fixed the time limit for each instance to the same time as used by the method LR/EF/CF. The results from TA are taken directly from their paper and the computational experiments were done on AMD Athlon XP2600 with 2.08GHz and 1GB of RAM. We note that despite the higher computation times of the hybrid methods, mainly the method LR/EF/CF, they present better gaps than the Lagrangian heuristics TA and LR/CF proposed in Toledo and Armentano (2006) and Fiorotto and de Araujo (2014), respectively. Note that this difference increases significantly when the problems with high setup cost (HS) are considered. In most cases, we have the same conclusion comparing the results found by the hybrid methods with CPLEX.

	CP	LEX	Г	ТА		LR/CF		$/\mathrm{EF}$	LR/EF/CF	
capacity/setup	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
NCLSLT	1.7	81.8	2.0	3.9	1.5	3.3	1.4	70.1	1.3	81.8
TCLSLT	1.8	71.5	3.4	8.9	2.4	4.1	2.3	35.2	1.9	71.5
NCHSLT	5.8	20.8	11.7	5.3	8.9	4.9	5.8	16.9	4.6	20.8
TCHSLT	7.6	17.3	18.5	7.8	12.6	6.5	6.1	25.7	5.7	17.3
NCLSHT	3.2	68.5	2.5	4.9	1.7	3.5	1.7	76.7	1.5	68.5
TCLSHT	2.8	86.1	4.2	14.4	2.9	4.7	2.6	27.7	2.0	86.1
NCHSHT	6.9	23.2	13.8	5.9	9.9	5.2	5.9	17.3	4.9	23.2
TCHSHT	8.1	17.1	23.5	8.4	13.9	6.6	6.2	20.9	5.7	17.1

Table 1: General average gap and computational times for each class.

Note that for Table 2 and the following tables, we are not able to give the results for TA, since the results were not provided according to this classification in the paper of Toledo and Armentano (2006).

Table 2 shows that both the gaps and the computation times increase with an increasing number of periods. On the other hand, Table 3 shows that while the computations times increase with an increasing number of items, the gaps decrease. This was also observed by Trigeiro et al. (1989) for the single machine lot sizing problem with set up times. Finally we observe from Table 4 that with the decomposition methods, the gaps increase but the computation times decrease with an increasing number of machines. In Tables 2 to 4, the new hybrid method always give average gaps that are much better compared to the other decomposition methods. This is also the case when we compare the results to CPLEX, except for the case with 25 items in Table 3 and the case with 2 machines in Table 4 for which CPLEX

provides slightly better gaps.

	CPLEX		LR/CF		LR/EF		LR/EF/CF	
Periods	Gap	Time	Gap	Time	Gap	Time	Gap	Time
6	4.2	11.2	4.7	2.8	2.9	7.8	2.4	11.2
12	4.3	39.3	6.6	6.7	4.2	29.1	3.8	39.3
18	5.7	93.9	8.9	13.5	4.9	72.1	4.2	93.9

Table 2: General average gap and computational times aggregated per period

	CP	LEX	LR	$/\mathrm{CF}$	LR/EF		LR/F	EF/CF
Items	Gap	Time	Gap	Time	Gap	Time	Gap	Time
6	9.1	6.9	11.3	2.9	7.1	5.0	6.4	6.9
12	3.9	23.9	6.4	6.0	3.4	15.9	2.7	23.9
25	1.2	113.2	2.5	14.1	1.5	88.1	1.3	113.2

Table 3: General average gap and computational times aggregated per item.

	CP	LEX	LR	$/\mathrm{CF}$	LR/EF		LR/F	EF/CF
Machines	Gap	Time	Gap	Time	Gap	Time	Gap	Time
2	<b>2.4</b>	69.1	4.7	8.2	3.2	57.8	2.6	69.1
4	5.5	41.2	6.9	6.8	3.8	29.7	3.3	41.2
6	6.3	33.88	8.6	4.5	5.0	21.6	4.5	33.8

Table 4: General average gap and computational times aggregated per machine.

In Tables 5 to 8 we set for each class the lower bound found by the linear relaxation to 100%, and calculated the other values relative to this. For example, for the first class in Table 5, the linear relaxation after the branching is 0.3% better than the linear relaxation.

Table 5 presents the lower bounds found by CPLEX for each class 1) from linear relaxation at the root node (Linear Rel. column); 2) after applying the cuts (cuts column); and 3) after branching (branches column). These lower bounds are compared with those generated by all methods that are being analyzed. Note that the proposed hybrid methods found, in all cases, better lower bounds than the TA and LR/CF methods. In relation to CPLEX, even after branching, the lower bounds found using the hybrid methods are better, mainly for the problems with high setup cost. This means that, if the lower bounds generated by these methods were used in the root node of CPLEX, in addition to getting better gaps, it would also helps the solver to prune more in the solution tree and find better feasible solutions.

In relation to the two proposed methods, it is observed that the differences are small, however, the method LR/EF/FC obtained a small advantage in all instances.

Analyzing the lower bound results separately per period, item and machine (Tables 6, 7 and 8), we conclude that the hybrid methods found on average better lower bounds in all configurations. The difference is particularly pronounced for the instances with a small number of items or a high number of machines. On the other hand, the difference in lower bounds

	CI	PLEX (LE	3)	TA	LR/CF	LR/EF	LR/EF/FC
capacity/setup	Linear Rel.	Cuts	Branching	LB	LB	LB	LB
NCLSLT	100	100.27	100.34	99.94	100.34	100.67	100.84
TCLSLT	100	100.42	100.54	99.78	100.46	101.29	101.30
NCHSLT	100	101.29	101.55	99.73	100.79	104.12	104.33
TCHSLT	100	101.39	101.81	99.34	101.87	105.66	105.81
NCLSHT	100	100.26	100.35	99.93	100.39	100.90	100.96
TCLSHT	100	100.38	100.52	99.70	100.54	102.05	101.19
NCHSHT	100	101.43	101.73	99.54	100.82	104.99	105.15
TCHSHT	100	101.42	102.13	98.44	102.76	106.46	107.13

Table 5: General average lower bounds for each class.

is small for the instances with a large number of items or a small number of machines. The number of periods has a less pronounced impact on the lower bounds, even though here we observe again that the hybrid methods provide much better lower bounds.

	CI	PLEX (LE	3)	ТА	LR/CF	LR/EF	LR/EF/FC
Periods	Linear Rel.	Cuts	Branching	LB	LB	LB	LB
6	100	101.12	101.50	99.55	100.93	102.58	102.78
12	100	100.75	100.87	99.45	100.89	103.16	103.36
18	100	100.58	100.62	99.65	101.15	103.69	103.89

Table 6: General average lower bounds aggregated per period.

	CI	PLEX (LE	3)	ТА	LR/CF	LR/EF	LR/EF/FC
Items	Linear Rel.	Cuts	Branching	LB	LB	LB	LB
6	100	102.09	102.68	98.91	102.04	106.29	106.63
12	100	100.40	100.54	99.81	100.73	102.66	102.83
25	100	100.08	100.14	99.93	100.20	100.48	100.56

	CI	PLEX (LE	3)	ТА	LR/CF	LR/EF	LR/EF/FC
Machines	Linear Rel.	Cuts	Branching	LB	LB	LB	LB
2	100	100.47	101.05	99.45	100.77	101.70	101.88
4	100	100.71	100.82	99.62	100.99	103.68	103.81
6	100	100.95	100.98	99.61	101.95	103.95	104.09

Table 8: General average lower bounds aggregated per machine.

In the Tables 9 to 12 we compare the upper bounds. We fixed the upper bound found by CPLEX to 100%, and calculated again the upper bounds found by the other methods relative to this. Table 9 shows the heuristic's behavior (upper bounds) for all classes. The results show that we improved the heuristic proposed by Fiorotto and de Araujo (2014) considering that we found better upper bounds for all classes. Note that these improvements are bigger for the classes with high setup cost. Note that for 6 out of the 8 classes, we also improved on average CPLEX upper bounds, whereas for the two remaining classes the increase is only 0.45% and

0.20%. The high percentage of feasible solutions (% FS) found by the hybrid methods show the efficiency of these heuristic. CPLEX, however, did not find feasible solutions for a considerable number of instances within the time limit imposed. Instances with high setup costs seem specifically difficult for CPLEX.

	CP	LEX	LR,	/CF	LR/E	F/CF	LR,	/EF
capacity/setup	UB	%FS	UB	%FS	UB	%FS	UB	%FS
NCLSLT	100	92.22	100.79	100.00	100.20	100.00	99.99	100.00
TCLSLT	100	92.96	101.52	100.00	100.72	100.00	100.45	100.00
NCHSLT	100	82.96	101.97	100.00	100.29	100.00	99.33	100.00
TCHSLT	100	69.25	99.23	99.62	97.04	100.00	96.11	99.62
NCLSHT	100	95.92	99.15	100.00	98.72	99.25	98.37	100.00
TCLSHT	100	86.29	99.66	100.00	98.86	98.88	98.60	100.00
NCHSHT	100	70.00	101.84	100.00	100.23	100.00	99.11	100.00
TCHSHT	100	80.00	104.34	99.25	101.25	99.62	100.20	99.62

Table 9: General average upper bounds for each class.

Finally, Tables 10, 11 and 12 compare the upper bounds considering the number of periods, items and machines. We note that the new heuristic is better mainly for problems with 6 periods, 6 items and 4 or 6 machines.

	CP	LEX	LR/CF		LR/EF		LR/EF/CF	
Periods	UB	%FS	UB	%FS	UB	%FS	UB	%FS
6	100	94.16	99.27	100.00	98.65	99.72	97.74	100.00
12	100	76.25	100.97	99.72	99.93	99.86	99.55	99.86
18	100	78.33	102.53	99.72	100.65	99.44	99.98	99.72

Table 10: General average upper bounds aggregated per period.

	CP	LEX	LR/CF		$LR_{\prime}$	$/\mathrm{EF}$	LR/EF/CF	
Items	UB	%FS	UB	%FS	UB	%FS	UB	%FS
6	100	65.27	100.57	99.44	98.22	99.58	97.01	99.44
12	100	87.50	101.46	100.00	100.28	99.44	99.64	100.00
25	100	96.11	101.57	100.00	100.53	100.00	100.43	100.00

Table 11: General average upper bounds aggregated per item.

	CPLEX		LR/CF		LR/EF		LR/EF/CF	
Machines	UB	%FS	UB	%FS	UB	%FS	UB	%FS
2	100	97.91	101.66	99.46	100.79	99.30	100.51	99.44
4	100	80.83	100.97	100.00	99.52	100.00	98.80	100.00
6	100	70.00	100.25	100.00	98.03	100.00	97.26	100.00

Table 12: General average upper bounds aggregated per machine.

# 8 Conclusion

In this paper, the lot sizing problem with capacity constraints and parallel machines was studied. A reformulation of the problem using the variable redefinition approach proposed by Eppen and Martin (1987) was used. Based on the literature, two hybrid solution methods that combine Lagrangian relaxation and Dantzig-Wolfe decomposition were extended to the problem considered. In both methods, the Dantzig-Wolfe decomposition is applied using as linking constraints the flow constraints. The problem is hence decomposed per period and per machine instead of the classical per item decomposition. A feasibility heuristic based on production transfers in order to satisfy the demand constraints is developed. This strategy was compared to two Lagrangian heuristics proposed in the literature and also compared to CPLEX 12.6. The computational results show that the proposed hybrid methods are efficient with respect to both the lower bounds and upper bounds compared the existing methods.

### Acknowledgements

This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (Process numbers 2010/16727-9, 2013/00965-6 and 2011/22647-0) and by the Natural Sciences and Engineering Research Council of Canada (grant 342182-09). This support is gratefully acknowledged.

# References

- [Andriolo et al.(2014)] A. Andriolo, D. Battini, A. Persona, F. Sgarbossa, R. W. Grubbström, A Century of Evolution from Harris's Basic Lot Size Model: Survey and Research Agenda. *Int. J. Production Economics*, (2014), In press.
- [de Araujo et al.(2013)] S. A. de Araujo, B. De Reyck, Z. Degraeve, I. Fragkos, R. Jans, Period Decompositions for the Capacity Constrained Lot Size Problem with Setup Times. Les Cahiers du GERAD, G-2013-76 (2013), 1–23.
- [Bahl(1983)] H. C. Bahl, Column generation based heuristic algorithm for multi-item scheduling. IIE Trans, 15 (1983) 136–141.
- [Barnhart et al.(1998)] C.E.L. Barnhart, G.L. Johnson, M.W.P. Nemhauser, P. H. Vance, Branch-andprice: Column generation for solving huge integer programs. Operations Research, 46 (1998) 316–329.
- [Belvaux and Wolsey(2000)] G. Belvaux, L. A. Wolsey, Bc-prod: a specialized branch-and- cut system for lot-sizing problems. *Management Science*, **46** (2000), 993–1007.

- [Ben Amor et al.(2007)] H. Ben Amor, J. Desrosiers, J.M. Valèrio de Carvalho, Dual-optimal inequalities for stabilized column generation. Operations Research, 54 (2007), 454–463.
- [Bitran and Matsuo(1986)] G. R. Bitran, H. Matsuo, The multi-item capacitated lot size problem: Error bounds of mannes formulations. *Management Science*, **32** (1986), 350–359.
- [Bixby et al.(1992)] R. E. Bixby, J. W. Gregory, I. J. Lustin, R. E. Marsten, D. F. Shanno, Very large-scale linear programming: a case study in combining interior point and simplex methods. *Operations Research*, 40 (1992) 885–897.
- [Camerini et al.(1975)] P. M. Camerini, L. Fratta, F. Maffioli, On Improving Relaxation Methods by Modified Gradient Techniques. *Mathematical Programming Study*, 3 (1975), 26–54.
- [Carreno(1990)] J. J. Carreno, Economic lot scheduling for multiple products on parallel identical processors. *Management Science*, **36** (1990), 348–358.
- [Cattrysse et al.(1990)] D. Cattrysse, J. Maes, L. N. van Wassenhove, Set partitioning and column generation heuristics for capacitated dynamic lotsizing. *European Journal of Op*erational Research, 46 (1990) 38–47.
- [Cattrysse et al.(1993)] D. G. Cattrysse, M. Salomon, R. Kuik, V. L. N. Wassenhove, A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. *Management Science*, **39** (1993), 447–486.
- [Dantzig and Wolfe(1960)] G. B. Dantzig, P. Wolfe, Decomposition principles for linear programming. Operations Research, 8 (1960), 101–111.
- [Degraeve and Jans(2007)] Z. Degraeve, R. Jans, A new dantzig-wolfe reformulation and branch-and-price algorithm for the capacited lot-sizing problem with setup times. Operations Research, 55 (2007), 909–920.
- [De Matta and Guignard(1994)] R. De Matta, M. Guignard, Dynamic production scheduling for a process industry. Operations Research, 42 (1994), 492–503.
- [De Matta and Guignard(1995)] R. De Matta, M. Guignard, The performance of rolling production schedules in a process industry. *IIE Transactions*, 27 (1995), 564–573.
- [Dzielinski and Gomory(1965)] B. P. Dzielinski, R. E. Gomory, Optimal programming of lot sizes inventory and labor allocations. *Management Science*, **11** (1965) 874–890.
- [Eppen and Martin(1987)] G. B. Eppen, R. K. Martin, Solving Multi-Item Capacitated Lot-Sizing Problems Using Variable Redefinition. Operations Research, 6 (1987), 832–848.

- [Fandel and Stammen-Hegener(2006)] G. Fandel, C. Stammen-Hegener, Simultaneous lot sizing and scheduling for multi-product multi-level production, *International Journal of Production Economics.* **104** (2006), 308–316.
- [Fiorotto and de Araujo(2014)] D. J. Fiorotto, S. A. de Araujo, Reformulation and a Lagrangian Heuristic for Lot Sizing Problem on Parallel Machines. Annals of Operations Research, (2014), In press.
- [Gondzio et al.(2013)] J. Gondzio, P. G. Brevis, P. Munari, New developments in the primal dual column generation technique. *European Journal of Operational Research*, 224 (2013), 41–51.
- [Haase(2005)] K. Haase, Solving large-scale capacitated lot-sizing problems close to optimality. Working paper, Technische Universität Dresden, (2005).
- [Helber and Sahling(2010)] S. Helber, F. Sahling, A fix-and-optimize approach for the multilevel capacitated lot sizing problem. *Journal Production Economics*, **123** (2010), 247–256.
- [Hindi(1995)] K. S. Hindi, Computationally efficient solution of the multi-item, capacitated lot-sizing problem. *Comput Ind Eng*, 28 (1995), 709–719.
- [Hindi(1996)] K. S. Hindi, Solving the CLSP by a tabu search heuristic. European Journal of Operational Research. 47 (1996) 151–161.
- [Huisman et al.(2005)] D. Huisman, R. Jans, M. Peeters, A. P. M. Wagelmans, Combining column generation and lagrangian relexation. G. Desaulniers, J. Desrosiers, M. Solomon, eds. Column Generation. Springer, New York, (2005), 247–270.
- [Jans(2009)] R. Jans, Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints. INFORMS Journal on Computing, 21 (2009), 123–136.
- [Jans and Degraeve(2004a)] R. Jans, Z. Degraeve, Improved lower bounds for capacitated lot sizing problem with setup time. Operation Research Letters, 32 (2004a), 185–195.
- [Jans and Degraeve(2004b)] R. Jans, Z. Degraeve, An industrial extension of the discrete lot sizing and scheduling problem. *IIE Transactions*, **36** (2004b), 47–58.
- [Kang et al.(1999)] S. Kang, K. Malik, L. J. Thomas, Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Science*, 45 (1999), 273– 289.

- [Lambrecht and Vanderveken(1979)] M. Lambrecht, H. Vanderveken, Heuristic procedures for the single operation, multi-item loading problem. AIIE Transactions, 11 (1979), 319–326.
- [Lasdon and Terjung(1971)] L. S. Lasdon, R. C. Terjung, An efficient algorithm for multi-item scheduling. Operations Research, 19 (1971) 946–969.
- [Manne(1958)] A. S. Manne, Programming of economic lot sizes. *Management Science*. 4 (1958), 115–135.
- [Marinelli(2007)] F. Marinelli, M. E. Nenni, A. Sforza, Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. Annals of Operations Research, 150, (2007), 177–192.
- [Mergaux and van Wassenhove(1984)] L, P. Mergaux, L. N. van Wassenhove, Production Planning With Capacity Constraints, Master thesis, Division of Industrial Management, Katholieke Universiteit Leuven, Dutch, 1984.
- [Meyr(2002)] H. Meyr, Simultaneous lotsizing and scheduling on parallel machines. European Journal of Operational Research. 139 (2002), 277–292.
- [Meyr and Mann(2013)] H. Meyr, M. Mann, A decomposition approach for the general lotszing and scheduling problem for parallel production lines. *European Journal of Operational Research.* 229 (2013), 718–731.
- [Ozdamar and Birbil(1998)] L. Ozdamar, S. I. Birbil, Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, **110** (1998), 525–547.
- [Pimentel et al.(2010)] C. M. O. Pimentel, F. P. Alvelos, J. M. V. Carvalho, Comparing Dantzig-Wolfe decompositions and branch-and-price algorithms for the multi-item capacitated lot sizing problem. Optimization Methods and Software, 25 (2010), 229–319.
- [Salomon et al.(1991)] M. Salomon, L. G. Kroon, R. Kuik, L. N. van Wassenhove, Some extensions of the discrete lot-sizing and scheduling problem. *Management Science*, **37** (1991), 801–812.
- [Salomon et al.(1993)] M. Salomon, R. Kuik, L. N. van Wassenhove, Statistical search methods for lotsizing problems. Operations Research, 41 (1993), 453–468.
- [Stadtler(2003)] H. Stadtler, Multilevel lot sizing with set up times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. Operations Research, 51 (2003), 487–502.

- [Tempelmeier and Buschkuhl(2009)] H. Tempelmeier, L. Buschkuhl, A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures. Or Spectrum, **31** (2009), 385–404.
- [Toledo and Armentano(2006)] F. M. B. Toledo, V. A. Armentano, A Lagrangian-Based Heuristic for the Capacitated Lot-Sizing Problem in Parallel Machines. *European Journal* of Operational Research, **175** (2006), 1070–1083.
- [Trigeiro et al.(1989)] W. W. Trigeiro, J. Thomas, J. O. McClain, Capacitated lot sizing with setup times. *Management Science*, **35** (1989), 353–366.
- [Wagner and Whitin(1958)] H. M. Wagner, T. M. Whitin, Dynamic version of the economic lotsize model. *Management Science*, 5 (1958), 89–96.