# MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs

**Nabil Absi**[1,2*], **Safia Kedad-Sidhoum**[1†]

[1]Laboratoire LIP6, 4 place Jussieu, 75 252 Paris Cedex 05, France

[2]Dynasys S.A., 10 Avenue Pierre Mendes France, 67 300 Schiltigheim, France

**Abstract**

We address a multi-item capacitated lot-sizing problem with setup times that arises in real-world production planning contexts. Demand cannot be backlogged, but can be totally or partially lost. Safety stock is an objective to reach rather than an industrial constraint to respect. The problem is NP-hard. We propose mixed integer programming heuristics based on a planning horizon decomposition strategy to find a feasible solution. The planning horizon is partitioned into several sub-horizons over which a freezing or a relaxation strategy is applied. Some experimental results showing the effectiveness of the approach on real-world instances are presented. A sensitivity analysis on the parameters of the heuristics is reported.

**Keywords :** Lot-sizing, setup time, shortage cost, safety stock deficit, production planning, mixed integer programming, heuristic.

## Introduction

The production planning problems encountered in the industry are generally intractable due to several practical constraints. In this context, the decision maker has to find a good feasible solution in a reasonable execution time rather than an optimal one. The problem under consideration arises from an industrial context. In this way, the resolution approach proposed in this paper is motivated by developing a tool that helps in finding a good feasible solution for the multi-item capacitated lot-sizing problem with setup times (MCLSP) in which there is a time-varying demand for a set of $N$ items denoted $\mathcal{I} = \{1, 2, \cdots, N\}$ over $T$ periods. The production should satisfy a restricted capacity and must take into account a set of additional constraints.

[*] This work has been partially financed by DYNASYS S.A., under research contract no. 588/2002.

[†] Corresponding author. E-mail: Safia.Kedad-Sidhoum@lip6.fr

The production of an item requires one or more resources per period. This can be induced by the length of the time bucket or the need to use several parallel resources, typically, material and human. We denote by $R$ the number of available resources.

Launching the production of an item at a given period involves a time-varying capacity and a fixed consumption of resource usually called setup time in lot-sizing literature. For each period, an inventory cost is attached to each item as well as a variable unit production cost and a setup cost.

In practice, it often happens that no backlogging is allowed. Indeed, the customers do not allow any delay on the deliveries. In this case, the orders are cancelled and we talk about a partial or a total loss of requirement, this is called a shortage on the demand. The most important cause of shortages is the capacity resource limitations.

The MCLSP problem has the distinctive feature of allowing requirement shortages because we deal with problems with tight capacities. The objective here is to evaluate the volume of the demand that could meet the available capacity rather than to estimate the resources capacity. Indeed, when we are in lack of capacity to produce the total demand, we try to spread the available capacity among the items by minimizing the total amount of demand shortages. Thus, we introduce in the model a unit cost parameter for each item at each period for the requirement not met regarding the demand. These costs should be viewed as penalty costs and their values are very high in comparison with other cost components.

The use of safety stock is widely prevalent in industry to counter variability that may be present in a supply chain environment. In our study, a safety stock is an objective or a target to be reached rather than an industrial constraint to respect. It can happen that we cannot reach this safety stock; in this case we talk about safety stock deficits for which we introduce a unitary cost parameter for each item at each period. These costs should also be viewed as penalty costs and their values are lower than shortage costs and very high in comparison with other cost components.

An item could have two or more setup times. The first one depends only on the item, and the second one is shared by several items. In this case, items that have the same setup time are gathered in a same group. An item can belong to one or more groups. Thus, a group can contain one or more items. Launching the production of an item at a given period involves a setup time and a setup cost for the group to which it belongs. We denote by $J$ the number of groups.

Moreover, production planning allows not only determining the quantities to produce to meet the demands, but also to choose between several manufacturing processes. Indeed, it often happens that alternative modes of production are available since several production lines or operating sets are used such as salesmen or subcontractors. Each alternative has specific production cost and resource consumption parameters. We denote by $V$ the number of operating sets.

In addition, for economic reasons and in order not to make small crop years, the decision maker can be brought to fix a threshold of production (minimum production) which it should be exceeded when the manufacture of an item is launched. This model is often used when we have big setup costs or times.

In this context, the MCLSP problem consists in finding a production planning that minimizes the demand shortages, the safety stock deficits as well as the setup, the inventory and the production costs.

Production planning models involving multiple items, restrictive capacities and significant setup times have often been studied in the literature. Obtaining optimal and sometimes even feasible solutions remains challenging. Trigeiro *et al.* [26] were among the firsts to try to solve such models. They proposed a lagrangean relaxation based heuristic to solve the single-machine, multi-item, capacitated lot-sizing problem with setup times to obtain near-optimal solutions.

Miller *et al.* [20], Belvaux and Wolsey [4], Leung *et al.* [14] and Pochet and Wolsey [21] proposed exact methods to solve multi-item capacitated lot-sizing problems by strengthening the LP formulations with valid inequalities and then using a mixed integer programming (MIP) solver.

Different approaches were studied in the literature to solve the classical multi-item lot-sizing problem with setup times ([22], [9]). But if all the industrial features of the problem previously described are taken into account, such approaches are difficult to generalize and to implement.

In this paper, we present a hybrid method based on a MIP formulation of the problem and a decomposition strategy of the planning horizon in order to provide a feasible solution to the MCLSP problem and try to meet the maximum amount of client's needs and to reach the safety stock levels. To the best of our knowledge, the problem we consider has never been tackled before in the literature. For more details about the characteristics of the MCLSP problem, the reader can refer to Absi [1].

Florian *et al.* [11] and Bitran and Yanasse [5] have shown that the single-item capacitated lot-sizing problem is NP-hard, even for many special cases. Chen and Thizy [7] have proved that multi-item capacitated lot-sizing problem with setup times is strongly NP-hard.

There are few references dealing with lot-sizing problems with shortage costs or safety stocks. Recently, Sandbothe and Thompson [23] addressed a single-item lot-sizing problem with constant capacity and shortage costs. The authors proposed an $O(T^3)$ forward dynamic programming algorithm to solve the problem. Aksen *et al.* [3] proposed a dynamic programming method to solve the same problem without capacity constraints in $O(T^2)$. Loparic *et al.* [16] proposed valid inequalities for the single-item uncapacitated lot-sizing problem with sales instead of fixed demands and lower bounds on stock variables. Other authors used specific MIP-based heuristics to solve lot-sizing problems. We can quote, Clark [8], Kelly [12], Mercé and Fontan [18], Pochet and Van Vyve [27], Stadtler [24] and Suerie and Stadtler [25].

The main contributions of this paper are twofold. First, we propose a MIP formulation for the MCLSP problem. Secondly, we develop MIP-based heuristics to solve this problem. These methods are integrated in an Advanced Planning and Scheduling (APS) software. An outline of the remainder of the paper is as follows. Section 1 describes a MIP formulation of the MCLSP problem. In Section 2 we illustrate the planning horizon decomposition strategy. In Section 3 we develop two MIP-based heuristics using this planning horizon decomposition. Finally, computational results and sensitivity analysis are reported in Section 4 to show the effectiveness of these MIP-based heuristics.

# 1 Formulation of the MCLSP problem

In this section, we present a MIP formulation of the MCLSP problem, which is an extension of the classical formulation previously studied by Miller [19] and Trigeiro *et al.* [26] for some special cases of the MCLSP problem. This formulation is usually called aggregated model, see [6]. Other formulations are studied in the literature. We can mention the facility location-based formulation introduced by Krarup and Bilde [13] and the shortest path formulation proposed by Evans [10].

In the sequel of the paper, we denote by $i = 1, \ldots, N$, $t = 1, \ldots, T$, $j = 1, \ldots, J$, $r = 1, \ldots, R$ and $v = 1, \ldots, V$ respectively for the index of an item, a period, a group, a resource and an operating set. We set $x_{vit}$ as the quantity of item $i$ produced at period $t$ using operating set $v$. To deal with the fixed setup times and costs for an item, we need also to define $y_{vit}$ as a binary variable equal to 1 if item $i$ is produced at period $t$ using operating set $v$ (i.e. if $x_{vit} > 0$). We need also to define a binary variable $z_{vjt}$ to deal with the group setup times and costs. This variable is equal to 1 if at least one item $i$ belonging to group $j$ is launched on operating set $v$ at period $t$. The variable $s_{it}$ is the inventory value for item $i$ at the end of period $t$. To try to meet the demand for an item $i$ at period $t$, we could anticipate the production over some periods of time. Therefore, $\sigma_{it}$ denotes the last period at which an item $i$ produced at period $t$ can be consumed.

The demand shortage for item $i$ at period $t$ is modelled by a non-negative variable $r_{it}$ with a very high unit penalty cost in the objective function. Indeed, the aim of the model considered is to meet the customer's demand and thus to have the minimum amount of requirements not met.

Let $I_{it}^{+}$ and $I_{it}^{-}$ represent respectively overstock variable and safety stock deficit variable of item $i$ at period $t$. $I_{it}^{-}$ has a high unit penalty cost in the objective function. However, this penalty is lower than demand shortage cost. We set $l_{it}$ as a parameter which represents the safety stock of item $i$ at period $t$. Thus, the variable $s_{it}$ can be replaced by $I_{it}^{+} + l_{it} - I_{it}^{-}$. In addition, the following notations are used in order to describe the mathematical model of the MCLSP problem:

$d_{it}$: demand for item $i$ at period $t$.

$c_{rt}$: available capacity of resource $r$ in period $t$.

$f_{virt}$: setup time for item $i$, using operating set $v$, on resource $r$, at period $t$.

$v_{virt}$: variable resource consumption for item $i$, using operating set $v$, on resource $r$, at period $t$.

$g_{vjrt}$: setup time for group $j$, using operating set $v$, on resource $r$, at period $t$.

$$a_{ij} = \begin{cases} 1 \text{ if item } i \text{ belongs to group } j. \\ 0 \text{ otherwise} \end{cases}$$

$p_{vit}^{\min}$: minimum production level for item $i$, using operating set $v$, on resource $r$, at period $t$.

$\alpha_{vit}$: production unit cost for item $i$, using operating set $v$, on resource $r$, at period $t$.

$\beta_{vit}$: setup cost for item $i$, using operating set $v$, on resource $r$, at period $t$.

$\omega_{vjt}$: setup cost for group $j$, using operating set $v$, on resource $r$, at period $t$.

$\gamma^+_{it}$: inventory unit cost for item $i$ at period $t$.

$\gamma^-_{it}$: safety stock deficit unit cost for item $i$ at period $t$.

$\varphi_{it}$: shortage unit cost on the demand of item $i$ at period $t$.

We denote: $\delta_{it} = l_{it} - l_{i(t-1)}$.

Using these variables and parameters, we formulate the MCLSP problem as follows:

$$\min \sum_{v,i,t} \alpha_{vit} x_{vit} + \sum_{v,i,t} \beta_{vit} y_{vit} + \sum_{v,j,t} \omega_{jt} z_{vjt} + \sum_{i,t} \varphi_{it} r_{it} + \sum_{i,t} \gamma^+_{it} I^+_{it} + \sum_{i,t} \gamma^-_{it} I^-_{it} \tag{1}$$

$$I^+_{i(t-1)} - I^-_{i(t-1)} + r_{it} + \textstyle\sum_{v=1}^{V} x_{vit} = d_{it} + \delta_{it} + I^+_{it} - I^-_{it}, \qquad \forall i, \forall t \tag{2}$$

$$\textstyle\sum_{v=1}^{V} \sum_{i=1}^{N} (v_{virt} x_{vit} + f_{virt} y_{vit}) + \sum_{v=1}^{V} \sum_{j=1}^{J} g_{vjrt} z_{vjt} \leq c_{rt}, \qquad \forall r, \forall t \tag{3}$$

$$x_{vit} \leq M y_{vit}, \qquad \forall v, \forall i, \forall t \tag{4}$$

$$x_{vit} \geq p^{\min}_{vit} y_{vit}, \qquad \forall v, \forall i, \forall t \tag{5}$$

$$y_{vit} \geq a_{ij} z_{vjt}, \qquad \forall v, \forall i, \forall j, \forall t \tag{6}$$

$$r_{it} \leq d_{it}, \qquad \forall i, \forall t \tag{7}$$

$$I^-_{it} \leq l_{it}, \qquad \forall i, \forall t \tag{8}$$

$$r_{it}, I^+_{it}, I^-_{it} \geq 0, \qquad \forall i, \forall t \tag{9}$$

$$x_{vit} \geq 0, \qquad \forall v, \forall i, \forall t \tag{10}$$

$$y_{vit} \in \{0, 1\}, \qquad \forall v, \forall i, \forall t \tag{11}$$

$$z_{vjt} \in \{0, 1\}, \qquad \forall v, \forall j, \forall t \tag{12}$$

The objective function (1) minimizes the total cost induced by the production plan (production costs, inventory costs, shortage costs, safety stock deficit costs and setup costs). Constraints (2) are the inventory flow conservation equations through the planning horizon. Constraints (3) are the capacity constraints, the overall consumption must remain lower than the available capacity. Constraints (4) and constraints (5) impose that, if we produce an item then the production must not exceed a maximum production level $M$ and must go over a minimum production. $M$ could be set to the minimum between the total requirement on section $[t, \ldots, \sigma_{it}]$ of the horizon and the highest quantity that we can produce for item $i$ regarding the available capacity. Thus, $M = \min \left\{ \sum_{t'=t}^{\sigma_{it}} d_{it'}; \min_{r=1,\ldots,R} \left\{ (c_{rt} - f_{virt} - \sum_{j=1}^{J} a_{ij} g_{vjrt}) / v_{virt} \right\} \right\}$. Constraints (6) illustrate that the setup time and the setup cost for a group $j$ are taken into account if at least one item $i$ belonging to group $j$ is launched on operating set $v$ at period $t$. Constraints (7) and (8) define upper bounds on respectively the requirement not met and the safety stock deficit for item $i$ on period $t$. Constraints (9), (10), (11) and (12) characterize the variable's domain: $r_{it}, I^+_{it}$ and $I^-_{it}$ are non-negative for $i = 1, \ldots, N$ and $t = 1, \ldots, T$, $x_{vit}$ is non-negative for $i = 1, \ldots, N$, $t = 1, \ldots, T$ and $v = 1, \ldots, V$. $y_{vit}$ is a binary variable for $i = 1, \ldots, N$, $t = 1, \ldots, T$ and $v = 1, \ldots, V$, and $z_{vjt}$ is a binary variable for $j = 1, \ldots, J$, $t = 1, \ldots, T$ and $v = 1, \ldots, V$. The MCLSP problem has $VT(N + J)$ binary variables and $(3N + R + 2VN + VJN)T$ constraints.

Finding a good feasible solution to the monolithic model presented at Section 1 can be extremely hard, especially if the volume of information treated is very significant. The use of a branch-and-bound method to solve such problems is current in industry that uses a generic tool such as APS softwares, but the CPU time remains exorbitant. A way to deal with such problems is to work on models with smaller size, for which we can find near-optimal solutions. To key idea of the heuristics presented in the sequel of the paper is thus to decompose the horizon in several sub-sections as described in what follows.

## 2 Horizon decomposition

The approach that we propose is an iterative method; it solves heuristically at each stage a reduced mixed integer problem. To obtain such a mixed linear problem, we reduce the number of binary variables and the number of constraints. To carry out such a reduction, we use an approach based on planning horizon decomposition.

The principle of the horizon decomposition lies on maintaining the integrity constraints of the binary variables belonging to a fixed section of the horizon, and relaxing these constraints on the other ones. By fixing all the setup variables $y_{vit}$ and $z_{vjt}$ as well as the production variables $x_{vit}$ at period $t$, we eliminate constraints (3), (4), (5), (10), (11) and (12) for this period. Such manipulation reduces considerably the CPU time to solve the sub-problems. Indeed, the search tree is limited to a reduced number of binary variables, and problem size decreases since the number of constraints and variables is restricted.

Our approach consists in a hybridization of a branch-and-bound method and a strategy of horizon decomposition. At each stage of the heuristic, the planning horizon is broken up into three parts (see Figure 1): a frozen window, a decision window and an approximation window.

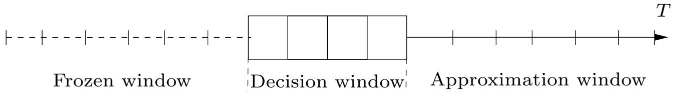

Figure 1: Horizon decomposition

These sections of the planning horizon are described in what follows:

**Frozen window:** In this part all the setup variables $y_{vit}$ and $z_{vjt}$ are fixed by former resolutions within the framework of an iterative process. The production variables $x_{vit}$ can also be frozen. More precisely, we can either freeze the setup variables or freeze both setup and production variables.

**Decision window:** In this sub-section the integrity constraints on setup variables $y_{vit}$ and $z_{vjt}$ are maintained. No modification is made to the mathematical model of the MCLSP problem reduced to this window.

**Approximation window:** In the approximation window the integrity constraints on setup variables $y_{vit}$ and $z_{vjt}$ are relaxed. Thus, the production plan provided by this section of the horizon is not feasible. It gives a lower bound on the capacity needed to satisfy the demand of this section of horizon.

The idea of the approximation window is motivated by industrial observations. Indeed, requirements at the beginning of planning horizon represent the costumer's orders on small time-buckets, e.g. day or week; the decisions taken at these periods are work orders which will be used as an entry for a shop scheduling subsequent resolution. Contrarilly, the requirements at the end of the horizon are the forecasts on big time-bucket, e.g. month or quarter, they can change when a new production planning is carried out after a rolling horizon step. The impact of the integrity constraints relaxation on setup variables at the end of the horizon is thus not as significant as the relaxation of these constraints at the beginning of the horizon.

Based on this decomposition approach, we propose in the following section two MIP-based heuristics, the Fix-and-Relax and the Double-Fix-and-Relax heuristics. We also present an alternative to these heuristics which differ on the way production variables are frozen.

# 3 MIP-based heuristics

In this section, we present two MIP-based heuristics to find a feasible solution to the model presented at Section 1 by solving smaller problems, from combinatorial and size point of view. This is carried out by using the horizon decomposition approach presented in Section 2. The production planning is fixed according to a shifting forward strategy of the decision window through the horizon. Figure 2 describes this principle between two successive stages $k$ and $k+1$ of the heuristics. The main difference between the heuristics considered in this paper lies in the way of freezing setup and production variables. In what follows, we present the principle of the Fix-and-Relax and Double-Fix-and-Relax heuristics.

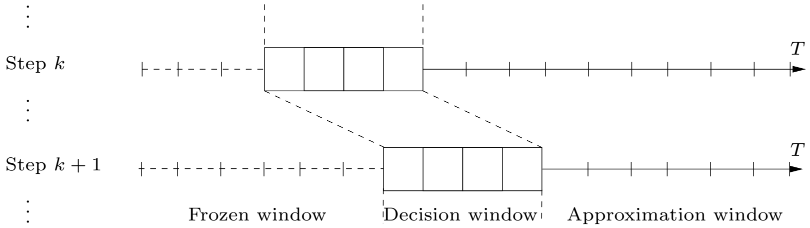

Figure 2: Rolling Horizon

## 3.1 Fix-and-Relax heuristic

In a first step, we start by by fixing a decision window at the beginning of the horizon and an approximation window on the remainder of the horizon. Thus, some of the MCLSP problem constraints are relaxed as described in Section 2 and the problem is solved using a branch-and-bound algorithm. A second step consists then in shifting forward the decision window while keeping an overlapping section with the precedent decision window. This principle is reiterated. Therefore, at step $k$, setup variables related to the section that precedes the decision window are frozen according to the solution obtained at step $k-1$. The MCLSP problem is then solved by a branch-and-bound algorithm at step $k$. This procedure is repeated until the end of the planning horizon is reached. The solution provided by this procedure is feasible. Indeed, all the integrity constraints are satisfied at the end of the procedure.

The Fix-and-Relax heuristic is controlled by four parameters, two of them are relating to the decomposition of the planning horizon and the two others are connected with the stopping criteria of the branch-and-bound method. They are described as follows:

- $\sigma_k$: Size of the decision window at stage $k$ of the algorithm,
- $\delta_k$: Number of overlapping periods between the decision windows of stages $k-1$ and $k$, with $\sigma_k > \delta_k > 0$,
- $Opt_k$: Minimum optimality percent to reach at stage $k$ of the algorithm,
- $Time_k$: Time-Limit at stage $k$ of the algorithm.

In addition, we denote by $P^k_{a_k b_k}$, the MCLSP problem solved at step $k$ of the heuristic for $k \geq 1$. $[a_k, b_k]$ is the decision window of step $k$. By convention, $[1, 0] = \emptyset$. We recall the following properties:

- Setup variables $y_{vit}$ and $z_{vjt}$ are frozen in section $[1, a_k - 1]$ according to the solution obtained at stage $k-1$ of the heuristic,
- Integrity constraints on the setup variables $y_{vit}$ and $z_{vjt}$ are maintained in the decision window $[a_k, b_k]$,
- Integrity constraints on the setup variables $y_{vit}$ and $z_{vjt}$ are relaxed in the approximation window $[b_k + 1, T]$.

$K$ denotes the total number of steps for the Fix-and-Relax method. If $\sigma_k$ and $\delta_k$ are constant for all $k$, that is $\sigma_k = \sigma$ and $\delta_k = \delta$, then $K = \lceil (T - \sigma)/(\sigma - \delta) \rceil + 1$. We can notice that $K$ is bounded by $T$ in the worst case.

The principle of the Fix-and-Relax method is presented in figure 3.

We can notice that the overlapping periods between the decision windows of two successive stages $k$ and $k+1$ of the algorithm permits at stage $k$ to re-open the decisions taken at stage $k$. This questioning makes possible the construction of a better production plan since the approximation window gives only a default estimation of the capacity needed to meet the demand. An undervaluation of the resource capacity can cause shortages on the demands. However, by adding these overlapping periods, an additional anticipation can be made on the total demand. We will show through the example 3.1 a situation for which it is interesting to have overlapping periods between the decision windows of iterations $k$ and $k+1$.

We can observe that there is no frozen section at the first step of the algorithm. All the sub-problems $P^k_{a_k b_k}$ are solved using a branch-and-bound algorithm (CPLEX callable library). The Fix-and-Relax heuristic is described by the algorithm 1. The solution obtained at the end of algorithm 1 is a feasible solution for the MCLSP problem.

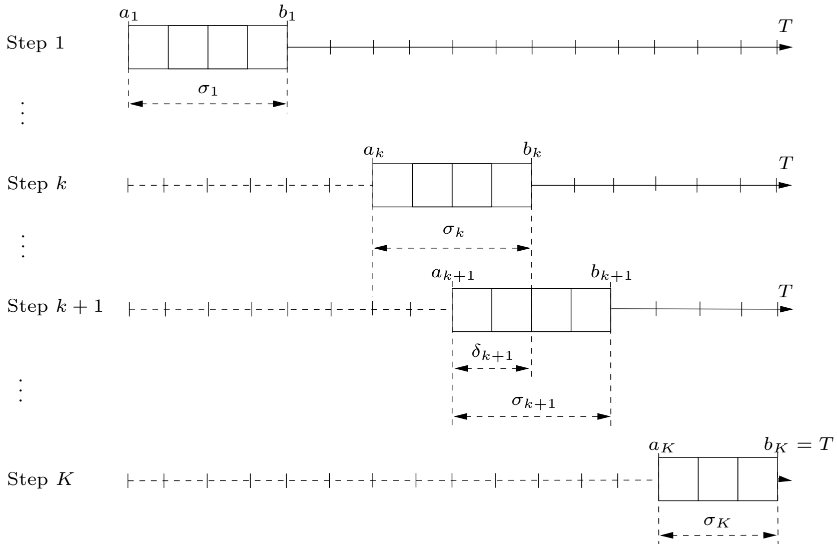

Figure 3: Fix-and-Relax Method

**Algorithm 1** Fix-and-Relax heuristic

```
$k \leftarrow 1$, $a_k \leftarrow 1$, $b_k \leftarrow \sigma_k$
while $b_k < T$ do
    Solve the sub-problem $P^k_{a_k b_k}$
    $k \leftarrow k+1$, $a_k \leftarrow b_{k-1} - \delta_k$, $b_k \leftarrow b_{k-1} + \sigma_k - \delta_k$
    if $b_k > T$ then
        $b_k \leftarrow T$
    end if
end while
Solve the sub-problem $P^k_{a_k b_k}$
```

**Example 3.1.** *We show through this example a situation where it is important to have overlapping periods between the decision windows of two successive stages of the heuristic. We consider the MCLSP problem with the dimensions, the costs and the data respectively defined in tables 1, 2 and 3.*

| $V$ | $N$ | $J$ | $R$ | $T$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 5 |

Table 1: MCLSP dimensions

| $t$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\varphi_{it}$ | 250 | 212, 5 | 175 | 137, 5 | 100 |
| $\beta_{vit}$ | 20000 | 20000 | 20000 | 20000 | 30000 |
| $\alpha_{vit}$ | 10 | 10 | 10 | 10 | 10 |
| $\gamma^+_{it}$ | 1 | 1 | 1 | 1 | 1 |

Table 2: MCLSP costs

| $t$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $d_{1t}$ | 800 | 1000 | 0 | 1000 | 1000 |
| $f_{11t}$ | 500 | 500 | 500 | 500 | 500 |
| $v_{11t}$ | 1 | 1 | 1 | 1 | 1 |
| $c_{1t}$ | 1500 | 1500 | 1500 | 2000 | 2000 |

Table 3: MCLSP Data

*We execute the Fix-and-Relax heuristic twice with $\sigma_k = 3$ for all $k$. The first execution is parameterized with $\delta_k = 0$ for all $k$, while the second one is parameterized with $\delta_k = 1$ for all $k$. Sub-problems are solved optimally using Cplex 9.0. The results are presented in tables 4 and 5.*

| $t$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $x_{11t}$ | 1000 | 1000 | 0 | 1500 | 0 |
| $y_{11t}$ | 1 | 1 | 0 | 1 | 0 |
| $s_{1t}$ | 200 | 200 | 200 | 700 | 0 |
| $r_{1t}$ | 0 | 0 | 0 | 0 | 300 |

Table 4: Fix-and-Relax results with $\delta_k = 0$ for all $k$

| $t$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $x_{11t}$ | 800 | 1000 | 500 | 1500 | 0 |
| $y_{11t}$ | 1 | 1 | 1 | 1 | 0 |
| $s_{1t}$ | 0 | 0 | 500 | 1000 | 0 |
| $r_{1t}$ | 0 | 0 | 0 | 0 | 0 |

Table 5: Fix-and-Relax results with $\delta_k = 1$ for all $k$

*Tables 4 and 5 show that using the first parameter set, we obtain 300 for the shortages at period 5, while this quantity is reduced to zero when using the second parameter set. This improvement is due to the overlapping periods that allow an additional anticipation of the production to meet the demand. Indeed, the production has increased in the third period.*

## 3.2 Double-Fix-and-Relax

The principle of the Double-Fix-and-Relax heuristic is closely related to the Fix-and-Relax method. The main differences are described below. Initially we set an approximation window which size is equal to the planning horizon. The MCLSP problem is thus solved using the simplex method, since all variables are non-negative. In a second step, three windows are considered, a decision window at the beginning of the planning horizon followed successively by an approximation window and a frozen window. For the last one, the setup and production variables are fixed according to the results found at the first step. The MCLSP is then solved using a branch-and-bound algorithm with this new horizon decomposition.

At each of the subsequent stages, a shifting forward operation is applied for the decision and the approximation windows. An overlapping period between the decision window of two successive stages $k-1$ and $k$ is maintained as described previously. The setup variables of the horizon section which precedes the decision window of stage $k$ are frozen according to the solution obtained at stage $k-1$ of the algorithm. Similarly, the MCLSP is then solved using a branch-and-bound algorithm with this new horizon decomposition. This procedure is repeated until the right ending point of the decision window is greater or equal to $T$. The solution provided at the end of this heuristic is a feasible production plan. Indeed, the Double-Fix-and-Relax method respects all the integrity constraints at the end of the procedure.

The motivation of setting an approximation window at the end of the planning horizon is detailed in Section 2. The rationale of the Double-Fix-and-Relax heuristic is to start by finding a default estimation of the production plan on the whole planning horizon by relaxing all the integrity constraints imposed on the setup variables. Such a plan captures the interaction, if it exists, between the first and the last periods of the planning horizon. Once this interaction is estimated in terms of capacity needs, the production plan at the end of the horizon is frozen in order to reduce the size of the sub-problems. The production variables fixed for these periods are progressively relaxed with the shift of the decision window. Indeed, at each stage of the algorithm, the decision window is followed by an approximation window which size must be large enough to correct the first default estimation and to carry out necessary anticipation on the total demand.

The Double-Fix-and-Relax heuristic is controlled by five parameters, three of them are relating to the decomposition of the planning horizon and the two others ones are connected with the stopping criteria of the branch-and-bound method. In addition to the Fix-and-Relax parameters described in section 3.1, we define the new parameter $\rho_k$ as the size of the approximation window at stage $k$ of the algorithm.

We denote by $P^0$ the MCLSP problem where all the integrity constraints are relaxed. We denote by $P^k_{a_k b_k c_k}$, the problem solved at each step $k$ of the heuristic with $k \geq 1$. We recall the features of the $P^k_{a_k b_k c_k}$ problem below (by convention, if $\alpha > \beta$ for section $[\alpha, \beta]$ then $[\alpha, \beta] = \emptyset$):

- Setup variables $y_{vit}$ and $z_{vjt}$ are frozen in the section of horizon $[1, a_k - 1]$ according to the solution obtained at stage $k-1$ of the heuristic,
- Integrity constraints on the setup variables $y_{vit}$ and $z_{vjt}$ are maintained in the decision window defined by the horizon section $[a_k, b_k]$,
- Integrity constraints on the setup variables $y_{vit}$ and $z_{vjt}$ are relaxed in the approximation window defined by the horizon section $[b_k + 1, c_k]$,
- Setup variables $y_{vit}$, $z_{vjt}$ and production variables $x_{vit}$ are frozen in the section of the horizon $[c_k + 1, T]$ according to the solution obtained at stage $k-1$ of the heuristic.

$K$ denotes the total number of iterations for the Double-Fix-and-Relax method. We recall that $K$ is in $O(T)$. The principle of the Double-Fix-and-Relax method is presented in figure 4.

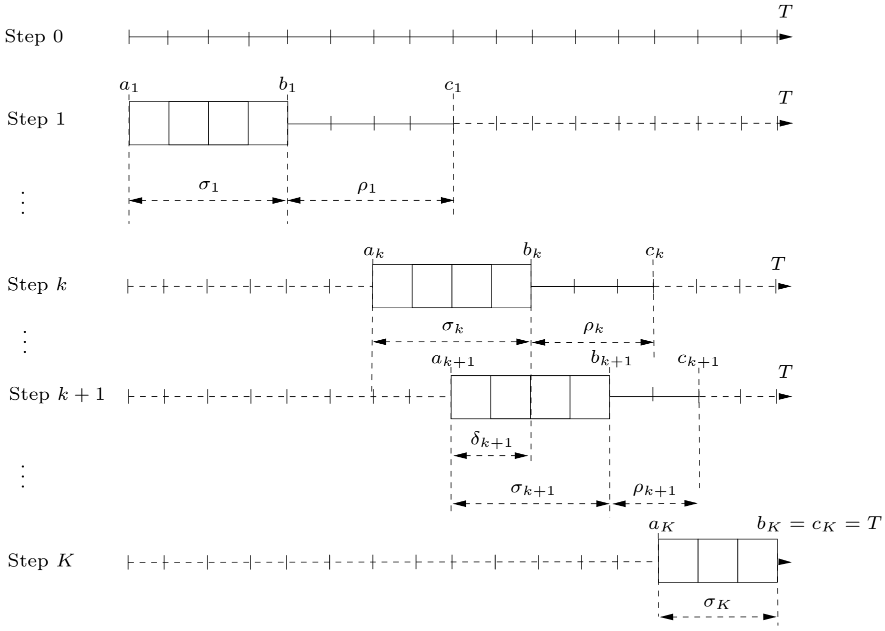

Figure 4: Double-Fix-and-Relax Method

Each sub-problem $P^k_{a_k b_k c_k}$ is solved using a branch-and-bound algorithm. Double-Fix-and-Relax heuristic is described by algorithm 2. The solution obtained is feasible for the MCLSP problem.

**Algorithm 2** Double-Fix-and-Relax

```
Solve the sub-problem $P^0$
$k \leftarrow 1$, $a_k \leftarrow 1$, $b_k \leftarrow \sigma_k$, $c_k \leftarrow b_k + \rho_k$
while $b_k < T$ do
    Solve the sub-problem $P^k_{a_k b_k c_k}$
    $k \leftarrow k+1$, $a_k \leftarrow b_{k-1} - \delta_k$, $b_k \leftarrow b_{k-1} + \sigma_k - \delta_k$, $c_k \leftarrow b_k + \rho_k$,
    if $c_k > T$ then
        $c_k \leftarrow T$
    end if
    if $b_k > T$ then
        $b_k \leftarrow T$
    end if
end while
Solve the sub-problem $P^k_{a_k b_k c_k}$
```

Some alternative algorithms could be derived from both of the two methods previously described (see Sections 3.1 and 3.2). Indeed, the production variables $x_{vit}$ could be fixed when we fix the setup variables. Such a procedure reduces considerably the size of the sub-problems solved at each stage of the heuristic. Therefore, the CPU time are lowered at each stage. Nevertheless, a disadvantage of such an alternative is the reduction of possible production anticipations. In fact, when the production is frozen, no additional production could be allowed.

### 3.3 Elements comparison

An advantage of the Double-Fix-and-Relax method in comparison with the Fix-and-Relax method is that less variables and constraints are considered at each step. Indeed, freezing the end of the horizon reduces the size of the sub-problems induced. Maintaining all the integrity constraints of the MCLSP problem leads to models for which it is very difficult to find a feasible solution by using a branch-and-bound method on real instances.

The principle of the Fix-and-Relax and Double-Fix-and-Relax methods is to solve iteratively several sub-problems $P^k_{a_k b_k}$ and $P^k_{a_k b_k c_k}$ with overlapping periods between two successive decision windows. If we assume that $\sigma_k = \sigma$ and $\delta_k = \delta$, instead of solving the MCLSP problem with $VT(N+J)$ binary variables and $(3N + R + 2VN + VJN)T$ constraints, the Fix-and-Relax and Double-Fix-and-Relax methods solve a maximum of $\lceil (T-\sigma)/(\sigma-\delta) \rceil + 1$ problems with $V(N+J)\sigma$ binary variables.

## 4 Computational experiments

In this section, we present some experimental issues. All of our algorithms are implemented in C++ and integrated in a generic APS software, n.SKEP[1]. We use the callable CPLEX 9.0 library [15] to solve MIP problems.

[1] n.SKEP : Supply Chain Management Software, DynaSys S.A.

## 4.1 Algorithms and implementation

Here is the list of the algorithms implemented to assess the heuristics effectiveness. We compare the Fix-and-Relax and Double-Fix-and-Relax methods with a monolithic resolution of the initial MCLSP problem. The MCLSP problem as well as the sub-problems generated at each stage of the heuristics are solved using CPLEX 9.0 solver which provides a branch-and-bound algorithm to solve MIP problems. Moreover, we can use with the solver several standard MIP cuts to strengthen the LP relaxation at each node. An alternative consisting in fixing the production variables at the same time as the setup variables is also developed.

- **BC:** algorithm based on the standard branch-and-cut of the solver;
- **FR:** algorithm based on the Fix-and-Relax method and the standard branch-and-cut of the solver to solve sub-problems $P^k_{a_k b_k}$;
- **FR1:** algorithm based on the Fix-and-Relax method and the standard branch-and-cut of the solver to solve sub-problems $P^k_{a_k b_k}$, where the production variables are fixed at the same time as the setup variables;
- **DFR:** algorithm based on Double-Fix-and-Relax method and the standard branch-and-cut of the solver to solve sub-problems $P^k_{a_k b_k c_k}$;
- **DFR1:** algorithm based on Double-Fix-and-Relax method and the standard branch-and-cut of the solver to solve sub-problems $P^k_{a_k b_k c_k}$. Here, the production variables are fixed at the same time as the setup variables.

For all the algorithms, we used the aggregated model defined in Section 1. The computations are performed on a Pentium IV 2.66 Ghz PC.

## 4.2 Test Instances

We carried out some tests on a set of instances resulting from real-world situations. In fact, the heuristics we have proposed are implemented in an APS software n.SKEP developed by Dynasys and they are used by its customers. These real benchmarks come from process industry contexts such as food industry. In these environments, manufacturers have several resources and production lines, the items are gathered into family and are constrained to minimum production runs. The most important objective of the decision maker is to minimize shortages on demand as well as safety stock deficits. These instances are characterized by a large number of groups and significant setup times. Items that share a common setup time are gathered into the same group. The shortage and deficit costs respectively for the demand and the safety stock could be viewed as penalty costs and their values are higher than other cost components. They have the feature of decreasing over the horizon. Actually, demands in the first periods of the horizon correspond to real orders by opposition to the demands in the last periods that are usually only forecasts. The characteristics of the instances are described in table 6.

| Instances | V | N | J | R | T |
|---|---|---|---|---|---|
| I1 | 3 | 264 | 144 | 21 | 19 |
| I2 | 3 | 469 | 215 | 20 | 19 |
| I3 | 3 | 292 | 144 | 20 | 30 |
| I4 | 3 | 514 | 215 | 20 | 30 |

Table 6: Test Instances

Moreover, the resource capacities are not very tight for I1 and I2. The instance I2 is characterized by high setup times. I3 and I4 are the largest instances, and they have minimum production constraints as well as high setup times. They are also characterized by tight capacities.

To assess the effectiveness of the heuristics, we have performed computational tests on a series of extended instances from the lot-sizing library LOTSIZELIB [17], initially described in Trigeiro *et al.* [26]. Trigeiro *et al.* [26] instances are denoted by $tr_{N-T}$, where $N$ is the number of items and $T$ is the number of periods. These are characterized by a variable resource consumption equal to one, and enough capacity to satisfy all the requirement over the planning horizon. They are also characterized, by an important setup cost, a small fixed resource requirement (setup time).

Since these instances have enough capacity to satisfy all the requirements over the planning horizon, we make some modifications to induce shortages. We have derived 12 new benchmarks[2] from the $tr_{N-T}$ instances by augmenting the fixed resource requirements (setup times), the variable resource requirements and by adding safety stocks. We have also generated shortage and safety stock deficit costs. More details are given below. These new benchmarks fall into 2 classes of 6 instances each:

- The class A was obtained by increasing the variable resource requirements and adding safety stocks. Variable resource requirements are multiplied by a coefficient $(1+\eta)$ such that $0 \leq \eta \leq 0.001 \times c_t$, $c_t$ represents the available resource capacity at period $t$. Safety stock coverage is the number of periods that must hold demand in stock. It is expressed in number of periods. Safety stock expressed in coverage is mainly used to cover the possible variation between forecasts and carried out sales. At each period, safety stock is calculated according to the future demands. Thus, safety stock is equal to the sum of the demands on the section of the horizon which starts at this period and which has the length of the safety stock coverage. Safety stock coverage is equal to $0.1 \times T$.
- The class B is based on the first one. In fact, we carried out some modifications on fixed resource requirements which are increased by multiplying them by a coefficient $(1+\tau)$ such that $\tau \approx 0.1 \times c_t$.

Shortage and safety stock deficit costs are considered as penalty costs and their values must be higher than other cost components. Therefore, They are fixed such that $\varphi_{it} >> \max_{i',t'} \left\{\alpha_{i't'}; \beta_{i't'}; \gamma^{+}_{i't'}; \gamma^{-}_{i't'}\right\}$ and $\gamma^{-}_{it} >> \max_{i',t'} \left\{\alpha_{i't'}; \beta_{i't'}; \gamma^{+}_{i't'}\right\}$. Moreover, $\varphi_{it}$ and $\gamma^{-}_{it}$ have the feature that they decrease over the horizon. In fact, demands in the first periods of the horizon correspond to real orders and not forecasts by opposition to the demands in the last periods that are usually only predictions. They are generated in the

[2] Test problems can be obtained from http://www-poleia.lip6.fr/ safia/lotsizing

same way for all the described instances.

## 4.3 Results

To compare the implemented algorithms, we use the following criteria. The first one called GAP is equal to $(UB - LB) / (UB)$, where $LB$ and $UB$ represent respectively the lower bound and the upper bound values at the termination of the branch-and-bound algorithm. The second criteria is a CPU time denoted $CPU - Time$. $\%Lost$ and $\%Deficit$ represent respectively the percentage of demand shortages and the percentage of safety stock deficits.

The proposed heuristics provide only upper bounds to the MCLSP problem. In order to evaluate the optimality percentage of these methods, we use the lower bound found at the end of the BC method.

The branching strategy in all the branch-and-bound algorithms is depth-first search to find a feasible solution. Upper bounds are either obtained when LP solutions are integral or by the standard LP based heuristics provided by the solver.

Fix-and-Relax and Double-Fix-and-Relax methods are controlled by several parameters. An empirical preliminary analysis showed that the following parameters lead to good results. For the Fix-and-Relax method, we set: $\sigma_k = 3$ for all $k$, $\delta_k = 1$ for all $k > 1$, $opt_k = 5\%$ for all $k$ and a time-limit of 300 seconds for each stage. At each step $k$, the branch-and-bound algorithm stops if $(UB_k - LB_k) / (UB_k)$ is lower than $opt_k$ or if the time-limit is reached. $LB_k$ and $UB_k$ represent respectively the lower bound and the upper bound values at the termination for the branch-and-bound algorithm of a step $k$ of the heuristic.

Apart from the previous setting, we define at stage $k$ of the Double-Fix-and-Relax algorithm, $\rho_k = 4$ for all $k$. The $BC$ method is controlled by a minimum optimality percent of 5% and a time-limit of 3600 seconds. The branch-and-bound algorithm stops if the minimum GAP or the time-limit criterion is satisfied. Table 7 summarizes the computational results when a time-limit and a GAP stopping criteria are used for all the algorithms on instances I1, I2, I3 and I4.

Clearly DFR method outperforms FR, FR1 and DFR1 regarding all the criteria, that is optimality percent, CPU time, shortages and deficits. We can notice that DFR leads to GAPs that are close to the ones found by BC on the instances I1, I2 and I4 with significantly lower CPU times. DFR and BC methods have almost the same percentage of shortages on instances I1, I2 and I4, but BC method gives less deficits on safety stocks. For instance I3, BC method does not find a good feasible solution after 3600 seconds. DFR gives a better solution after 629 seconds; FR1 finds an equivalent feasible solution after 3662 seconds. For instance I1, FR leads the best solution, nevertheless the CPU time is 1224 seconds. From table 7 we can conclude that DFR method gives the best compromise between the CPU time and the solution quality.

We also carried out tests on the instances of class A and class B. For the Fix-and-Relax method, we set: $\sigma_k = 3$ for all $k$, $\delta_k = 1$ for all $k > 1$, $opt_k = 1\%$ for all $k$ and a time-limit of 60 seconds for each stage. Apart from the previous setting, we define at stage $k$ of the Double-Fix-and-Relax algorithm, $\rho_k = 4$ for all

| $Method$ | $CPU - Time$ | $GAP$ | $UB$ | $LB$ | $\%Lost$ | $\%Deficit$ |
|---|---|---|---|---|---|---|
| Instance I1 | | | | | | |
| BC | 3600 | 25,44% | 616057049,33 | 459347279,99 | 0,00% | 0,86% |
| FR | 1133 | 24,04% | 604747858,07 | 459347279,99 | 0,01% | 0,95% |
| FR1 | 1224 | 29,68% | 653254932,91 | 459347279,99 | 0,03% | 0,97% |
| DFR | 365 | 27,62% | 634629402,32 | 459347279,99 | 0,00% | 0,88% |
| DFR1 | 521 | 30,99% | 665577091,20 | 459347279,99 | 0,01% | 1,12% |
| Instance I2 | | | | | | |
| BC | 3600 | 8,41% | 423228776,24 | 387628214,43 | 0,03% | 0,91% |
| FR | 200 | 15,80% | 460348758,04 | 387628214,43 | 0,03% | 1,50% |
| FR1 | 303 | 22,86% | 502473221,20 | 387628214,43 | 0,04% | 2,10% |
| DFR | 135 | 11,64% | 438691571,36 | 387628214,43 | 0,03% | 1,49% |
| DFR1 | 186 | 14,49% | 453295802,18 | 387628214,43 | 0,04% | 1,30% |
| Instance I3 | | | | | | |
| BC | 3600 | 72,98% | 4422837201,10 | 1195061357,10 | 1,07% | 10,15% |
| FR | 3480 | 80,73% | 6200212481,48 | 1195061357,10 | 1,32% | 17,90% |
| FR1 | 3662 | 24,29% | 1578441356,50 | 1195061357,10 | 0,16% | 6,66% |
| DFR | 629 | 24,44% | 1581640954,47 | 1195061357,10 | 0,16% | 8,10% |
| DFR1 | 1020 | 27,41% | 1646320778,18 | 1195061357,10 | 0,19% | 6,72% |
| Instance I4 | | | | | | |
| BC | 3600 | 12,47% | 6765340128,67 | 5921842374,92 | 3,04% | 4,22% |
| FR | 1770 | 22,52% | 7642976846,79 | 5921842374,92 | 3,38% | 6,62% |
| FR1 | 1567 | 22,78% | 7668991587,14 | 5921842374,92 | 3,15% | 6,29% |
| DFR | 483 | 17,55% | 7182226467,62 | 5921842374,92 | 3,32% | 5,60% |
| DFR1 | 1020 | 28,12% | 8238379889,67 | 5921842374,92 | 3,67% | 5,98% |

Table 7: Computational results I1, I2, I3 and I4.

$k$. The $BC$ method is controlled by a minimum optimality percent of 1% and a time-limit of 200 seconds. The branch-and-bound algorithm stops if the minimum GAP criterion or the time-limit criterion is satisfied. Table 8 summarizes the computational results when a time-limit and a GAP stopping criteria are used for all the algorithms on instances of class A and B.

From table 8, we can notice that class B problems are much more difficult than class A ones regarding the CPU times and the GAPs derived. Class B problems do have larger setup times than class A ones.

DFR and FR methods outperform FR1 and DFR1 regarding all the criteria. We can also notice that DFR and FR leads to GAPs that are close to the ones found by BC on the instances of class A with insignificant CPU times. The two methods give better GAPs than the ones found by BC on the instances of class B with significantly lower CPU times. We can easily notice that adding a frozen window at the beginning of the horizon induce a higher shortage amount and a lower safety stock deficit. We can also see that freezing the beginning of the horizon involves higher CPU times and optimality percents. DFR gives better GAPs and CPU times than FR for the instances of class A, while FR involves better GAPs and CPU times than DFR for the instances of class B. Generally, FR has less deficit on safety stock than DFR. The Methods DFR and FR give the best results while the methods FR1 and DFR1 give the worse results.

According to tables 7 and 8, we can say that freezing the end of the planning horizon helps to find good results. On the other hand, adding a frozen window at the beginning of the horizon does not allow significant improvements on the solution quality.

## 4.4 Sensitivity analysis

The performance of the Fix-and-Relax and Double-Fix-and-Relax methods depends on the parameter choice. In order to study the calibration sensitivity of these methods, we have performed computational tests on a

| $N$ | $T$ | $Method$ | $CPU-Time$ | $GAP$ | $UB$ | $LB$ | $\%Lost$ | $\%Deficit$ |
|---|---|---|---|---|---|---|---|---|
| Class A | | | | | | | | |
| 6 | 15 | BC | 200 | 1,18% | 4649063,73 | 4594355,62 | 27,05% | 68,06% |
| 6 | 15 | FR | 0,59 | 1,51% | 4664906,20 | 4594355,62 | 27,26% | 67,73% |
| 6 | 15 | FR1 | 0,64 | 2,50% | 4712137,71 | 4594355,62 | 27,35% | 66,45% |
| 6 | 15 | DFR | 0,64 | 1,51% | 4664906,20 | 4594355,62 | 27,26% | 67,73% |
| 6 | 15 | DFR1 | 0,67 | 3,17% | 4744841,17 | 4594355,62 | 27,96% | 67,48% |
| 6 | 30 | BC | 200 | 4,86% | 5373962,39 | 5112526,07 | 13,93% | 42,46% |
| 6 | 30 | FR | 2,67 | 6,28% | 5455078,94 | 5112526,07 | 14,06% | 42,93% |
| 6 | 30 | FR1 | 2,7 | 7,10% | 5503152,13 | 5112526,07 | 14,63% | 39,36% |
| 6 | 30 | DFR | 2,17 | 6,08% | 5443339,47 | 5112526,07 | 14,22% | 38,84% |
| 6 | 30 | DFR1 | 2,27 | 6,82% | 5486538,01 | 5112526,07 | 14,64% | 38,26% |
| 12 | 15 | BC | 200 | 1,63% | 8931582,99 | 8786329,68 | 24,03% | 68,81% |
| 12 | 15 | FR | 1,72 | 2,65% | 9025396,59 | 8786329,68 | 24,37% | 69,04% |
| 12 | 15 | FR1 | 1,64 | 3,31% | 9087554,06 | 8786329,68 | 24,87% | 69,45% |
| 12 | 15 | DFR | 1,59 | 3,03% | 9060634,23 | 8786329,68 | 24,58% | 69,89% |
| 12 | 15 | DFR1 | 1,7 | 4,23% | 9174185,95 | 8786329,68 | 25,20% | 71,06% |
| 12 | 30 | BC | 200 | 5,91% | 10488446,95 | 9868131,34 | 13,28% | 36,73% |
| 12 | 30 | FR | 7,14 | 6,58% | 10563117,63 | 9868131,34 | 13,39% | 41,38% |
| 12 | 30 | FR1 | 8,3 | 8,66% | 10803451,49 | 9868131,34 | 14,09% | 37,15% |
| 12 | 30 | DFR | 5,92 | 5,90% | 10487023,27 | 9868131,34 | 13,15% | 42,75% |
| 12 | 30 | DFR1 | 6,31 | 8,34% | 10766018,53 | 9868131,34 | 13,95% | 39,96% |
| 24 | 15 | BC | 200 | 1,46% | 16619159,95 | 16375783,15 | 24,00% | 67,35% |
| 24 | 15 | FR | 4,22 | 2,01% | 16712435,78 | 16375783,15 | 24,28% | 69,26% |
| 24 | 15 | FR1 | 3,72 | 3,57% | 16981781,35 | 16375783,15 | 24,87% | 68,81% |
| 24 | 15 | DFR | 3,38 | 1,81% | 16678253,35 | 16375783,15 | 24,25% | 69,58% |
| 24 | 15 | DFR1 | 3,31 | 3,91% | 17041520,56 | 16375783,15 | 25,03% | 70,51% |
| 24 | 30 | BC | 200 | 3,09% | 27184143,95 | 26345431,03 | 19,02% | 27,97% |
| 24 | 30 | FR | 14,8 | 3,41% | 27275422,81 | 26345431,03 | 18,50% | 31,27% |
| 24 | 30 | FR1 | 17,19 | 4,42% | 27563791,40 | 26345431,03 | 19,18% | 27,86% |
| 24 | 30 | DFR | 12,53 | 3,74% | 27369808,34 | 26345431,03 | 18,74% | 30,11% |
| 24 | 30 | DFR1 | 14,7 | 4,21% | 27504422,93 | 26345431,03 | 19,11% | 27,34% |
| Class B | | | | | | | | |
| 6 | 15 | BC | 200 | 2,27% | 5830113,28 | 5698078,38 | 34,47% | 59,13% |
| 6 | 15 | FR | 0,69 | 2,73% | 5858168,59 | 5698078,38 | 34,94% | 62,79% |
| 6 | 15 | FR1 | 0,67 | 2,52% | 5845495,92 | 5698078,38 | 34,67% | 60,13% |
| 6 | 15 | DFR | 0,75 | 2,39% | 5837544,14 | 5698078,38 | 34,60% | 59,49% |
| 6 | 15 | DFR1 | 0,64 | 2,52% | 5845187,18 | 5698078,38 | 34,67% | 60,17% |
| 6 | 30 | BC | 200 | 5,14% | 7887366,21 | 7482152,71 | 22,20% | 39,53% |
| 6 | 30 | FR | 1,86 | 4,47% | 7832065,14 | 7482152,71 | 21,95% | 40,24% |
| 6 | 30 | FR1 | 1,84 | 4,29% | 7817366,30 | 7482152,71 | 21,94% | 41,27% |
| 6 | 30 | DFR | 1,75 | 4,47% | 7832065,14 | 7482152,71 | 21,95% | 40,24% |
| 6 | 30 | DFR1 | 1,53 | 4,25% | 7814117,13 | 7482152,71 | 21,91% | 41,75% |
| 12 | 15 | BC | 200 | 4,34% | 13376789,44 | 12796410,89 | 38,12% | 60,67% |
| 12 | 15 | FR | 3,58 | 3,73% | 13292194,25 | 12796410,89 | 37,74% | 60,98% |
| 12 | 15 | FR1 | 3,33 | 5,63% | 13559460,73 | 12796410,89 | 39,05% | 60,57% |
| 12 | 15 | DFR | 4,17 | 3,88% | 13312283,78 | 12796410,89 | 37,79% | 61,00% |
| 12 | 15 | DFR1 | 3,11 | 5,07% | 13479203,38 | 12796410,89 | 38,69% | 59,68% |
| 12 | 30 | BC | 200 | 16,18% | 17242514,85 | 14453922,45 | 23,72% | 42,12% |
| 12 | 30 | FR | 20,34 | 14,92% | 16988946,09 | 14453922,45 | 23,24% | 42,86% |
| 12 | 30 | FR1 | 19,05 | 17,05% | 17424705,06 | 14453922,45 | 24,69% | 43,13% |
| 12 | 30 | DFR | 24,09 | 15,03% | 17009714,00 | 14453922,45 | 23,70% | 46,02% |
| 12 | 30 | DFR1 | 18,81 | 16,62% | 17334236,26 | 14453922,45 | 24,45% | 42,15% |
| 24 | 15 | BC | 200 | 7,55% | 28920671,65 | 26737370,61 | 43,68% | 62,30% |
| 24 | 15 | FR | 10,58 | 6,69% | 28655048,09 | 26737370,61 | 43,52% | 60,50% |
| 24 | 15 | FR1 | 8,86 | 7,16% | 28799376,04 | 26737370,61 | 43,99% | 60,45% |
| 24 | 15 | DFR | 17,84 | 5,68% | 28348357,04 | 26737370,61 | 42,84% | 60,81% |
| 24 | 15 | DFR1 | 7,2 | 6,49% | 28593139,10 | 26737370,61 | 43,39% | 60,94% |
| 24 | 30 | BC | 200 | 14,84% | 46725651,41 | 39792758,6 | 33,18% | 44,23% |
| 24 | 30 | FR | 45,33 | 12,57% | 45512020,24 | 39792758,6 | 32,33% | 43,86% |
| 24 | 30 | FR1 | 60,56 | 14,88% | 46746268,85 | 39792758,6 | 33,66% | 43,50% |
| 24 | 30 | DFR | 56,28 | 12,72% | 45591716,51 | 39792758,6 | 32,44% | 46,02% |
| 24 | 30 | DFR1 | 98,69 | 15,87% | 47297771,66 | 39792758,6 | 33,95% | 43,65% |

Table 8: Computational results for Class A and B.

series of extended instances from the lot-sizing library LOTSIZELIB [17], initially described in Trigeiro *et al.* [26]. Particularly, we analyze the variation of the parameters on the CPU time and the quality of the solutions found by the heuristic Double-Fix-and-Relax since the previous computational results showed that is the best method. An empirical analysis showed that large values of $\sigma_k$ do not lead to good values of CPU Time. We have noticed that having $\sigma_k$ lower that 5 periods gives the best compromise between the GAP and the CPU time.

In what follows, we assume $\sigma_k = \sigma$, $\delta_k = \delta$, $Time_k = time$ and $opt_k = opt$ for all $k$. The parameter $\sigma$ takes the values $1, 2, 3$ and $4$. The parameter $\delta$ takes values between: $0, \ldots, \sigma - 1$. The stopping criteria for all the algorithms are set to $opt = 1\%$ and a time-limit of 60 seconds for each stage. We also study the variation of the stopping criteria parameters *time* and *opt*. The time-limit for each step *time* takes the values $1, 3, 5, 10, 20$ and $30$ seconds with $opt = 0$. paramater *opt* takes the values $1\%, 2\%, 3\%, 4\%, 5\%$ and $6\%$ with $time = 60$ seconds. The parameter $\rho$ is equal to 4 for all the algorithms.

The GAP values are calculated with the upper bound found by the heuristic and the best lower bound obtained at the end of BC method. We allow a maximum of 200 seconds CPU time for the BC method. Computational results are summarized by the following figures. Figures 5 and 6 report respectively the variation of the quality of the solution computed by GAP and CPU Time according to $\sigma$, with $\delta$ varying between 0 and $\sigma - 1$, GAP and CPU Time represent the averages of several tests according to the variation of $\delta$. Figures 7 and 8 report respectively the variation of GAP and CPU Time according to $\delta$, with $\sigma$ between 1 and 4, GAP and CPU Time represent the averages of different tests according to the variation of $\sigma$.

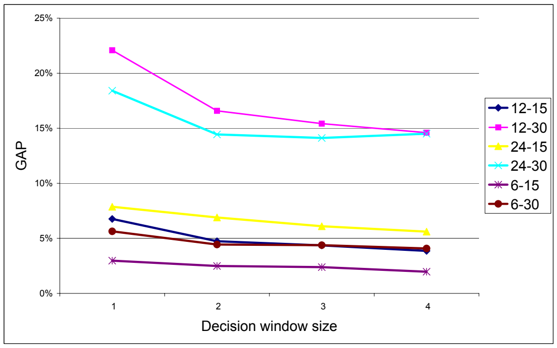

Figure 5: GAP variation according to $\sigma$

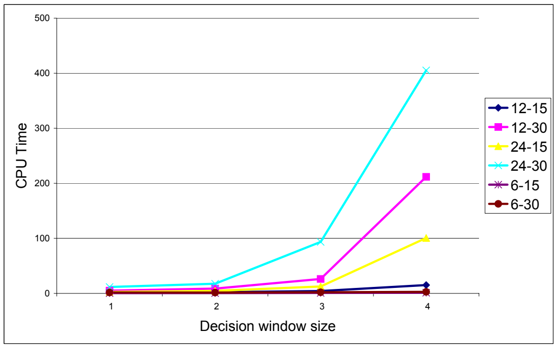

Figure 6: CPU Time variation according to $\sigma$

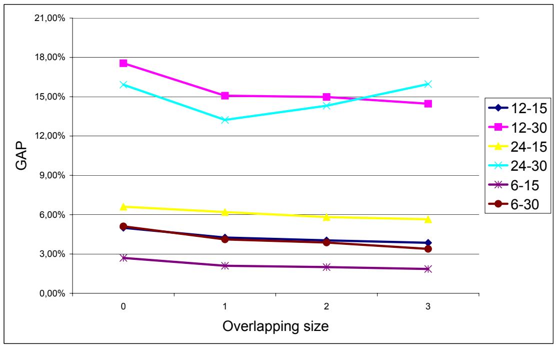

Figure 7: GAP variation according $\delta$

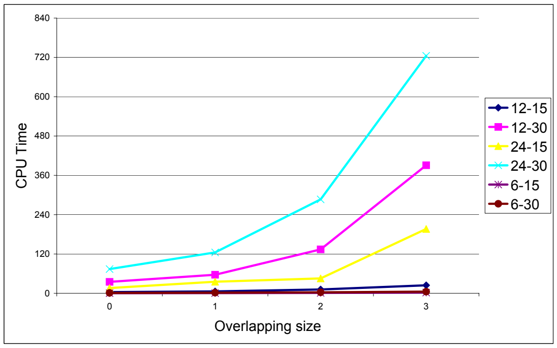

Figure 8: CPU Time variation according to $\delta$

From figures 5, 6, 7 and 8, we can notice that the increase of $\sigma$ and $\delta$, grows up considerably the CPU times while allowing an improvement on the GAPs induced.

Figures 9 and 10 report respectively the variation of GAP and CPU Time according to *opt*. The parameters $\sigma$ and $\delta$ are fixed respectively to 3 and 1 since the previous analysis shows that this tuning gives the best compromise between the GAP and the CPU time.

From figures 9 and 10 we can notice that the increase of *opt* reduces the CPU times but it grows up the

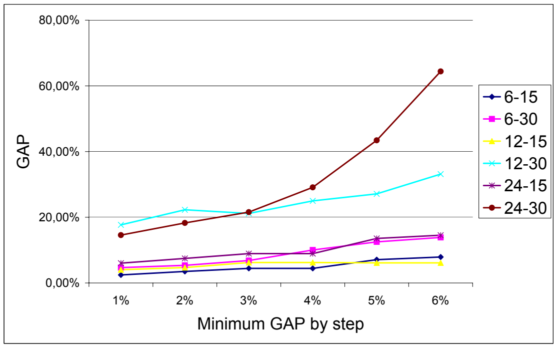

Figure 9: GAP variation according to *opt*

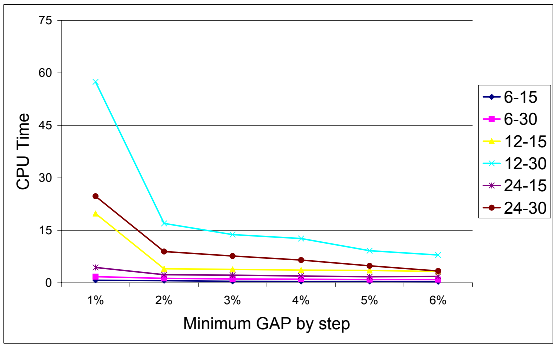

Figure 10: CPU Time variation according to *opt*

GAP considerably.

We have also studied the variation of the GAP parameter according to the time-limit. We have noticed that the GAP parameter decreases when the time-limit increases for values lower than 5 seconds. When the time-limit is greater than 5 seconds, GAP does not decrease considerably.

In our computational experiments, we have assumed that the parameters $\sigma_k$ and $\delta_k$ are constant, it would be interesting to analyze the variation of these parameters thought the heuristics execution. For example, taking a large values of $\sigma_k$ and $\delta_k$ at the beginning of the heuristic and reduce these values when shifting forward the decision window.

# 5 Conclusion

In this paper, we propose a mathematical formulation of a new multi-item capacitated lot-sizing problem with setup times. This formulation takes into account several industrial constraints, shortage costs, safety stock deficit costs, several item-groups, multiple resources, multiple operating sets and minimum production constraints. To derive a good feasible solution, we develop new MIP-based heuristics. These hybrid algorithms are based on a planning horizon decomposition and a branch-and-bound approach. Computational experiments showing the effectiveness of the proposed methods are given. The principle of these heuristics can be used to solve other intractable lot-sizing problems. These methods can be improved by using a better approximation of the setup variables at the end of the horizon; these heuristics can also be used in conjunction with a polyhedral approach to solve sub-problems optimally (see [2], [20]). Feasible solutions provided by heuristics can be used as upper bounds to accelerate branch-and-cut methods. Another issue of this work is to use multicriteria optimization approach. In fact, the objective function is a sum of several cost components. Shortage and safety stock deficit costs are higher in comparison with other cost components. It would be interesting to find a set of effective or non-dominated solutions based on these two criteria.

# References

[1] N. Absi. *Modélisation et résolution de problèmes de lot-sizing à capacité finie.* PhD thesis, Université Pierre et Marie Curie (Paris VI), 2005.

[2] N. Absi and S. Kedad-Sidhoum. Multi-item capacitated lot-sizing problem with setup times and shortage costs. Working paper - Available at http://www.optimization-online.org/DB_HTML/2005/09/1212.html, July 2005.

[3] D. Aksen, K. Altinkemer, and S. Chand. The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, 147:558–566, 2003.

[4] G. Belvaux and L.A. Wolsey. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, 46:724–738, 2000.

[5] G. Bitran and H.H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28:1174–1186, 1982.

[6] N. Brahimi, S. Dauzère-Pérès, N.M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168 (1):1–16, 2006.

[7] W.H. Chen and J.M. Thizy. Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26:29–72, 1990.

[8] A.R. Clark. Hybrid heuristics for planning lot setups and sizes. *Computers & Industrial Engineering*, 44:545–562, 2003.

[9] M. Diaby, H.C. Bahl, M.H. Karwan, and S. Zionts. Capacitated lot-sizing and scheduling by lagrangean relaxation. *European Journal of Operational Research*, 59(3):444–458, 1992.

[10] J.R. Evans. An efficient implementation of the wagner-whitin algorithm for dynamic lot-sizing. *Journal of Operations Management*, 5 (2):229–235, 1985.

[11] M. Florian, J.K. Lenstra, and A.H.G. Rinnoy Kan. Deterministic production planning : algorithms and complexity. *Management Science*, 26:669–679, 1980.

[12] J.D. Kelly. Chronological decomposition heuristic for scheduling: A divide & conquer method. *AIChE Journal*, 48(12):2995–2999, 2002.

[13] J. Krarup and O. Bilde. Plant location, set covering and economic lot sizes: An o(mn)-algorithm for structured problems, in optimierung bei graphentheoretischen and ganzzahligen probleme. *L. COLLATZ et al. (eds), Birkhauser Verlag, Basel*, pages 155–180, 1977.

[14] J.M. Leung, T.L. Magnanti, and R. Vachani. Facets and algorithms for capacitated lot-sizing. *Mathematical Programming*, 45:331–359, 1989.

[15] CPLEX Callable Library. Ilog sa. http://www.ilog.com, 2005.

[16] M. Loparic, Y. Pochet, and L.A. Wolsey. The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89:487–504, 2001.

[17] LOTSIZELIB. http://www.core.ucl.ac.be:16080/wolsey/lotsizel.htm, 1999.

[18] C. Mercé and G. Fontan. Mip-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics*, 85:97–111, 2003.

[19] A. J. Miller. *Polyhedral Approaches to Capacitated Lot-Sizing Problems.* PhD thesis, Georgia Institute of Technology, 1999.

[20] A. J. Miller, G.L. Nemhauser, and M.W.P. Savelsbergh. On the polyhedral structure of a multi-item production planning model with setup times. *Mathematical Programming*, 94:375–405, 2003.

[21] Y. Pochet and L.A. Wolsey. Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, 37:53–67, 1991.

[22] R. Roundy. Efficient, effective lot-sizing for multi-product, multi-stage production systems. *Operations Research*, 41:371–386, 1993.

[23] R.A. Sandbothe and G.L. Thompson. A forward algorithm for the capacitated lot size model with stockouts. *Operations Research*, 38 (3):474–486, 1990.

[24] H. Stadtler. Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research*, 51(3):487–502, 2003.

[25] C. Suerie and H. Stadtler. The capacitated lot-sizing problem with linked lot sizes. *Management Science*, 49:1039–1054, 2003.

[26] W. Trigeiro, L.J. Thomas, and J.O. McLain. Capacitated lot-sizing with setup times. *Management Science*, 35:353–366, 1989.

[27] M. Van Vyve Y. Pochet. A general heuristic for production planning problems. *INFORMS Journal on Computing*, 16:316–327, 2004.