4OR-Q J Oper Res (2015) 13:173–198
DOI 10.1007/s10288-014-0266-3

RESEARCH PAPER

A Lagrangian heuristic for capacitated single item lot sizing problems

Nadjib Brahimi · **Stéphane Dauzère-Pérès**

Received: 27 June 2014 / Published online: 23 July 2014

Abstract This paper presents a new Lagrangian heuristic to solve the general capacitated single item lot sizing problem (CSILSP) where the total costs of production, setup, and inventory are to be minimized. We introduce a pre-smoothing procedure to transform the problem into a CSILSP with non-customer specific time windows and relax constraints that are specific to the CSILSP. The resulting uncapacitated single item problems with non-customer specific production time windows can be solved in polynomial time. We also analyze the performance of the Lagrangian heuristic for solving the CSILSP with non-customer specific time windows. Finally, the heuristic is adapted to the customer specific case. The introduction of pre-smoothing and the relaxation of CSILSP-specific constraints help to decrease the gap between lower bounds and upper bounds from 26.22 to 1.39 %, on average. The heuristic can be used to solve aggregate production planning problems, or can be integrated into a general procedure to solve more complex lot sizing problems.

Keywords Lot sizing · Time windows · Single item · Lagrangian relaxation

Mathematics Subject Classification 90 OR Math. Programming · 90C11 Mixed Integer Programming · 90C59 Approximation methods and heuristics

N. Brahimi (✉)
Department of Industrial Engineering and Management, College of Engineering,
University of Sharjah, P.O. Box 27272, Sharjah, United Arab Emirates
e-mail: nbrahimi@sharjah.ac.ae

S. Dauzère-Pérès
Department of Manufacturing Sciences and Logistics, CNRS UMR 6158, LIMOS,
Ecole des Mines de Saint-Etienne - CMP, 13541 Gardanne, France
e-mail: dauzere-peres@emse.fr

1 Introduction

The capacitated single item lot sizing problem (CSILSP) is a planning problem in which time-varying demands for a single product over T periods have to be met while satisfying capacity constraints. More precisely, the problem consists of determining the quantities to be produced/supplied in all periods to fulfill the demands while considering the maximum production/supply capacity in each period. The objective is to minimize the total cost which combines production cost, setup cost, and inventory holding cost. There is a rich literature on the single item lot sizing problem (SILSP) and its extensions since the seminal paper by Wagner and Whitin (1958). A comprehensive survey can be found in Brahimi et al. (2006b). In addition to capacity constraints, other extensions of the SILSP include backlogging (Zangwill 1969), remanufacturing (Simpson 1978), bounded inventories (Love 1973), perishability (Nahmias 1982) and production time windows (Dauzère-Pérès et al. 2002). In this paper, we solve the CSILSP with production time windows and we use some properties to convert the classical CSILSP into a lot sizing problem with time windows to solve it in a more efficient way.

The CSILSP can be used by companies that want to focus on a single aggregate product. By performing multiple simulation runs, it is possible to forecast and plan for future capacities. The CSILSP is also relevant for solving more complex lot sizing problems in particular as a subproblem when decomposing multi-item, multi-level problems. Moreover, identifying the characteristics of the CSILSP provides insights on more complex problems.

The general CSILSP is an NP-hard problem as shown by Bitran and Yanasse (1982). The best algorithms to solve the general CSILSP are pseudo-polynomial and their efficiency depends on the magnitude of the data. This is the case for both dynamic programming algorithms proposed by Chen et al. (1994) and Shaw and Wagelmans (1998). However, there are special cases of the problem, in particular when capacities are constant, which are solvable in polynomial time (Ex. Florian and Klein (1971) and Van Hoesel and Wagelmans (1996) for the case without backlogging; and Ou (2012) for the backlogging case).

In this paper, we propose a Lagrangian heuristic to solve the general CSILSP. The originalities of the heuristic are summarized below.

1. Capacity constraints specific to the CSILSP are identified and relaxed.
2. Before running the heuristic, the original problem is modified by pre-smoothing demands in backward and forward directions.
3. To update the Lagrangian multipliers, learning capabilities have been embedded into the heuristic to change the step size.
4. The heuristic applies both to the CSILSP with and without production time windows.

Several heuristics have been developed to solve the capacitated multi-item lot sizing problem (CLSP) and the uncapacitated single item lot sizing problem (USILSP). Simple heuristics for the USILSP are often used as building blocks to solve more complex problems. Examples of such heuristic include Silver and Meal (1973), Bitran et al. (1984), Saydam and Evans (1990), and Vachani (1992). The CLSP received a lot of interest in the Operations Research literature and many types of heuristics were developed. They include simple heuristics such as Dixon and Silver (1981) and Maes and Van Wassenhove (1986) and more advanced heuristics such as those based on Lagrangian relaxation. Among the earliest works that implemented Lagrangian heuristics to solve the CLSP, we can cite Thizy and Van Wassenhove (1985) and Chen and Thizy (1990) for the CLSP without setup times and Trigeiro et al. (1989) and Merle et al. (2000) when setup times are considered. Other extensions on which Lagrangian relaxation was used include, for example, the multi level case (Billington et al. 1986), problems with backlogging (Millar and Yang 1994), lot sizing with parallel machines (Toledo and Armentano 2006), and lot sizing with production time windows (Brahimi et al. 2006a, 2010b). A review of CLSPs and those which were solved using Lagrangian relaxation can be found in Quadt and Kuhn (2008) and Buschkühl et al. (2010). On the other hand, little research has been conducted on heuristic approaches for the CSILSP though the solution approaches of the CSILSP can be integrated in the solution of more complex problems similarly to USILSP approaches. The only CSILSP heuristics are polynomial approximation schemes (e.g. Van Hoesel and Wagelmans 2001) but their practical value remains limited as they still require high computational times even for moderate factors (Hardin et al. 2007).

In this paper, we adapt some of the most successful CLSP heuristics to solve the CSILSP with and without time windows. The adapted heuristics are the TTM heuristic of Trigeiro et al. (1989) and its extension to the lot sizing problem with time windows proposed in Brahimi et al. (2006a). We show that, by considering specifities of the CSILSP, the Lagrangian heuristics become much more efficient than if they were applied as-is on the single item case, by just setting the number of items to 1. The first specificity is related to the capacity constraint in which binary setup variables can be included. This further strengthen the formulation of the problem but it mainly improves the performance of the Lagrangian heuristics. The second specificity concerns the possibility of forward and backward smoothing demands. Backward pre-smoothing is common in the literature (e.g. Wolsey 2006). It starts at the last period T and repeatedly shifts demands exceeding capacity to previous periods. A constant inventory holding cost is calculated and added to the objective at the end of the optimization. Forward pre-smoothing is a new idea. It consists of considering the availability of demands. In classical single item lot sizing problems, it is assumed that all demands can be processed as early as the first period of the planning horizon as long as there is enough capacity. This implies that all demands are available in the beginning of the planning horizon. Although forward pre-smoothing is comparable to backward pre-smoothing applied to the available demands, it generates a problem with a different structure: A CSILSP with non-customer specific production time windows (CSILSP-NCS). By introducing the concept of production time windows in 2002, Dauzère-Pérès et al. (2002) show the importance of considering constraints on the availability of demands with different applications such as remanufacturing (van den Heuvel and Wagelmans 2008), bounded inventory (Wolsey 2006), major and minor demands (Hwang and Jaruphongsa 2008), and raw material availability (Brahimi et al. 2010a).

The CSILSP with production time windows consists of processing customer demands *which are not necessarily available at the first period of the planning horizon*. A time window demand $d_{t1,t2}$ is characterized by the fact that production cannot start

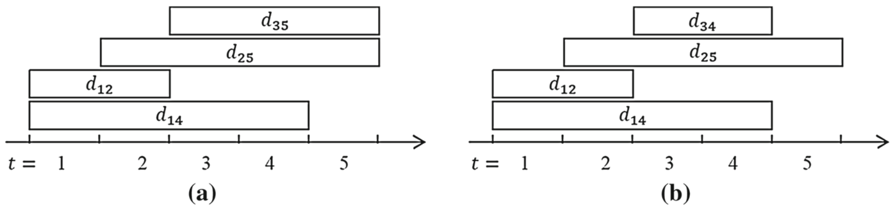

Fig. 1 Demand time windows: **a** non-customer specific (NCS) and **b** customer specific (CS)

before its release period $t1$ and must be delivered not later than its due date $t2$. In the *Non-Customer Specific (NCS)* problem, given any two time windows, one cannot be *strictly* included in the other. In other words, for any pair of time windows with positive demands $((t1, t2), (t3, t4))$, either "$t1 \leq t3$ and $t2 \leq t4$" or "$t1 \geq t3$ and $t2 \geq t4$". The general case where there is no restriction on the structure of time windows is called *Customer Specific (CS)*. Figure 1 illustrates the difference between (a) non-customer specific and (b) customer specific time windows. Case (b) is customer specific because the time window of d_{34} is strictly included in time window [2, 5] of d_{25}.

Dauzère-Pérès et al. (2002) also identified some interesting properties of the problem and suggested dynamic programming algorithms to solve the uncapacitated case using an exponential time algorithm for the CS problem and an $O(T^4)$ algorithm for the NCS problem. Later, Wolsey (2006) further analyzed the two cases and proposed improved algorithms, in particular, an $O(T^2)$ algorithm for the NCS problem. The CS problem was also solved by Hwang (2007) using an $O(T^4)$ algorithm. The equivalence between lot sizing problems with production time windows, the lot sizing problem with bounded inventory, the lot-sizing problem with remanufacturing options, and the lot sizing problem with cumulative capacities is discussed in van den Heuvel and Wagelmans (2008).

The remainder of the paper is organized as follows. Section 2 is devoted to the mathematical models used in the development of the lagrangian heuristic and the determination of feasibility conditions of the problem. Characteristics of the problem are discussed in Sect. 3. The Lagrangian heuristic is introduced in Sect. 4 and numerical results are given and analyzed in Sect. 5. Conclusions and perspectives are presented in Sect. 6.

2 The problem

The first considered problem is the classical capacitated single item lot sizing problem (CSILSP) solved using an approach in which the problem is converted into a lot sizing problem with non-customer specific time windows (CSILSP-NCS). This section recalls the aggregate and disaggregate formulations of the CSILSP. A natural extension of this work is the application of the heuristic to the CSILSP-NCS, as it does not require any conversion of demands. This is why we extend the CSILSP formulations for the CSILSP-NCS. Finally, we propose a modified version of the heuristic for the CSILSP with customer specific time windows (CSILSP-CS). Specific constraints are added to the previous formulations.

2.1 Formulations

2.1.1 Formulations of the CSILSP

The CSILSP problem is first modeled using the standard formulation (aggregate model). The decision variables are X_t, the quantity of product to be produced/supplied in period t; I_t, the inventory level at the end of period t; and Y_t, a binary setup variable which is equal to 1 if $X_t > 0$, and 0 otherwise. The parameters are the following:

T: number of periods in the horizon,

d_t: Customer aggregate demand that must be delivered at the end of period t,

h_t: Holding cost per unit of product in stock at the end of period t,

p_t: Unit production/supplying cost in period t,

s_t: Fixed setup cost in period t,

C_t: Production/Supply capacity at period t.

The aggregate model of the SILSP without capacity constraints can be written:

Model CSILSP-AGG

$$\text{Minimize} \sum_{t=1}^{T}(h_t I_t + p_t X_t + s_t Y_t) \tag{1}$$

Subject to:

Capacity Constraints (Constraints (6) **or** (7) **below) and**

$$I_t = I_{t-1} + X_t - d_t \quad \forall t \tag{2}$$

$$X_t \leq \sum_{k=t}^{T} d_k Y_t \quad \forall t \tag{3}$$

$$X_t, I_t \geq 0 \quad \forall t \tag{4}$$

$$Y_t \in \{0, 1\} \quad \forall t \tag{5}$$

The objective (1) is to minimize total setup, production, and holding costs. Constraints (2) are the inventory balance equations. We assume without loss of generality that the starting and ending inventories are null. Constraints (3) link the continuous production variables to the binary setup variables. The capacity constraints can be written as:

$$X_t \leq C_t \quad \forall t \tag{6}$$

Or:

$$X_t \leq C_t Y_t \quad \forall t \tag{7}$$

In the mathematical programming formulation of multi-item capacitated lot sizing problems, the capacity constraints are modeled with (6) where all items are considered. One of the objectives of this study is to show that it is not efficient to apply a lagrangian heuristic that is developed for multi-item problems directly on single-item problems. In Lagrangian relaxations for multi-item lot sizing problems, capacity constraints (equivalent to Constraints (6)) are usually relaxed, resulting in uncapacitated single-item lot sizing problems solved in $O(T \log T)$ (see for example Wagelmans et al. 1992). In the case of CSILSPs, we will show that considerable improvements are brought to the solution when Constraints (7) are used as capacity constraints.

Section 5 presents a comparison of the performance of our heuristics with the results obtained using a commercial solver. Though the above formulation is the most common way to model the CSILSP, it is not the most suitable to use in a mixed integer linear programming commercial solver. We apply the solver to the following disaggregate formulation which is based on the FAcility Location formulation (FAL). A new decision variable Z_{kt} is introduced, which corresponds to the quantity produced/supplied in period k to satisfy the demand in period t.

Model CSILSP-FAL

$$\text{Minimize} \sum_{t=1}^{T} \sum_{k=t}^{T} (p_t + \sum_{j=t}^{k-1} h_j) Z_{tk} + \sum_{t=1}^{T} s_t Y_t \tag{8}$$

Subject to:

$$\sum_{k=1}^{t} Z_{kt} = d_t \quad \forall t \tag{9}$$

$$Z_{tk} \leq d_k Y_t \quad \forall t, k \geq t \tag{10}$$

$$\sum_{k=t}^{T} Z_{tk} \leq C_t \quad \forall t \tag{11}$$

$$Z_{tk} \geq 0 \quad \forall t, k \geq t \tag{12}$$

$$Y_t \in \{0, 1\} \quad \forall t \tag{13}$$

The objective (8) minimizes the total cost. Constraints (9) ensure that all demands are satisfied without delay. Constraints (10) link the continuous variables Z_{tk} with the binary variables Y_t. Constraints (11) are capacity restrictions.

2.1.2 Formulations of the CSILSP with non-customer specific time windows

The CSILSP with production time windows is characterized by the time window demands $d_{t1,t2}$. In the next paragraphs, we call CSILSP-NCS the CSILSP with non-customer specific time windows and CSILSPS-CS the case where time window demands are customer specific. In this section, we consider the formulation of the CSILSP-NCS. From the formulations of the CSILSP, we keep the same notations.

The aggregate demand d_t still corresponds to total demand due in period t such that:

$$d_t = \sum_{t1=1}^{t} d_{t1,t} \quad \forall t \tag{14}$$

In other words, the CSILSP problem is a special case of the CSILSP-NCS where all demands are available in the first period of the planning horizon. That is, $d_t = d_{1,t}, \forall t$.

The MILP formulation of the CSILSP-NCS is composed of the objective function (1) and Constraints (2)–(6) together with the following availability constraints:

$$\sum_{k=1}^{t} X_k \leq \sum_{k=1}^{t} \sum_{j=k}^{T} d_{k,j} \quad \forall t \tag{15}$$

It will also be shown in the numerical experiments that the Lagrangian relaxation heuristic applied to the CSILSP-NCS performs much better if Constraints (6) are replaced with Constraints (7).

2.1.3 Formulations of the CSILSP with customer specific time windows

In the CSILSP with customer specific time windows (CSILSP-CS), time windows can be strictly inclusive as shown in the example of Fig. 1b. The formulation of this problem corresponds to that of the CSILSP-NCS in addition to the following constraints which state that there must always be enough production in any time window $(t1, t2)$:

$$\sum_{k=t1}^{t2} X_k \geq \sum_{k=t1}^{t2} \sum_{j=k}^{t2} d_{k,j} \quad \forall t1, t2 \ (t1 \leq t2) \tag{16}$$

The FAL formulation for the CSILSP-NCS and CSILSP-CS is composed of the objective function (8), Constraints (9)–(13), and the following constraint on demand availability:

$$\sum_{k=1}^{t1} Z_{k,t2} \leq \sum_{k=1}^{t1} d_{k,t2} \quad \forall t1, t2 \ (t1 \leq t2) \tag{17}$$

2.2 Problem feasibility

It is well-known that the necessary and sufficient condition for the CSILSP to have a feasible solution is that:

$$\sum_{k=1}^{t} C_k \geq \sum_{k=1}^{t} d_k \quad \forall t \tag{18}$$

Lemma 1 *A necessary and sufficient condition for the existence of a feasible solution for the capacitated single item problem with time windows is:*

$$\sum_{k=t1}^{t2} C_k \geq \sum_{k=t1}^{t2} \sum_{l=k}^{t2} d_{kl} \quad \forall t1, t2\ (t1 \leq t2) \tag{19}$$

Proof Consider the aggregate formulation of the customer specific problem which represents the general case. It is possible to eliminate inventory variables and replace them in the formulation by: $\sum_{k=1}^{t} X_k - \sum_{k=1}^{t} \sum_{j=1}^{k} d_{jk}$. Thus, the inventory balance equations (2) and non-negativity constraints (4) on inventory variables will be replaced with the following two constraints:

$$\sum_{t=1}^{T} X_t = \sum_{t=1}^{T} \sum_{j=1}^{t} d_{jt} \tag{20}$$

$$\sum_{i=1}^{t} X_i \geq \sum_{k=1}^{t} \sum_{j=1}^{k} d_{jk} \quad t = 1, \ldots, T-1 \tag{21}$$

Constraints (21) are the only constraints which impose minimal quantities to be produced within every interval. There should be enough capacity within the corresponding intervals. This results in Condition (19).

We prove that (19) is a sufficient condition by showing that it is always possible to find a feasible solution if this condition is satisfied. Assume that (19) is valid. Then, starting at the first period of the planning horizon, produce the minimum of the available capacity and the total demand that becomes available at the current period. By proceeding in this way, time window Constraints (16) are never violated. Also Constraints (15) are satisfied since what becomes available is produced immediately. □

The above lemma is useful to be able to generate problems with feasible solutions in the experimental section.

Notice that the feasibility check of the CSILSP is achieved in $O(T)$ while that of CSILSP-NCS and CSILSP-CS requires $O(T^2)$ iterations to be achieved.

3 Specificities of the capacitated single item problems

A direct approach for solving the CSILSP using lagrangian relaxation is to take existing lagrangian relaxation heuristics implemented for the capacitated multi-item lot sizing problem (ex. Thizy and Van Wassenhove 1986; Trigeiro et al. 1989) and set the number of items (N) equal to 1. After implementing different Lagrangian heuristics for problems with and without time windows (Brahimi et al. 2006b) and with and without setup times (Brahimi et al. 2010b), we have realized that the solution quality was poor (larger gaps between upper and lower bounds) as the number of items gets smaller and the worst results were obtained for $N = 1$. We closely studied the specific properties of the single item problems and considered them in the Lagrangian heuristic, which lead to considerable improvements. These properties are presented below.

Property 1 **Capacity constraints**

In the CSILSP with or without time windows, capacity constraints can be expressed as Constraints (6) or (7), since, either there is no production in period t ($X_t = Y_t = 0$, thus $X_t \leq C_t$) or, if $Y_t = 1$ and $X_t > 0$, then X_t should not exceed available capacity. In multi-item lot sizing problems, we can only use Constraints (6) (augmented with the index of the items). Note that we can expect better results using Constraints (7) as they limit the feasible region. We will show in the experimental results that the relaxation of Constraints (7) yields much better lower bounds than the relaxation of Constraints (6). In the literature, authors use both Constraints (6) (e.g. Van Hoesel and Wagelmans 2001) and Constraints (7) (e.g. Akbalik and Pochet 2009). But, to the best of our knowledge, Constraints (7) have never been exploited in a heuristic before.

Property 2 **Possibility of pre-smoothing demands**

There exists an equivalent problem to CSILSP with or without time windows where:

$$\sum_{k=1}^{t} d_{k,t} \leq C_t \quad \forall t \tag{22}$$

The left hand side of the inequality corresponds to the total demand that must be delivered at the end of period t. This inequality is usually considered as an assumption in some CSILSP papers (e.g. Wolsey 2006). However, in practice, demands can exceed capacities in some periods, while the problem is still feasible. The problem can be solved with these demands or demands can be pre-processed to meet Condition (22). To the best of our knowledge, this is the first attempt to analyze the impact of this processing on the performance of some heuristics for the CSILSP.

The preprocessing of demands, which we called demand pre-smoothing, is done as follows. Starting at the end of the planning horizon, shift the demand which exceeds capacity from the current period to the preceding period. This means that extra demand at the current period is necessarily produced (and stocked) in an earlier period. The resulting holding cost should be considered and added to the objective function of the new problem. Go on shifting quantities until the beginning of the planning horizon (see Sect. 3.1). The problem is infeasible if, at the end of the procedure, we find that the aggregate demand of the first period exceeds capacity.

Property 3 **Possibility of pre-smoothing availabilities**

There exists an equivalent problem to CSILSP where:

$$\sum_{k=t}^{T} d_{t,k} \leq C_t \quad \forall t \tag{23}$$

where the left hand side corresponds to the total demand becoming available at period t.

Though the assumption that demands are smaller than capacities is relevant and relatively easy to implement, Condition (23) is less obvious and its consideration leads to major changes to the problem. As mentioned in Sect. 2.1.2, the single item lot sizing problem without time windows is a special case of the single item problem with non-customer specific time windows where all demands are available in the beginning of the horizon (period $t = 1$). By pre-smoothing availabilities of this problem, demands available in $t = 1$ exceeding capacity, are shifted to the next period. If the updated available demand in $t = 2$ exceeds capacity, then the extra availability is shifted to the next period, and so on. For the CSILSP, this will force some demands to be available after $t = 1$, transforming the problem into a general CSISLP with non-customer specific time windows. The same procedure (See Sect. 3.2) is applied to the CSILSP-NCS and CSILSP-CS which will transform their demands $d_{t1,t2}$.

The two following sections detail the procedure to create equivalent problems of the CSILSP with and without time windows based on Properties 2 and 3.

3.1 Transformation of aggregate demands

Let GD_t be the total demand due in t, i.e. $GD_t = \sum_{k=1}^{t} d_{k,t}$. Because in a feasible solution, $X_t \leq C_t$, if $GD_t > C_t$, we know that at least the amount $GD_t - C_t$ must be produced and stocked before t. Thus, starting at the last period, if $GD_t > C_t$ then consider that GD_t is equal to C_t and increase GD_{t-1} by $GD_t - C_t$. This is done by shifting to $t - 1$ time window demands due at t in the increasing order of their availability periods. At each shifting from t to $t - 1$, the difference between the total cost value of the new problem (TC') and that of the previous problem (TC) is $TC - TC' = (GD_t - C_t) \times h_{t-1}$. The difference ($\Delta CH$) between the costs of the original problem and the last resulting problem is added at the end to any feasible solution of the modified problem. This cost difference is calculated using Algorithm 1 where the aggregate demands are stored in a temporary variable TGD_t which is only used to calculate the cost difference ΔCH. Instead of TGD_t, Algorithm 2 modifies the original aggregate demand GD_t and makes the necessary transformations on demands. The procedure preserves the structure of the time windows of the problem whether they are customer specific or non-customer specific.

It is possible to reduce the running time of Algorithm 2 in the case of infeasible problems by modifying the while loop on line 5. However, we noticed that this phase of the solution approach takes a negligible amount of time compared to the Lagrangian heuristic (Sect. 4).

3.2 Transformation of available demands

Let VD_t be the available demand at period t, i.e. $VD_t = \sum_{k=t}^{T} d_{t,k}$. Since, in a feasible solution, $X_t \leq C_t$, if $VD_t > C_t$, one can consider that the actual available demand which can possibly be produced in t is $VD'_t = C_t$ and add the difference $VD_t - C_t$ to the available demand in the next period. That is, the new available demand in period $t + 1$ becomes $VD'_{t+1} = VD_{t+1} + (VD_t - C_t)$. This is achieved by shifting to $t + 1$ time window demands available at t (i.e. $d_{t,k(k \geq t)}$) in decreasing order of their due

Algorithm 1 Calculating Cost Difference

Initially $\Delta CH := 0$;
Let $TGD_t = \sum_{k=1}^{t} d_{k,t}, \forall t$;
for $t := T$ to 1 **do**
  if $C_t < TGD_t$ **then**
    $TGD_{t-1} := TGD_t + (TGD_t - C_t)$;
    $\Delta CH := \Delta CH + (TGD_t - C_t) \times h_{t-1}$;
    $TGD_t := C_t$;
  end if
end for

Algorithm 2 Smoothing Aggregate Demands

Initially $t2 := T$;
Let ΔD be a real variable;
while $t2 \geq 1$ **do**
  $t1 := 1$;
  while $GD_{t2} > C_{t2}$ **and** $t1 \leq T$ **do**
    if $d_{t1,t2} = 0$ **then**
      $t1 := t1 + 1$;
    else
      $\Delta D := \min(GD_{t2} - C_{t2}, d_{t1,t2})$;
      $d_{t1,t2-1} := d_{t1,t2-1} + \Delta D$;
      $GD_{t2-1} := GD_{t2-1} + \Delta D$;
      $GD_{t2} := GD_{t2} - \Delta D$;
      $d_{t1,t2} := d_{t1,t2} - \Delta D$;
      if $d_{t1,t2} = 0$ **then**
        $t1 := t1 + 1$;
      end if
    end if
  end while
  $t1 := t2 - 1$;
end while

dates. These demands have the same due periods but their release periods are increased by one.

Note that this is the same procedure applied to CSILSP, CSILSP-NCS, and CSILSP-CS. In the case of CSILSP, its demands $d_{1,1}, d_{1,2}, \ldots, d_{1,T}$ will be transformed into new time window demands $d'_{t1,t2}$ which are non-inclusive (non-customer specific case) and satisfying the property:

$$\sum_{k=1}^{t} d'_{k,t} = d_{1,t} \quad \forall t \tag{24}$$

The procedure for pre-smoothing available demand can considerably improve the efficiency of the algorithms for solving the classical capacitated single item problems without time windows, which are transformed in this case to equivalent CSILSP-NCS problems.

The pseudo code of Algorithm 3 summarizes the procedure.

Algorithm 3 Smoothing Available Demands

```
Initially t1 := 1;
Let ΔD a real variable.
while t1 ≤ T do
  t2 := T;
  while VD_{t1} > C_{t1} and t2 ≥ 1 do
    if d_{t1,t2} = 0 then
      t2 := t2 − 1;
    else
      ΔD := min (VD_{t1} − C_{t1}, d_{t1,t2});
      d_{t1+1,t2} := d_{t1+1,t2} + ΔD;
      VD_{t1+1} := VD_{t1+1} + ΔD;
      VD_{t1} := VD_{t1} − ΔD;
      d_{t1,t2} := d_{t1,t2} − ΔD;
      if d_{t1,t2} = 0 then
        t2 := t2 − 1;
      end if
    end if
  end while
  t1 := t1 + 1;
end while
```

Table 1 First modification of available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	37	**16**	73	**124**
	3	0	0	0	0	0	**0**	23	0
	4	0	0	0	6	25	10	120	41
	5	0	0	0	0	23	14	25	37
	6	0	0	0	0	0	20	41	20
Sum		0	22	38	17	107	60		

Table 2 Second modification of available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	**37**	0	73	**108**
	3	0	0	0	0	**0**	16	23	16
	4	0	0	0	6	25	10	120	41
	5	0	0	0	0	23	14	25	37
	6	0	0	0	0	0	20	41	20
Sum		0	22	38	17	107	60		

3.3 An illustrative example

The example in Tables 1, 2, 3, 4, 5 and 6 illustrates the algorithm. In Table 1, the first available demand exceeding capacity is $VD_2 = 124$ ($C_2 = 73$). Demand $d_{2,6} = 16$

Table 3 Third modification of available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	2	0	73	73
	3	0	0	0	0	35	**16**	23	**51**
	4	0	0	0	6	25	**10**	120	41
	5	0	0	0	0	23	14	25	37
	6	0	0	0	0	0	20	41	20
Sum		0	22	38	17	107	60		

Table 4 Fourth modification of available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	2	0	73	73
	3	0	0	0	0	**35**	0	23	**35**
	4	0	0	0	6	**25**	26	120	57
	5	0	0	0	0	23	14	25	37
	6	0	0	0	0	0	20	41	20
Sum		0	22	38	17	107	60		

Table 5 Fifth modification of available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	2	0	73	73
	3	0	0	0	0	23	0	23	23
	4	0	0	0	6	37	26	120	69
	5	0	0	0	0	23	**14**	25	**37**
	6	0	0	0	0	0	**20**	41	20
Sum		0	22	38	17	107	60		

is shifted so that it becomes available at period 3 (Table 2). In Table 2, VD_2 is still larger than C_2. $\min\left[d_{2,5}, (VD_2 - C_2)\right] = 35$ is shifted so that it becomes available at period 3 (Table 3). In Table 3, $VD_3 > C_3$, and $d_{3,6}$ is shifted to period 4 (Table 4). The process is repeated in Tables 4 and 5. The final result can be found in Table 6.

4 Lagrangian heuristic

The Lagrangian relaxations are based on the aggregate formulation of the problems. The relaxations are applied with and without pres-smoothing to see the impact of these procedures on the quality of the solutions and lower bounds. For the CSILSP-

Table 6 Final available demands

		Demand periods						C_t	VD_t
		1	2	3	4	5	6		
Availability periods	1	0	0	0	0	22	0	137	22
	2	0	22	38	11	2	0	73	73
	3	0	0	0	0	23	0	23	23
	4	0	0	0	6	37	26	120	69
	5	0	0	0	0	23	2	25	25
	6	0	0	0	0	0	32	41	32
Sum		0	22	38	17	107	60		

CS problems, we relax time window constraints (16) and either capacity constraints (6) or (7). For the CSILSP and the CSILSP-NCS problems, we relax either capacity constraints (6) or (7). The relaxations always result in uncapacitated single item lot sizing problems with non-customer specific time windows represented by Constraints (1)–(5) and (15), which are solved using the algorithm of Wolsey (2006) which runs in $O(T^2)$.

In these relaxations, we associate the Lagrangian vectors ν and π to time window constraints (16) and capacity constraints (Constraints (6) or (7)), respectively.

Then, we consider the general scheme of the Lagrangian relaxation (see e.g. Parker and Rardin 1988) presented in Algorithm 4 for the sake of completeness.

Algorithm 4 Lagrangian Relaxation Heuristic

Step 1: **Initialization.**
  a. Initialize all multipliers to 0.
  b. Set iteration number $k = 1$.
  c. Initialize step length, λ.
  d. Initialize lower bound $LB = -\infty$.
Step 2: **Solving the relaxed problem.** Solve Lagrangian Problem $L(\nu, \pi)^k$ for current values of multipliers ν, π and calculate current lower bound LB^k (Section 4.1).
Step 3: **Incumbent saving.** If $LB < LB^k$, then $LB := LB^k$
Step 4: **Smoothing heuristic.** Use the values of X, I and Y obtained in *Step 2* to find a feasible solution using a heuristic procedure (Section 4.3).
Step 5: **Updating multipliers.** Lagrangian multipliers are updated using the subgradient optimization method (Section 4.4).
Step 6: **Stopping conditions.** If any stopping condition is met, save the best solution obtained so far and stop.
Step 7: **Update step length.** Update λ as shown in Section 4.4.
Step 8: Increment k and go to Step 2.

4.1 The relaxed problem

The relaxed problem, corresponding to an uncapacitated single item problem with non-customer specific time windows, is solved using the algorithm of Wolsey (2006) which runs in $O(T^2)$ time.

The relaxed problem is:

$$\text{Minimize } COST^L = \sum_{t=1}^{T}\left[h_t I_t + \left(p_t + \pi_t - \sum_{k=1}^{t}\sum_{l=t}^{T} v_{kl}\right) X_t + (s_t - \pi_t C_t) Y_t\right] + C,$$

subject to Constraints (1)–(5) and (15)
if Constraints (7) are relaxed, and

$$\text{Minimize } COST^L = \sum_{t=1}^{T}\left[h_t I_t + \left(p_t + \pi_t - \sum_{k=1}^{t}\sum_{l=t}^{T} v_{kl}\right) X_t + s_t Y_t\right] + C,$$

Subject to Constraints (1)–(5) and (15)
if Constraints (6) are relaxed.
In the latter case, the constant C in the objective is

$$C = \sum_{t1=1}^{T}\sum_{t2=t1}^{T} v_{t1,t2} \sum_{k=t1}^{t2}\sum_{l=k}^{t2} d_{kl} - \sum_{t=1}^{T} \pi_t C_t$$

while C is

$$C = \sum_{t1=1}^{T}\sum_{t2=t1}^{T} v_{t1,t2} \sum_{k=t1}^{t2}\sum_{l=k}^{t2} d_{kl}$$

when Constraints (7) are relaxed.

Note that if the original problem is of NCS type, the terms $\sum_{t1=1}^{T}\sum_{t2=t1}^{T} v_{t1,t2} \sum_{k=t1}^{t2}\sum_{l=k}^{t2} d_{kl}$ and $\sum_{k=1}^{t}\sum_{l=t}^{T} v_{kl}$ will be equal to zero in the objective function.

4.2 Initial feasible solution

Pre-smoothing of aggregate demands automatically generates a feasible solution. This is why, even if we do not apply pre-smoothing (in order to analyze its impact), we use it initially to build a first feasible solution.

4.3 Smoothing heuristic

To construct a feasible solution, unsatisfied relaxed constraints are detected. In the relaxation of customer specific problems, we first satisfy time window constraints, then capacity constraints. Time window constraints (16) are satisfied using the procedure STWPE presented in details in our paper Brahimi et al. (2006a). Capacity constraints are satisfied using a modified version of the Trigeiro-Thomas-MacClain (TTM) smoothing heuristic (Trigeiro et al. 1989). The modified TTM heuristic is applied in such a way that time window constraints are not violated while shifting extra quantities backward and forward. More details about the combination of the TTM heuristic with the STWPE heuristic can be found in Brahimi et al. (2006a).

4.4 Updating step size

The Lagrangian multipliers are updated using the subgradient method. The step size λ^k of iteration k is calculated using the formula:

$$\lambda^k = \frac{\delta_k(UB^* - LB^k)}{\left\|\sigma^k\right\|}$$

where σ^k is the subgradient at iteration k for the current vector of Lagrangian multipliers and $\left\|\sigma^k\right\|$ is its magnitude. UB^* is the value of the best feasible solution and LB^k is the current lower bound. The scalar δ_k is initially equal to 2. Frequently, the sequence δ_k is determined by starting with $\delta_0 = 2$ and δ_k is reduced by a factor of 2 whenever LB^k has failed to increase in a specified number of iterations (see Held et al. 1974). A better approach which proved very efficient with our heuristic consists of implementing learning capabilities into the step size by allowing δ_k to increase if there is improvement on the lower bound and decrease it if the lower bound fails to increase as suggested in Zamani and Lau (2010). Thus, if there is no improvement on the lower bound, $\delta_{k+1} = \delta_k \times U1$, and $\delta_{k+1} = \delta_k \times U2$ if the lower bound is improved. The relationship between U1 and U2 is defined by Zamani and Lau (2010) using the formula: $U2 = (3 - U1)/2$ and $0 \leq U1 \leq 1.0$.

5 Numerical results

All numerical tests were carried out on a personal computer with Intel Core i7-2.4 Ghz CPU with 16 GB RAM. The FAL models were implemented on FICO Xpress solver FICO 2013. The Lagrangian heuristics were implemented using the C++ programming language. The parameters U1 and U2 defined in Sect. 4.4 are set to 0.9 and 1.05, respectively. The Lagrangian heuristic was stopped after 500 iterations.

The generated data sets are for planning horizons of $T = 24$, 50, 100, and 300 periods. A generated instance is accepted if the feasibility condition presented in Lemma 1 is satisfied. The other parameters are presented below.

Capacity tightness. A lot-for-lot schedule was considered and capacity utilization was averaged over all periods. The resulting value was multiplied by a capacity tightness factor (**CDF**) taking values 1.05 (very tight), 1.2 (normal), and 2.0 (loose) to obtain the capacity at each period.

Setup/holding cost factor (SH). The holding cost was generated from a uniform distribution in [3, 4]. The setup cost was generated from a uniform distribution in

[$3 \times$ SH, $4 * $SH], where SH was fixed to 200 and 400. The ratio SH also reflects the time between orders (TBO) such that TBO is high for SH = 400 and low for SH = 200.

For the CSILSP-CS, time window densities and minimum time window lengths were also considered.

Time window density (TWD) for the CSILSP-CS. For a given horizon length T, the maximum number of time window demands is $T(T+1)/2$. The time window density (TWD) is the ratio of the number of positive time window demands to the maximum number of time window demands. It takes the values 0.3, 0.5, and 0.7. This parameter is particularly important as it affects the size of the problems and the linear programming formulations.

Minimum time window length (MINL). For a given demand $d_{t1,t2}$, the time window length $(t2-t1+1)$ can vary between a minimum length LMIN and a maximum length LMAX. If LMIN $= 1$, that is, there are several periods t where $d_{tt} > 0$, then these periods must be production periods and setup variables are already set to 1 for them, which makes the problem very easy to solve. Furthermore, we are supposing that there is a reasonable number of periods between the availability of demands and their due dates. Thus, LMIN is given the values 4, 8, and 12 and LMAX is always equal to T.

Finally, the production cost is generated from a uniform distribution in [16, 20].

The data sets were generated in a full factorial manner. For each combination of parameters, five instances were generated randomly. This results in 120 instances for CSILSP and CSILSP-NCS and 1080 instances for CSILSP-CS.

The gap between the lower bounds (LB) and upper bounds (UB) was calculated using the formula:

$$\text{Gap} = 100 \times \frac{\text{UB-LB}}{\text{UB}.} \tag{25}$$

In the following, we show the effect of considering the properties of the capacitated single item lot sizing problem. The results are compared to situations where we ignore these properties. The results are also compared with the solutions and performance of the commercial solver FICO-Xpress.

5.1 Analysis of the impact on lower bounds and upper bounds of the Lagrangian heuristics

This section presents the first phase of the numerical tests. It is limited to problems with up to 100 time periods. FICO-Xpress was able to find the optimal solutions in less than 1 Second on average for problems with $T = 24$ and $T = 50$ periods, and in less than 20 s on average when $T = 100$, though some problems required more than 600 s. Problems with $T = 300$ periods were difficult to solve for the solver. FICO-Xpress failed to find any feasible solutions for several problems after more than 60 s of CPU time for CSILSP-CS. This is why Sect. 5.2 focuses on the analysis of large size problems and compares the performance of the Lagrangian heuristic with that of the solver.

Table 7 CPU time of Lagrangian heuristic (seconds)

Horizon (T)	CPU time (s)
24	0.02
50	0.06
100	0.24
300	2.39

The average CPU times of the Lagrangian heuristic are less than 0.3 s for $T \leq 100$ and 2.39 s for $T = 300$ periods as shown on Table 7.

For problems with $T \leq 100$, after obtaining their optimal solutions using FICO-Xpress, the Lagrangian heuristic upper bounds and lower bounds were compared with these optimal solutions. In Tables 8, 9, and 10, $Gap^{opt-LB} = 100 \times (opt - LB)/opt$ is the gap between the optimal solution (opt) and the best lower bound (LB) of the Lagrangian relaxation before and after considering the properties of the single item problems. Similarly, $Gap^{UB-opt} = 100 \times (UB - opt)/opt$ is the gap between the optimum and the best feasible solution of the Lagrangian heuristic. The first and second columns in Tables 8, 9 and 10 show the type of parameter and its values. In problems without time windows (Table 8) and with NCS time windows (Table 9), the considered parameters are the horizon length (T), the capacity tightness (CDF), and the setup-over-holding cost factor (SH). The results for problems with CS time windows consider the time window density (TWD) and the minimum time window length (MINL). Column titled "Basic" shows Gap^{opt-LB} and Gap^{UB-opt} without considering pre-smoothing or the new Constraints (7) (i.e. Constraints (6) are considered instead). Columns "Constraint (7)" and "pre-smoothing" present the gaps while considering the new properties individually. Finally, column "Both" shows the gaps when both properties are combined. It is important to note that the "Basic" case corresponds to the direct execution of CLSP Lagrangian heuristics TTM (in case there are no time windows) and the combination of STWPE and TTM (if time windows are considered).

5.1.1 CSILSP without time windows

For the classical capacitated single item lot sizing problem, the average gap of the lower bound to optimality improves from 26.22 to 1.39 % after considering single item properties (an improvement factor of 18). The average Gap^{UB-opt}, showing the quality of feasible solutions, is improved from 2.25 to 0.49 %. This indicates that the smoothing heuristic is effective and that the new properties impact the quality of the Lagrangian relaxation lower bounds.The largest impact comes from the addition of Constraints (7) which improves Gap^{opt-LB} from 26.22 to 10.34 % (instead of 16.69 % by only introducing pre-smoothing). However, results are strongly improved when both pre-smoothing and Constraints (7) are considered. The impact of the new properties is almost the same for small or large horizon problems. When capacity is tight (small CDF), the impact if very important. When CDF =1.05, Gap^{opt-LB} improves from more than 35–0.85 % (a factor of 42). This impact decreases with the loosest capacities and goes down to an improvement factor of 5 for CDF = 2.0. As SH

Table 8 Impact on classical problems without time windows

Param	Value	Gap^{opt-LB}				Gap^{UB-opt}			
		Basic	Constraints (7)	Pre-smoothing	Both	Basic	Constraints (7)	Pre-smoothing	Both
T	24	26.64	9.29	17.90	1.62	2.34	1.33	2.33	0.38
	50	24.78	11.72	17.46	1.42	2.36	1.44	2.36	0.39
	100	27.24	10.01	14.70	1.13	2.04	1.61	2.04	0.70
CDF	1.05	35.76	17.33	20.21	0.85	0.35	0.35	0.57	0.12
	1.2	31.50	10.94	15.00	1.72	1.81	1.77	3.42	0.58
	2	11.40	2.74	9.36	2.23	4.57	2.26	4.24	1.62
SH	200	21.24	7.60	13.91	1.29	1.55	0.88	1.89	0.39
	400	31.20	13.07	22.25	1.59	2.94	2.04	2.95	0.70
Average		26.22	10.34	16.69	1.39	2.25	1.46	2.24	0.49

Table 9 Impact on NCS problems

Param	Value	Gap^{opt-LB}				Gap^{UB-opt}			
		Basic	Constraints (7)	Pre-smoothing	Both	Basic	Constraints (7)	Pre-smoothing	Both
T	24	25.64	4.35	17.90	1.68	2.31	0.63	2.33	0.35
	50	23.95	4.02	17.46	1.45	2.36	1.21	2.36	0.37
	100	25.21	7.04	14.70	1.17	2.02	1.25	2.02	0.68
CDF	1.05	33.49	6.83	19.90	0.75	0.35	0.17	0.35	0.05
	1.2	29.99	5.61	19.47	1.08	1.79	1.56	1.81	0.26
	2	11.32	2.97	10.69	2.46	4.55	1.36	4.55	1.10
SH	200	19.55	5.57	10.59	1.10	1.55	0.87	1.55	0.39
	400	30.32	4.71	22.78	1.76	2.92	1.20	2.93	0.55
Average		24.93	5.14	16.69	1.43	2.23	1.03	2.24	0.47

Table 10 Impact on CS problems

Param	Value	Gap^{opt-LB}				Gap^{UB-opt}			
		Basic	Constraints (7)	Pre-smoothing	Both	Basic	Constraints (7)	Pre-smoothing	Both
T	24	34.30	4.19	21.20	1.41	1.45	0.66	1.46	0.23
	50	30.38	7.87	17.89	1.22	2.23	1.02	2.23	0.32
	100	29.10	10.05	14.79	1.04	1.95	1.15	1.95	0.58
CDF	1.2	39.15	10.54	19.76	0.69	0.57	0.50	0.58	0.10
	5	26.17	5.19	16.73	1.58	2.67	1.18	2.67	0.58
	10	20.08	3.38	15.69	1.95	3.94	1.76	3.94	0.63
SH	200	27.41	7.18	14.32	1.15	1.67	0.84	1.67	0.35
	400	38.96	7.73	25.24	1.38	2.30	1.14	2.30	0.42
TWD	0.3	29.52	6.86	17.13	1.23	1.90	0.86	1.91	0.37
	0.5	31.49	7.50	18.21	1.25	1.89	1.03	1.89	0.40
	0.7	37.30	8.87	20.31	1.11	1.73	0.95	1.75	0.28
MINL	4	28.18	5.62	17.54	1.39	2.10	0.99	2.10	0.38
	8	34.00	7.98	19.10	1.13	1.66	0.90	1.67	0.36
	12	32.60	11.92	15.06	0.97	1.83	0.93	1.83	0.41
Average		31.26	7.37	17.96	1.22	1.88	0.94	1.88	0.37

Table 11 Gaps for $T = 300$: case without time windows

Param	Value	Gap^L	Gap^{X10}
CDF	1.05	0.07	11.62
	1.2	0.96	0.41
	2	4.38	1.01
SH	200	1.31	2.70
	400	2.30	5.99
Average		1.8	4.35

increases, the impact of adding Constraints (7) becomes more important. The impact of pre-smoothing is more important for SH = 200 than SH = 400.

Considering the general performance of the Lagrangian heuristic, the lower bound quality considerably increases as the capacity increases. The lower bound also improves as the horizon length increases and as the SH factor (and consequently the time between orders) decreases.

5.1.2 Problems with NCS time windows

The impact of adding Constraints (7) is more important with NCS problems, especially in the case of tight capacities (CDF = 1.05), in which the gain factor is almost 5 compared to a gain factor of 2 in the case of problems without time windows. The contribution of the pre-smoothing procedure is also substantial though less important compared to no-time windows case. The contribution of pre-smoothing is less important here because the structure of the problem does not change. The problem is non customer specific and it remains so after pre-smoothing.

The performance of the new Lagrangian heuristic shows the same behavior as the case without time windows. However, the negative impact of an increased TBO is more visible in this case.

5.1.3 Problems with CS time windows

The improvement brought by relaxing Constraints (7) instead of Constraints (6) is significant (cf. Table 10). With respect to time between orders (TBO) represented by SH ratio, Constraints (7) make the Lagrangian lower bound more stable and less sensitive to the increase of SH.

The effect of considering pre-smoothing on aggregate and available demands is substantial though less important than when Constraints (7) are used. After introducing pre-smoothing, the lower bound presents an opposite behavior with the variation of the parameters (except for CDF and SH) compared to the use of Constraints (7). For example, as T increases, the gap increases when Constraints (7) are considered and decreases if pre-smoothing is used.

As expected, the contribution of the pre-smoothing procedures is much more significant with tight capacity problems. In very tight capacity problems, generally, setups are required in most periods. Thus what remains to attain the optimal solution are few shifts of production quantities based on the best production/holding cost combinations.

Table 12 Gaps for $T = 300$: case with NCS time windows

Param	Value	Gap^L	Gap^{X10}
CDF	1.05	0.07	5.94
	1.2	1.02	0.22
	2	4.74	1.00
SH	200	1.29	2.01
	400	2.59	2.76
Average		1.94	2.39

Table 13 Gaps for $T = 300$: case with CS time windows

Param	Value	Gap^L	Gap^{X10}	X-Feasible (%)
CDF	1.05	0.10	-	0.00
	1.2	1.04	3.29	13.33
	2	5.05	4.22	44.44
SH	200	1.35	3.31	17.69
	400	2.71	4.33	21.48
TWD	0.3	2.37	3.09	26.67
	0.5	1.97	5.66	23.33
	0.7	1.85	2.94	7.78
MINL	4	2.16	4.57	20.00
	8	2.06	3.87	18.89
	12	1.97	3.11	18.89
Average		2.06	4.15	19.26

In case of very large capacities, the problems become easy even without smoothing, since relaxing capacity will have little effect on the feasibility of the original problem; we are close to the uncapacitated case.

5.2 Lagrangian heuristic performance on large size problems

This section compares the performance of the Lagrangian heuristic with that of the commercial solver for problems with large planning horizons ($T = 300$ periods). Tables 11, 12 and 13 summarize these results. The Lagrangian gaps (between upper and lower bounds) are shown in columns Gap^L. Since the average CPU time of the Lagrangian relaxation was 2.39 s with a maximum of 3 s, FICO-Xpress solver was run for 10 s and the corresponding integrality gaps are given in columns Gap^{X10}. Note that the limitation of the solver to 10 s is based on the fact that we want to develop fast heuristics for the CSILSP. Nevertheless, we also ran FICO-Xpress for 2 minutes and the gaps were still larger than 1% in many instances and as large as 4% in some cases. Gap^L and Gap^{X10} are calculated using (25), in which UB and LB are replaced by the upper and lower bounds of the lagrangian heuristic and the solver, respectively.

On average, the Lagrangian heuristic outperforms the solver for all problems. Looking at the detailed results of CSILSP and CSILSP-NCS, while Xpress performs very poorly with tight capacities, the Lagrangian heuristic gaps are excellent (0.07%). This is due to the impact of demand pre-smoothing on tight capacities. On the other hand, the solver and particularly the facility location-based formulation, performs better with looser capacities. This is because problems are getting closer to uncapacitated problems.

The performance of the solver is poorer for CSILSP-CS. The last column in Table 13 (column "X-Feasible") shows the percentage of problems for which Xpress managed to find a feasible solution in 10 s. No feasible solution was found in 10 s for any problem with CDF = 1.05, and feasible solutions were found only for 13.33 and 44.44 % of problems with CDF = 1.2 and CDF = 2.0, respectively. Other hard problems for Xpress to solve are those with larger time window densities (TWD) as only 7.78 % of problems with TWD = 0.7 were solved. The Lagrangian heuristic always finds a feasible solution.

6 Conclusion

In this paper, we adapted a Lagrangian heuristic which was initially developed for multi-item lot sizing problems to solve capacitated single item lot sizing problems with time windows. The heuristic can solve both CSILSPs with and without time windows. We showed the importance of considering specific properties of capacitated single item problems instead of only setting the number of items equal to 1 in the heuristic. We presented the possible transformations of demands for problems with and without time windows to create tighter equivalent problems. The modifications required on algorithms for multi item problems were presented. The results showed significant improvements for problems without time windows and problems with non customer specific and customer specific time windows (with factors up to 18).

The efficiency and speed of the Lagrangian heuristic suggest that it can be used as a cornerstone in heuristics to solve more complex lot sizing problems. This can be achieved, for example, in a Lagrangian decomposition heuristic for capacitated multi-item, multi-level problems.

Acknowledgments The authors would like to express their gratitude to the anonymous reviewers for their very helpful comments which considerably contributed to the improvement of the quality of this paper.

References

Akbalik A, Pochet Y (2009) Valid inequalities for the single-item capacitated lot sizing problem with step-wise costs. Eur J Oper Res 198:412–434

Billington P, McClain J, Thomas L (1986) Heuristics for multilevel lot-sizing with a bottleneck. Manag Sci 32:989–1006

Bitran G, Yanasse H (1982) Computational complexity of the capacitated lot size problem. Manag Sci 28:1174–1186

Bitran GR, Magnanti TL, Yanasse HH (1984) Approximation methods for the uncapacitated dynamic lot size problem. Manag Sci 30:1121–1140

Brahimi N, Dauzère-Pérès S, Najid N (2006a) Capacitated multi-multi item lot sizing problems with time windows. Oper Res 54:951–967

Brahimi N, Dauzère-Pérès S, Najid N, Nordli A (2006b) Single item lot sizing problems. Eur J Oper Res 168:1–16

Brahimi N, Absi N, Dauzère-Pérès S, Kedad-Sidhoum S (2010a) Mathematical models and lagrangian heuristics for a two-level lot-sizing problem with bounded inventory. 8ème Conférence Francophone de Modélisation et Simulation, MOSIM, Hammamet, Tunisia

Brahimi N, Dauzère-Pérès S, Wolsey LA (2010b) Polyhedral and lagrangian approaches for lot sizing with production time windows and setup times. Comput Oper Res 37:182–188

Buschkühl L, Sahling F, Helber S, Tempelmeier H (2010) Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. OR Spectr 32:231–261

Chen H-D, Hearn DW, Lee C-Y (1994) A new dynamic programming algorithm for the single item capacitated dynamic lot size model. J Glob Optim 4:285–300

Chen W, Thizy J (1990) Analysis of relaxations for the multi-item capacitated lot-sizing problem. Ann Oper Res 26:29–72

Dauzère-Pérès S, Brahimi N, Najid N, Nordli A (2002) The single-item lot sizing problem with time windows. Technical report, 02/4/AUTO, Ecole des Mines de Nantes, France. https://www.researchgate.net/publication/239919300_Uncapacitated_Lot-Sizing_Problems_with_Time_Windows

Dixon P, Silver E (1981) A heuristic solution procedure for the multi-item, single-level, limited-capacity, lot-sizing problem. J Oper Manag 2:23–39

FICO (2013) Xpress-ive. Web: http://www.FICO.com

Florian M, Klein M (1971) Deterministic production planning with concave costs and capacity constraints. Manag Sci 18:12–20

Hardin JR, Nemhauser GL, Savelsbergh MW (2007) Analysis of bounds for a capacitated single-item lot-sizing problem. Comput Oper Res 34:1721–1743

Held M, Worlfe P, Crowder H (1974) Validation of subgradient optimization. Math Program 6:62–88

Hwang H-C (2007) Dynamic lot-sizing model with production time windows. Naval Res Logist 54:692–701

Hwang H-C, Jaruphongsa W (2008) Dynamic lot-sizing model for major and minor demands. Eur J Oper Res 184:711–724

Love S (1973) Bounded production and inventory models with piecewise concave costs. Manag Sci 20(3):313–318

Maes J, Van Wassenhove L (1986) A simple heuristic for the multi-item, single level capacitated lot sizing problem. Oper Res Lett 4:265–273

Merle Od, Goffin J-L, Trouiller C, Vial J-P (2000) Pardalos PM (ed) A lagrangian relaxation of the capacitated multi-item lot sizing problem solved with an interior point cutting plane algorithm. Approximation and complexity in numerical optimization. Springer, US, number 42 in Nonconvex optimization and its applications, pp 380–405

Millar H, Yang M (1994) Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. Int J Prod Econ 34:1–15

Nahmias S (1982) Perishable inventory theory: a review. Oper Res 30:680–708

Ou J (2012) Economic lot sizing with constant capacities and concave inventory costs. Naval Res Logist 59:497–501

Parker R, Rardin R (1988) Discrete optimization. Academic Press, San Diego

Quadt D, Kuhn H (2008) Capacitated lot-sizing with extensions: a review. 4OR, 6, pp 61–83

Saydam C, Evans J (1990) Comparative performance analysis of the wagner-whitin algorithm and lot-sizing heuristics. Comput Indus Eng 18:91–93

Shaw D, Wagelmans A (1998) An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. Manag Sci 44:831–838

Silver E, Meal H (1973) A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. Prod Invent Manag 14:64–74

Simpson V (1978) Optimum solution structure for a repairable inventory problem. Oper Res 26:270–281

Thizy J, van Wassenhove L (1986) A subgradient algorithm for the multi-item capacitated lot-sizing problem. IIE Trans 18:114–123

Thizy J-M, Van Wassenhove L (1985) Lagrangean relaxation for the multi-item capacitated lot-sizing problem: a heuristic implementation. IIE Trans (Institute of Industrial Engineers) 17:308–313

Toledo FMB, Armentano VA (2006) A lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. Eur J Oper Res 175:1070–1083

Trigeiro W, Thomas L, McClain J (1989) Capacitated lot sizing with set-up times. Manag Sci 35:353–366

Vachani R (1992) Performance of heuristics for the uncapacitated lot-size problem. Naval Res Logist (NRL) 39:801–813

van den Heuvel W, Wagelmans AP (2008) Four equivalent lot-sizing models. Oper Res Lett 36:465–470

Van Hoesel C, Wagelmans A (1996) O(t3) algorithm for the economic lot-sizing problem with constant capacities. Manag Sci 42:142

Van Hoesel C, Wagelmans A (2001) Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. Math Oper Res 26:339–357

Wagelmans A, van Hoesel S, Kolen A (1992) Economic lot sizing an o(n log n) algorithm that runs in linear time in the wagner-whitin case. Oper Res 40:145

Wagner H, Whitin T (1958) Dynamic version of the economic lot size model. Manag Sci 5:89–96

Wolsey L (2006) Lot-sizing with production and delivery time windows. Math Program Ser A 107:471–489

Zamani R, Lau SK (2010) Embedding learning capability in lagrangean relaxation: an application to the travelling salesman problem. Eur J Oper Res 201:82–88

Zangwill W (1969) A backlogging model and a multi-echelon model of a dynamic economic lot size production system: a network approach. Manag Sci 15:506–527