

Multi-Item Capacitated Lot-Sizing Problems with Setup Times and Pricing Decisions

Mehmet Önal,¹ H. Edwin Romeijn²

¹ *Innovative Scheduling, Inc., Gainesville Technology Enterprise Center (GTEC), 2153 SE Hawthorne Road, Suite 128, Gainesville, Florida 32641*

² *Department of Industrial and Operations Engineering, The University of Michigan, 1205 Beal Avenue, Ann Arbor, Michigan 48109-2117*

Received 19 November 2008; revised 8 September 2009; accepted 29 October 2009

DOI 10.1002/nav.20394

Published online 16 December 2009 in Wiley InterScience (www.interscience.wiley.com).

Abstract: We study a multi-item capacitated lot-sizing problem with setup times and pricing (CLSTP) over a finite and discrete planning horizon. In this class of problems, the demand for each independent item in each time period is affected by pricing decisions. The corresponding demands are then satisfied through production in a single capacitated facility or from inventory, and the goal is to set prices and determine a production plan that maximizes total profit. In contrast with many traditional lot-sizing problems with fixed demands, we cannot, without loss of generality, restrict ourselves to instances without initial inventories, which greatly complicates the analysis of the CLSTP. We develop two alternative Dantzig–Wolfe decomposition formulations of the problem, and propose to solve their relaxations using column generation and the overall problem using branch-and-price. The associated pricing problem is studied under both dynamic and static pricing strategies. Through a computational study, we analyze both the efficacy of our algorithms and the benefits of allowing item prices to vary over time. © 2009 Wiley Periodicals, Inc. *Naval Research Logistics* 57: 172–187, 2010

Keywords: Capacitated production and inventory planning; pricing; branch-and-price

1. INTRODUCTION

We consider the capacitated lot-sizing problem with setup times and pricing decisions (CLSTP). This problem generalizes the well-known capacitated lot-sizing problem with setup times (CLST) by incorporating pricing decisions for all items, thereby making demand endogenous to the model rather than exogenous. In the CLST, demands for multiple items over a finite and discrete planning horizon are given in advance and they are satisfied by producing in a single common facility with limited available resource time. The problem is to find the production plan that minimizes total costs of production and inventory carriage to satisfy the given demands. In contrast, in the CLSTP demands are not given; however, through pricing decisions, the demand levels to satisfy for each item in each period can effectively be chosen. Since the demand levels influence both revenues and costs, the goal becomes to maximize profit, which is defined to be total revenues minus total cost of satisfying the resulting demands.

Single-item uncapacitated lot-sizing problems were first studied in the seminal work of Wagner and Whitin [17].

Thomas [14] was the first to incorporate pricing decisions in an uncapacitated lot-sizing model, assuming a dynamic pricing strategy (i.e., prices can vary over time). Kunreuther and Schrage [11] developed a heuristic approach to solve this problem for the case, where a constant price should be set for the item over the planning horizon by restricting the form of the demand or revenue functions. Gilbert [7] and van den Heuvel and Wagelmans [9] developed exact algorithms for this problem; these algorithms run in polynomial time under further restricted demand functions. Single-item lot-sizing problems with finite but stationary production capacities under constant and dynamic pricing strategies were studied by Geunes et al. [5, 6]. Finally, Gilbert [8] considered a multiproduct planning problem with constant-priced goods that share procurement capacity under linear procurement cost functions.

Much research has been done to reduce the integrality gap of a traditional formulation of the CLST using continuous production and inventory, and binary setup variables. Lagrangean relaxation and Dantzig–Wolfe decomposition are two of the techniques that have been studied to find improved lower bounds. The two methods are equivalent in that one is the dual of the other and they both exploit the structure of the problem such that when the capacity

Correspondence to: H.E. Romeijn (romeijn@umich.edu)

constraints, which tie the different items, are removed, we are left with independent single item lot-size problems. Trigeiro et al. [15] and Hindi et al. [10] contain two examples of heuristic solution techniques that rely on the Lagrangean relaxation and subgradient optimization of the problem obtained by relaxing the capacity constraints. An alternative, and more successful, approach for this problem is Dantzig–Wolfe decomposition. Early attempts were made by Manne [13] and Dzielinski and Gomory [4]. Although the relaxation of their formulation provides a lower bound to the optimal solution value of the CLST, the corresponding integer programming formulation is not equivalent to the CLST in the sense that they may yield solutions that are suboptimal for the CLST. Recently, however, Degraeve and Jans [3] developed a correct Dantzig–Wolfe formulation of the CLST along with a corresponding branch-and-price algorithm.

In this article, we extend the CLST to account for pricing opportunities, and develop two alternative Dantzig–Wolfe decomposition formulations for the CLSTP along with corresponding branch-and-price algorithms. The column generation approach that is used to solve a relaxation of the problem formulation at each node of the branch-and-bound tree requires the solution of a pricing problem, which is shown to decompose into appropriate uncapacitated single-item lot-sizing problems with pricing decisions. As mentioned earlier, efficient algorithms to solve such problems to optimality exist in the absence of initial inventories. However, they become more challenging when initial inventories are present, and we develop effective polynomial-time algorithms for these problems that may also be of independent interest for solving uncapacitated single-item lot-sizing problems with pricing decisions.

The remainder of the article is organized as follows. In Section 2, we state our assumptions and present the formulation of the CLSTP. In Section 3, we develop our two Dantzig–Wolfe formulations and the associated column generation methods. In Section 4, we discuss algorithms for solving the resulting pricing problems under dynamic and static pricing strategies. In Section 5, we provide implementation details of our branch-and-price algorithm. We present computational results in Section 6 and conclude the article in Section 7.

2. THE CLSTP MODEL AND ASSUMPTIONS

We will formally describe the CLSTP and review earlier work on this problem. Assume there are N items whose (price-dependent) demands should be satisfied over a planning horizon of T periods. The amount of resource time available in the production facility in period t is C_t . (In the remainder of this article, we will assume that $C_t > 0$ for $t = 1, \dots, T$. If this is not the case, we reduce the model

by eliminating the corresponding setup and production variables and capacity constraints from the model.) Production for an item i in period t can only take place after a setup is performed, which incurs a fixed cost S_{it} and consumes a fixed amount of the available resource time b_{it} . In addition, for each unit item produced a variable production cost c_{it} is incurred and variable resource time a_{it} is consumed. Let x_{it} be the amount of production for item i in period t . Moreover, let the setup indicator variable y_{it} equal 1 if a setup takes place for item i in period t and 0 otherwise. Inventory for item i at the end of period t , I_{it} , is carried to the next period incurring a cost of h_{it} per unit carried, and no back-orders are allowed.

Let p_{it} be the price set for item i in period t and suppose that demand d_{it} changes with price p_{it} according to the relation $d_{it} = D_{it}(p_{it}) = \alpha_{it} - \beta_{it}\Pi_{it}(p_{it})$, where α_{it} and $\beta_{it} > 0$ are parameters and $\pi_{it} \equiv \Pi_{it}(p_{it})$ is a price-dependent parameter that is called the *price effect* on demand. Note that we could of course simply set $\alpha_{it} = 0$, $\beta_{it} = -1$, and $\Pi_{it}(p_{it}) = D_{it}(p_{it})$. However, as we will see later, the more general representation allows us to more effectively study, and develop efficient algorithms for, models that incorporate constraints on price patterns (such as the constraint that prices should be constant over the time horizon) while still allowing for a broad class of demand relations. We assume that the functions Π_{it} are strictly increasing, continuous, and convex. Notice that with these assumptions, Π_{it} has an inverse function P_{it} and there exists a one to one relationship between p_{it} and π_{it} given by

$$P_{it}(\pi_{it}) = \Pi_{it}^{-1}(\pi_{it}) = p_{it} \text{ or, equivalently,}$$

$$\Pi_{it}(p_{it}) = P_{it}^{-1}(p_{it}) = \pi_{it}.$$

Therefore, throughout the article, when we are referring to a certain price effect π_{it} we will be implicitly referring to a certain price p_{it} . This allows us to formulate the CLSTP in terms of π_{it} instead of p_{it} . We will represent the set of vectors of feasible prices for item i by Γ_i and assume that it is a nonempty polytope. We next define the revenue for item i in period t , R_{it} , as a function of the price effect π_{it} as

$$R_{it}(\pi_{it}) = p_{it}d_{it} = P_i(\pi_{it})(\alpha_{it} - \beta_{it}\pi_{it}).$$

Given the assumptions on Π_{it} , it is easy to show that the revenue function R_{it} is continuous and concave. We further assume that this concave revenue function has no more than J points at which it is nondifferentiable and achieves its supremum at a finite value $\pi'_{it} \equiv \arg \max_{\pi_{it}} R_{it}(\pi_{it})$. We can now formulate the CLSTP as follows:

$$\text{maximize } \sum_{i=1}^N \sum_{t=1}^T (R_{it}(\pi_{it}) - S_{it}y_{it} - c_{it}x_{it} - h_{it}I_{it}) \quad (1)$$

subject to

$$\sum_{i=1}^N (b_{it}y_{it} + a_{it}x_{it}) \leq C_t \quad t = 1, \dots, T \quad (2)$$

$$I_{i,t-1} + x_{it} = \alpha_{it} - \beta_{it}\pi_{it} + I_{it} \quad t = 1, \dots, T; i = 1, \dots, N \quad (3)$$

$$x_{it} \leq M_{it}y_{it} \quad t = 1, \dots, T; i = 1, \dots, N \quad (4)$$

$$\{\pi_{i1}, \dots, \pi_{iT}\} \in \Gamma_i \quad i = 1, \dots, N \quad (5)$$

$$y_{it} \in \{0, 1\} \quad t = 1, \dots, T; i = 1, \dots, N \quad (6)$$

$$I_{it}, x_{it} \geq 0 \quad t = 1, \dots, T; i = 1, \dots, N \quad (7)$$

$$I_{i0} = \bar{I}_{i0} \quad i = 1, \dots, N. \quad (8)$$

The objective function (1) maximizes the difference between the total revenues and the total costs. Constraints (2) are the capacity constraints; constraints (3) are the flow balance constraints; constraints (4) are the setup forcing constraints; and constraints (6) and (7) ensure integrality and non-negativity of the decision variables. In contrast with the CLST and other traditional lot-sizing problems, we cannot assume without loss of generality that the item inventories are equal to zero at the start of the planning period; we therefore assume that initial inventories are given by $\bar{I}_{i0} \geq 0$ and enforced by constraints (8). Note that M_{it} in constraint (4) should be an upper bound on the quantity of item i produced in period t that is satisfied without loss of optimality. In particular, we may set $M_{it} = \min\{\sum_{s=t}^T (\alpha_{is} - \beta_{is}\pi_{is}^L), (C_t - b_{it})/a_{it}\}$, where π_{it}^L is any valid lower bound on the price effect of item i in period t , so that the first term in the minimum is an the upper bound on the demand for item i in periods t, \dots, T . Later, we will identify such valid lower bounds under different pricing strategies.

The set Γ_i in constraint (5) represents the desired pricing strategy, and in this article, we study two particular pricing strategies in detail: A dynamic and a constant pricing strategy. The first strategy is one where the price of an item can vary from period to period. In that case, Γ_i is defined by lower and upper bound constraints only:

$$\{(\pi_{i1}, \dots, \pi_{iT}) : \pi_{it}^L \leq \pi_{it} \leq \pi_{it}^U, t = 1, \dots, T\}$$

where we may choose $\pi_{it}^L = \Pi_{it}^{-1}(0)$ to ensure that the price of an item is always non-negative, and $\pi_{it}^U = \alpha_{it}/\beta_{it}$ to ensure that the resulting demand is always non-negative. In fact, in the absence of initial inventories (i.e., $\bar{I}_{i0} = 0$), we can potentially tighten the lower bound constraints by recalling that the revenue function attains its maximum at π_{it}' , and that demands are nonincreasing in the price effect. This means that any demands resulting from a price effect below π_{it}' would not be profitable. Therefore, we can set

$\pi_{it}^L = \max\{\Pi_{it}^{-1}(0), \pi_{it}'\}$. (Note that, in the presence of initial inventories, higher demands could be profitable since the initial inventories are essentially procured costlessly.) In the second pricing strategy that we consider, the price of an item should be constant over the planning horizon. We follow earlier studies that consider this pricing strategy by restricting ourselves to demand functions for which the functions Π_{it} are stationary, i.e., $\Pi_{it} = \Pi_i$ for $t = 1, \dots, T$ and $i = 1, \dots, N$. In that case, Γ_i is given by:

$$\{(\pi_{i1}, \dots, \pi_{iT}) : \pi_{it} = \pi_{i1}, t = 2, \dots, T; \pi_{i1}^L \leq \pi_{i1} \leq \pi_{i1}^U\}$$

where, similarly to the dynamic pricing case, we may set $\pi_{i1}^L = \Pi_i^{-1}(0)$ to ensure that the price of an item is always non-negative; and $\pi_{i1}^U = \min_{1 \leq t \leq T} \{\alpha_{it}/\beta_{it}\}$ to ensure that the resulting demands are non-negative in each period. Interestingly, under this strategy and demand relation, we have that $\pi_{it} = \pi_{i1}$ for $t = 2, \dots, T$, so that we may have that, in the optimal solution, $\pi_{it} < \pi_{it}'$ in one or more periods t even when there are no initial inventories. However, when there are no initial inventories for item i , we can still find a potentially tighter lower bound on π_{i1} by considering the total revenue curve $R_i = \sum_{t=1}^T R_{it}$. Since each of the revenue functions R_{it} attains its maximum at a finite price effect π_{it}' , the function R_i will attain its maximum at a finite value $\pi_i' \equiv \arg \max_{\pi_i} R_i(\pi_i)$ as well. This means that this value provides a lower bound on π_{i1}^L provided it is no larger than π_{i1}^U . Therefore, without loss of optimality, we can set $\pi_{i1}^L = \max\{\Pi_i^{-1}(0), \min\{\pi_i', \pi_{i1}^U\}\}$.

3. DANTZIG-WOLFE DECOMPOSITIONS OF THE CLSTP

3.1. Framework Development

The idea behind applying Dantzig-Wolfe decomposition of multi-item capacitated lot-sizing problem is to write feasible solutions as a convex combination of extreme points of convex hulls of judiciously defined subproblems. For the CLSTP, these subproblems are single-item lot-sizing problems with pricing, in analogy with the CLST where these subproblems are single-item lot-sizing problems (see Manne [13] and Degraeve and Jans [3]). In both cases, the relaxation of such a reformulation is tighter than the relaxation of the traditional formulation since the traditional formulation of the subproblems does not have the so-called integrality property. In this section, we will develop two alternative Dantzig-Wolfe decompositions of the CLSTP.

As in the traditional CLST, the capacity constraints (2) are the constraints that tie the different items; if they are removed, the problem reduces to a collection of independent uncapacitated single item lot-sizing problems with pricing decisions

for each item i . Now let X_i be the region defined by constraints (3)–(8) for item i . Note that since Γ^i is bounded, X_i is a polytope as well. We can therefore build a Dantzig-Wolfe decomposition formulation of the CLSTP by writing any feasible solution to the problem as a convex combination of the finite set of extreme points E_i of $\text{conv}(X_i)$. For ease of exposition, we will refer to a typical element of E_i as either $(y_{i1}^j, \dots, y_{iT}^j, x_{i1}^j, \dots, x_{iT}^j, I_{i1}^j, \dots, I_{iT}^j, \pi_{i1}^j, \dots, \pi_{iT}^j)$ or, for short, j , with associated cost $\kappa_i^j = \sum_{t=1}^T (S_{it}y_{it}^j + c_{it}x_{it}^j + h_{it}I_{it}^j)$ and capacity consumption $\rho_{it}^j = a_{it}y_{it}^j + b_{it}x_{it}^j$. Letting the decision variable λ_i^j represent the weight of production and demand plan $j \in E_i$ in the convex combination, we obtain the following formulation of the CLSTP:

$$\text{maximize } \sum_{i=1}^N \sum_{t=1}^T R_{it} \left(\sum_{j \in E_i} \pi_{it}^j \lambda_i^j \right) - \sum_{i=1}^N \sum_{j \in E_i} \kappa_i^j \lambda_i^j$$

subject to

$$\sum_{i=1}^N \sum_{j \in E_i} \rho_{it}^j \lambda_i^j \leq C_t \quad t = 1, \dots, T \quad (9)$$

$$\sum_{j \in E_i} y_{it}^j \lambda_i^j \in \{0, 1\} \quad i = 1, \dots, N; t = 1, \dots, T \quad (10)$$

$$\sum_{j \in E_i} \lambda_i^j - 1 = 0 \quad i = 1, \dots, N \quad (11)$$

$$-\lambda_i^j \leq 0 \quad j \in E_i; i = 1, \dots, N. \quad (12)$$

Constraints (9) ensure that the weighted average of the capacity requirements of the extreme plans in period t should not exceed the available capacity in that period, whereas constraints (10) relate to the original binary setup indicator variables y_{it} and state that the weighted averages of the extreme point setup variables should represent a valid production plan and therefore be binary. Constraints (11) and (12) ensure that we indeed consider convex combinations of extreme point plans only. This reformulation leads to a branch-and-price algorithm in which we use the continuous relaxation of this formulation to determine upper bounds. This continuous relaxation itself will be solved through a column generation approach in which we repeatedly solve a so-called restricted master problem containing only a (relatively small) subset of the decision variables (columns) and determine whether additional columns should be added through an associated pricing problem. Since the aforementioned formulation of the CLSTP contains only a finite number of decision variables, this column generation algorithm will converge finitely. We will therefore refer to this formulation as the *Finite Formulation*.

Although the finiteness of the formulation is certainly a major advantage, this approach suffers from a major drawback as well: The master problems to be solved in the column generation phases are concave maximization problems. While effective algorithms for solving such nonlinear optimization problems exist, it is nevertheless questionable whether the need to repeatedly solve such a problem will yield an effective solution approach to the CLSTP. Our alternative approach is based on the following reformulation of the CLSTP:

$$\text{maximize } \sum_{i=1}^N \sum_{t=1}^T (r_{it} - S_{it}y_{it} - c_{it}x_{it} - h_{it}I_{it}) \quad (13)$$

subject to

$$\sum_{i=1}^N (b_{it}y_{it} + a_{it}x_{it}) \leq C_t \quad t = 1, \dots, T \quad (14)$$

$$r_{it} \leq R_{it}(\pi_{it}) \quad t = 1, \dots, T; \\ i = 1, \dots, N \quad (15)$$

$$(x_i, y_i, I_i, \pi_i) \in X_i \quad i = 1, \dots, N \quad (16)$$

where we note that, without loss of optimality, constraints (15) will be satisfied at equality. Letting \hat{X}_i be the region defined by constraints (15)–(16) for item i , it is not hard to see that any optimal solution to the CLSTP can be expressed as a convex combination of a finite number of extreme points of $\text{conv}(\hat{X}_i)$. However, this set of extreme points, say \hat{E}_i , may contain an uncountably infinite number of points (i.e., we cannot enumerate its elements and index them by the natural numbers), making a direct analogon of the earlier approach impossible since the full master problem would then contain an uncountably infinite number of decision variables. However, any finite subset of points from \hat{E}_i would define a restricted master problem whose continuous relaxation is a linear program. We will show that this leads to a branch-and-price algorithm in which we solve a continuous relaxation of the CLSTP using column generation, where the restricted master problems are linear programs. However, the price we have to pay for this is that the column generation phases may not terminate finitely. We will therefore refer to this formulation as the *Infinite Formulation*.

In the remainder of this section, we will develop pricing problems that are used to generate promising columns in the column generation phase of both the finite and infinite formulations.

3.2. Pricing Problem

3.2.1. Finite Formulation

Although we have not yet explicitly characterized the set of extreme points E_i , we are now able to derive the general form

of the pricing problem that, in the column generation phase of our algorithm, identifies one or more additional columns that should be added to the master problem. To this end, consider the continuous relaxation of the binary constraints (10):

$$\sum_{j \in E_i} y_{it}^j \lambda_i^j \leq 1 \quad i = 1, \dots, N; t = 1, \dots, T.$$

Since in any production schedule $j \in E_i$, we have that $y_{it}^j \in \{0, 1\}$ for all $t = 1, \dots, T$, constraints (11) and (12) imply that this constraint is redundant and we may remove it from the problem to obtain a continuous relaxation of the CLSTP. Since the objective function is concave and the feasible region is a polytope, a feasible solution vector $\bar{\lambda}$ is optimal to the relaxation problem if and only if there exists a solution to the following KKT conditions (in generalized form, to account for revenue functions that are not everywhere differentiable), where the dual variables θ_t are associated with constraints (9), ϕ_i with constraints (11), and γ_i^j with constraints (12):

$$\begin{aligned} & - \sum_{t=1}^T \partial R_{it} \left(\sum_{j \in E_i} \pi_{it}^j \bar{\lambda}_i^j \right) + \kappa_i^j \\ & \quad + \sum_{t=1}^T \theta_t \rho_{it}^j - \gamma_i^j + \phi_i \geq 0 \quad j \in E_i; i = 1, \dots, N \\ & \theta_t \left(\sum_{i=1}^N \sum_{j \in E_i} \rho_{it}^j \bar{\lambda}_i^j - C_t \right) = 0 \quad t = 1, \dots, T \\ & \phi_i \left(\sum_{j \in E_i} \bar{\lambda}_i^j - 1 \right) = 0 \quad i = 1, \dots, N \\ & \quad - \bar{\lambda}_i^j \gamma_i^j = 0 \quad j \in E_i; i = 1, \dots, N \\ & \quad \gamma_i^j \geq 0 \quad j \in E_i; i = 1, \dots, N \\ & \quad \theta_t \geq 0 \quad t = 1, \dots, T. \end{aligned}$$

Using the definitions of κ_i^j and ρ_{it}^j , the first set of KKT conditions is equivalent to

$$\begin{aligned} & - \sum_{t=1}^T u_{it} \pi_{it}^j + \sum_{t=1}^T (S'_{it} y_{it}^j + c'_{it} x_{it}^j + h_{it} I_{it}^j) + \phi_i \\ & \quad = \gamma_i^j \geq 0 \quad j \in E_i \quad (17) \end{aligned}$$

where

$$\begin{aligned} \bar{u}_{it} & \in \partial R_{it} \left(\sum_{j \in E_i} \pi_{it}^j \bar{\lambda}_i^j \right) \quad t = 1, \dots, T; i = 1, \dots, N \\ S'_{it} & = S_{it} + \theta_t b_{it} \quad t = 1, \dots, T; i = 1, \dots, N \end{aligned}$$

and

$$c'_{it} = c_{it} + \theta_t a_{it} \quad t = 1, \dots, T; i = 1, \dots, N.$$

Now suppose that $\bar{\lambda}$ is actually the optimal solution to a *restricted* relaxed master problem with only a subset of the extreme points in E_i ($i = 1, \dots, N$) along with a dual solution $(\bar{\theta}, \bar{\phi}, \bar{\gamma})$. If (17) happens to be satisfied then we can conclude that we have found the optimal solution to the relaxation. Otherwise, we can add to the master problem any extreme plans $j \in E_i$ for which (17) is violated. In particular, we may choose to add, for each item i , the extreme plan for which (17) is most violated (if any). We can find that plan by finding an extreme point optimal solution to the following pricing problem for each item i :

$$\begin{aligned} & \text{maximize} \quad \sum_{t=1}^T \bar{u}_{it} \pi_{it} - \sum_{t=1}^T (\bar{S}'_{it} y_{it} + \bar{c}'_{it} x_{it} + h_{it} I_{it}) - \bar{\phi}_i \\ & \text{subject to} \end{aligned} \quad (PP-F)$$

$$(x_i, y_i, I_i, \pi_i) \in X_i$$

where $\bar{S}'_{it} = S_{it} + \bar{\theta}_t b_{it}$ and $\bar{c}'_{it} = c_{it} + \bar{\theta}_t a_{it}$ ($t = 1, \dots, T; i = 1, \dots, N$). Clearly, if the optimal solution to this problem is negative, the optimal solution corresponds to the constraint in (17) that is most violated. Now (PP-F) is an uncapacitated single-item lot-sizing problem with pricing decisions, linear revenue functions, and initial inventories, and we will focus on solution approaches to this problem in Section 4.

3.2.2. Infinite Formulation

As mentioned earlier, the finiteness of the approach developed in Section 3.2.1 comes at the expense of a master problem that is a concave optimization problem. We could address this fact by restricting ourselves to piecewise-linear revenue functions, which would allow for the reformulation of the relaxation of the finite formulation as a linear programming problem. In general, however, we propose to use a different approach which, as it will turn out, essentially constructs a sequence of ever more accurate piecewise-linear concave lower approximations to the revenue functions. In light of the reformulation (13)–(16) of the CLSTP, intuition suggests that the pricing problem for generating additional columns should be

$$\begin{aligned} & \text{maximize} \quad \sum_{t=1}^T R_{it}(\pi_{it}) - \sum_{t=1}^T (\bar{S}'_{it} y_{it} + \bar{c}'_{it} x_{it} + h_{it} I_{it}) - \bar{\phi}_i \\ & \text{subject to} \end{aligned} \quad (PP-I)$$

$$(x_i, y_i, I_i, \pi_i) \in X_i.$$

Now (PP-I) is an uncapacitated single-item lot-sizing problem with pricing decisions, concave revenue functions, and initial inventories, and we will focus on solution approaches to this problem in Section 4. It is clear that this approach suffers from two potential complications: (i) The column generation algorithm could potentially generate an infinite number of columns and (ii) it is not obvious that the procedure converges to an optimal solution to the relaxation.

However, the correctness of the aforementioned algorithm follows from the fact that it is an application of the algorithm proposed by Dantzig [2] for convex programming problems. Dantzig [1] proved that this column generation algorithm either finds the optimum in a finite number of iterations or converges to the optimal solution as long as there exists a nondegenerate basic solution to the master problem. (In Section 5, we will address the problem of finding a nondegenerate basic solution to the master problem.) Since each iteration of this algorithm adds a new breakpoint to a piecewise-linear concave underapproximation to the concave revenue functions, this algorithm is sometimes also called *grid linearization* (see Lasdon [12]). This method for creating a piecewise-linear approximation is more efficacious than creating a linear approximation to the function in advance since, with this approach, we only introduce breakpoints as needed and make an accurate approximation to the function only in the neighborhood of promising solutions.

4. SOLVING THE PRICING PROBLEM

For both the finite and the infinite formulation, the pricing problem for a given item is of the following generic form:

$$\text{maximize } \sum_{t=1}^T R_t(\pi_t) - \sum_{t=1}^T (S_t y_t + c_t x_t + h_t I_t)$$

subject to (PP)

$$\begin{aligned} I_{t-1} + x_t &= \alpha_t - \beta_t \pi_t + I_t & t = 1, \dots, T \\ x_t &\leq M_t y_t & t = 1, \dots, T \\ \{\pi_1, \dots, \pi_T\} &\in \Gamma \\ y_t &\in \{0, 1\} & t = 1, \dots, T \\ I_t, x_t &\geq 0 & t = 1, \dots, T \\ I_0 &= \bar{I}_0. \end{aligned} \tag{18}$$

It will be convenient to eliminate the inventory variables using the inventory balance constraints (18):

$$I_{it} = \bar{I}_{i0} + \sum_{j=1}^t x_{ij} - \sum_{j=1}^t (\alpha_{ij} - \beta_{ij} \pi_{ij}) \quad t = 1, \dots, T.$$

Then, if we define

$$\begin{aligned} \tilde{R}_t(\pi_t) &= R_t(\pi_t) - \pi_t \beta_t \sum_{\tau=t}^T h_\tau & t = 1, \dots, T \\ \tilde{c}_t &= c_t + \sum_{\tau=t}^T h_\tau & t = 1, \dots, T \end{aligned}$$

we can formulate (PP) as

$$\text{maximize } \sum_{t=1}^T \tilde{R}_t(\pi_t) - \sum_{t=1}^T (S_t y_t + \tilde{c}_t x_t)$$

subject to

$$\begin{aligned} \bar{I}_0 + \sum_{s=1}^t x_s - \sum_{s=1}^t (\alpha_s - \beta_s \pi_s) &\geq 0 & t = 1, \dots, T \\ x_t &\leq M y_t & t = 1, \dots, T \\ \{\pi_1, \dots, \pi_T\} &\in \Gamma \\ y_t &\in \{0, 1\} & t = 1, \dots, T \\ x_t &\geq 0 & t = 1, \dots, T \end{aligned}$$

(where we have omitted the constant $-\sum_{t=1}^T h_t \bar{I}_0$ from the objective).

4.1. Dynamic Pricing Strategy

Recall that, under a dynamic pricing strategy, the set Γ contains only bound constraints. For this case, Thomas [14] proposes a dynamic programming algorithm that solves the problem in $O(T^2)$ time when no initial inventory is present and the inverse of the derivative of the revenue functions can be evaluated in constant time. This algorithm is based on the observation that there exists an optimal production and pricing plan that possesses the zero-inventory-ordering (ZIO) property and can therefore, like for the standard economic lot-sizing problem with fixed demands (see Wagner and Whitin [17]), be decomposed into a sequence of regeneration intervals. (We will, with a slight abuse of notation, refer to such solutions as *extreme point solutions*.) The problem can then be solved by determining the optimal price effects (or, equivalently, demands or prices) for the periods in each regeneration interval. Letting $f(s, t)$ be the maximum profit obtainable in the regeneration interval (s, t) (i.e., in periods s, \dots, t) and $F(t)$ the maximum profit obtainable in periods 1 through t , the following backward dynamic recursion solves the problem:

$$\begin{aligned} F(s) &= \max_{t \geq s} \{f(s, t) + F(t + 1)\} & s = 1, \dots, T \\ F(T + 1) &= 0. \end{aligned}$$

To generalize this result to the case with initial inventory, we go back to the underlying core observation that, for fixed price effects (i.e., fixed demands) the resulting concave cost minimization problem can be formulated as a shortest path problem in a network with a demand node for each period, in addition to two supply nodes corresponding to production and initial inventory. It is well-known that there exists an optimal solution to the problem in which the set of arcs carrying positive flow is acyclic. If we then, similarly to the problem without initial inventories, define a regeneration interval as starting in either the first period or a period with no incoming inventory, we can immediately conclude that we may again limit ourselves to solutions that decompose into regeneration intervals. Moreover, for any regeneration interval (s, t) with $s > 1$, we can determine the associated optimal profit $f(s, t)$ in precisely the same way as for the problem without initial inventories. For regeneration intervals of the form $(1, t)$, however, the situation is different. In the first regeneration interval of an optimal extreme point solution, production will take place in no more than one period, so that we should consider two different situations: (i) No production period and (ii) exactly one production period in $\{1, \dots, t\}$. For each candidate first regeneration interval $(1, t)$, we will therefore consider both options and set $f(1, t)$ to the largest corresponding profit. Interestingly, since we know that there exists an extreme point optimal solution to the overall problem, we may ignore any nonnegativity constraints on the inventories in the regeneration interval and discard any candidate solutions in which these are violated. Next, we consider in more detail the two situations mentioned earlier.

1. If there is no production in the regeneration interval $(1, t)$, the total demand satisfied in periods 1 through t is equal to the initial inventory \bar{I}_0 if $t < T$ and will not exceed \bar{I}_0 if $t = T$. In other words, ignoring the nonnegativity constraints on the inventory levels as discussed earlier we should solve the following optimization problem if the entire initial inventory is used up in regeneration interval $(1, t)$:

$$\text{maximize } \sum_{s=1}^t \tilde{R}_s(\pi_s)$$

subject to

$$\sum_{s=1}^t (\alpha_s - \beta_s \pi_s) = \bar{I}_0$$

$$\pi_s^L \leq \pi_s \leq \pi_s^U \quad s = 1, \dots, t.$$

Appropriately redefining \tilde{R} outside the bound constraints, we obtain that an optimal solution to this problem is given by values of r and π_t ($t = 1, \dots, t$) such that

$$\tilde{R}_s^+(\pi_s) \leq r \leq \tilde{R}_s^-(\pi_s) \quad s = 1, \dots, t \quad (19)$$

$$\sum_{t=1}^s (\alpha_s - \beta_s \pi_s) = \bar{I}_0$$

where $\tilde{R}_s^+(\pi_s)$ and $\tilde{R}_s^-(\pi_s)$ are, respectively, the right and left derivatives of \tilde{R}_s , respectively. (Note that, due to the bound constraints, we set $\tilde{R}_s^-(\pi_s^L) = 0$ and $\tilde{R}_s^+(\pi_s^U) = -\infty$.) Interestingly, this is exactly the same setting studied in Geunes et al. [5] for the lot-sizing problem with constant capacities and dynamic pricing decisions in a regeneration interval, where the first period is the only production period and the production capacity is equal to \bar{I}_0 . The results in that article then imply that this subproblem can be solved in $O((J + 1 + R \log v)T)$ time, where

$$v = \max_{s=1, \dots, t} \tilde{R}_s^+(\pi_s^U) - \min_{s=1, \dots, t} \tilde{R}_s^-(\pi_s^L)$$

and $O(R)$ is the time required to find a value of π_s satisfying (19) for some value of r .

In case $t = T$, we also have to account for the possibility that some of the initial inventories remain at the end of the planning horizon. This will only happen if there is no marginal revenue associated with satisfying any demand from these remaining inventories. In other words, this solution should satisfy $\tilde{R}_s^+(\pi_s) = 0$ for $s = 1, \dots, T$ and $\sum_{t=1}^s (\alpha_s - \beta_s \pi_s) \leq \bar{I}_0$. It is easy to see that such a solution, if one exists, can be found in $O(RT)$ time.

2. Next, suppose that production in regeneration interval $(1, t)$ takes place in period τ . We again ignore the nonnegativity constraints on the inventory levels as discussed earlier. Moreover, we will have that the production quantity in period τ satisfies

$$x_\tau = \bar{I}_0 - \sum_{s=1}^t (\alpha_s - \beta_s \pi_s)$$

so that we should solve the following optimization problem:

$$\text{maximize } \sum_{s=1}^t (\tilde{R}_s(\pi_s) - \tilde{c}_\tau \beta_s \pi_s)$$

subject to

$$\pi_s^L \leq \pi_s \leq \pi_s^U \quad s = 1, \dots, t$$

(where we have omitted the constant $-S_\tau - \tilde{c}_\tau \bar{I}_0 + \tilde{c}_\tau \sum_{s=1}^t \alpha_s$ from the objective). It is easy to see that an optimal solution to this subproblem satisfies

$$\tilde{R}_s^+(\pi_s) \leq \tilde{c}_\tau \beta_s \leq \tilde{R}_s^-(\pi_s) \quad s = 1, \dots, T$$

and this solution can be found in $O(RT)$ time. This result is consistent with a similar result in Geunes et al. [5] for regeneration intervals with a single fractional procurement period in the lot-sizing problem with constant capacities and dynamic pricing decisions.

Now observe that there are $O(T^2)$ regeneration intervals with $s > 1$, and the profits of these can be found in $O(RT^2)$ time. Moreover, for the $O(T)$ initial regeneration intervals, we can find the profit in case (i) in $O((J + 1 + R \log \nu)T)$, and the profit in case (ii) for each of the $O(T)$ fixed production periods in $O(RT)$ time. This means that (PP) under a dynamic pricing strategy can be solved in $O((J + 1 + R \log \nu)T^2)$ time. In particular, this immediately implies that, for the special case where the revenue functions are linear (which is relevant in our algorithm for the Finite Formulation), (PP) under a dynamic pricing strategy can be solved in $O((1 + \log \nu)T^2)$ time.

4.2. Constant Pricing Strategy

Under a constant pricing strategy, we add the constraints $\pi_t = \pi_1$ for $t = 2, \dots, T$ to the definition of the set Γ . For convenience, we will in this section simply denote the single price effect variable by π , and the corresponding bounds by π^L and π^U . van den Heuvel and Wagelmans [9] study this problem in the case of zero initial inventories and under the assumption that period 1 is a production period and develop an exact algorithm.

In general, suppose that τ is the first production period. Noting that we will again restrict ourselves to extreme point solutions, this production period is either (i) the first period of the second regeneration interval (and demand in the first regeneration interval is satisfied precisely by initial inventory only) or (ii) in the first regeneration interval (and demand in the first regeneration interval is satisfied by initial inventory and production in period τ). We will consider these two cases separately (where, when appropriate, we will use the notation from Section 4.1):

1. In this case, it is easy to see that this can only happen if $\tau > 1$ and

$$I_0 = \sum_{s=1}^{\tau-1} (\alpha_s - \beta_s \pi)$$

or, equivalently,

$$\pi = \pi^{(\tau)} \equiv \frac{\sum_{s=1}^{\tau-1} \alpha_s - I_0}{\sum_{s=1}^{\tau-1} \beta_s}.$$

If, in the latter case, $\pi^{(\tau)} \in [\pi^L, \pi^U]$ we can then determine all demand levels and solve a standard

economic lot-sizing problem over periods τ, \dots, T using the algorithm of Wagelmans et al. [16].

2. In this case, a particular price effect is only valid if the initial inventories are sufficiently high to satisfy all demands up to the first production period, so that we should ensure that

$$I_0 \geq \sum_{s=1}^{\tau-1} (\alpha_s - \beta_s \pi)$$

or, equivalently, we should restrict ourselves to values

$$\pi \in [\max\{\pi^L, \pi^{(\tau)}\}, \pi^U].$$

We next follow van den Heuvel and Wagelmans [9] and write the total lot-sizing costs as a function of the price effect π for a given set $S \subseteq \{\tau, \dots, T\}$ of production periods (with, of course, $\tau \in S$). Letting $\tau_t(S)$ denote the first production period in $S \cap \{t, \dots, T\}$, this cost function reads as follows:

$$\begin{aligned} C^{(\tau, S)}(\pi) &= \sum_{t \in S} S_t - \tilde{c}_\tau \bar{I}_0 + \sum_{t=\tau}^T \tilde{c}_{\tau_t(S)} (\alpha_t - \beta_t \pi) \\ &= \sum_{t \in S} S_t - \tilde{c}_\tau \bar{I}_0 + \sum_{t=\tau}^T \tilde{c}_{\tau_t(S)} \alpha_t - \sum_{t=\tau}^T \tilde{c}_{\tau_t(S)} \beta_t \pi \\ &= A^{(\tau, S)} - B^{(\tau, S)} \pi \end{aligned}$$

where $A^{(\tau, S)}$ and $B^{(\tau, S)}$ are defined appropriately. This means that $C^{(\tau, S)}$ is piecewise-linear and convex and, moreover, has the same structure as the cost function in van den Heuvel and Wagelmans [9]. We can therefore find the optimal price effect for this case using their algorithm.

It is interesting to note that, for the special case where the revenue functions are linear (which is relevant in our algorithm for the Finite Formulation), we only need to consider the $O(T)$ (valid) price effects in $\{\pi^L, \pi^U\} \cup \{\pi^{(\tau)} : \tau = 2, \dots, T\}$, yielding a much more efficient algorithm for solving the (PP) under a constant pricing strategy in this special case.

5. BRANCH-AND-PRICE ALGORITHM

In this section, we discuss some implementation details of our branch-and-price algorithm. In particular, we will address how we guarantee convergence of the column generation procedure for solving the infinite formulation. However, before

we proceed it is important to note that, since finite termination of the column generation algorithm for the infinite formulation cannot be guaranteed, we have to allow for a finite prespecified tolerance; in particular, we will consider the problem solved as soon as we find a solution that is within $\epsilon > 0$ in value of the optimal solution.

5.1. Initial Columns and Convergence

To start the column generation process at a particular node with either the finite or the infinite formulation, we need to find an initial basic feasible solution to the master problem at that node. When such a feasible solution can be found, convergence with finite formulation is guaranteed because the number of extreme points of the formulation is finite. However, to guarantee convergence with the infinite formulation Dantzig [1, 2] shows that at least one nondegenerate basic feasible solution to the master problem at a node should exist. In our implementation of this method, we will make sure that we start the algorithm with an initial nondegenerate basic feasible solution.

5.1.1. Dynamic Pricing Strategy

In the case of dynamic prices, we start with an initial solution in which no production takes place for any of the items. Furthermore, in these production plans, we set the prices as high as possible, i.e., $\pi_{it} = \pi_{it}^U$ ($t = 1, \dots, T$), so that the demands in all periods are zero. This means that, if there are positive initial inventories these are carried over to the end of the planning horizon without being consumed. This solution can be represented through a column for each item, each of which represents a “production plan” for the corresponding item. A basis for this initial solution then consists of the slack variables of the capacity constraints (9) and the λ_i variables corresponding to the initial production plans for items $i = 1, \dots, N$. Since capacities are strictly positive this basis is nondegenerate so that convergence of the column generation algorithm for both the finite and the infinite formulation is guaranteed.

5.1.2. Constant Pricing Strategy

Under the assumption of a constant price for each item, it may not be possible to find a feasible solution in which all item demands are zero in each period. In particular, setting the prices to their upper bounds, i.e., $\pi_{i1} = \pi_{i1}^U = \min_t \left\{ \frac{\alpha_{it}}{\beta_{it}} \right\}$ does not guarantee that the corresponding demands are all equal to zero, so that positive production may be required to satisfy these demands. In fact, finding a feasible integral solution to the master problem with the prices that correspond to the lowest possible demands (or recognizing that such a solution does not exist) is NP-hard (since it is the feasibility

question for an instance of the CLST problem). We therefore propose to use a two-phase column generation approach in which we first find a (nondegenerate) basic feasible solution to the problem (if one exists), and then solve our optimization problem from this starting solution. With the prices at their upper bounds as mentioned earlier, we start the first phase by attempting to satisfy the corresponding demands through initial inventories. If these initial inventories suffice to satisfy all demands, then we have obtained an initial nondegenerate basic feasible solution as in Section 5.1.1. However, if any demands remain unsatisfied, we solve an auxiliary problem where we relax the capacity constraints and maximize the minimum amount of unused capacity (which may be negative!) over periods $t = 1, \dots, T$. If the optimal objective function value of this problem is negative, we conclude that there is no feasible solution to our problem (at the current node in the branch-and-bound tree). Otherwise, if the optimal objective function value is non-negative, the optimal solution corresponds to a nondegenerate basic feasible solution which can be used to start column generation at that node.

5.2. Bounding

At each iteration of the column generation process, we calculate upper and lower bounds on the optimal objective function value of the relaxed problem. Let v_{RMP}^k be the optimal value of the restricted master problem at iteration k . Clearly, this provides a lower bound on the optimal solution value of the relaxed master problem. Moreover, letting $v_{\text{SP}_i}^k$ be the reduced costs of the columns that we have generated for items $i = 1, \dots, N$ at iteration k of the column generation process an upper bound on the optimal solution value of the relaxed master problem is given by

$$v_{\text{RMP}}^k + \sum_{i=1}^N v_{\text{SP}_i}^k$$

(see Dantzig [2] and Lasdon [12]). We stop whenever the difference between these upper and lower bounds is below the tolerance $\epsilon > 0$. (Note that, when using the Finite Formulation, we could use a tolerance of $\epsilon = 0$ if desired; however, when using the Infinite Formulation finite termination can only be guaranteed when a positive tolerance is used.)

5.3. Branching

When the column generation stops at the root node, we check whether the binary constraints are satisfied for the y_{it} variables. If none of the y_{it} variables are fractional, we conclude that we have found an integral solution at the root node (that is within the desired tolerance). Otherwise, we branch on one of the fractional y_{it} variables, branching down by imposing the constraint $y_{it} = 0$ and branching up by imposing the

Table 1. Comparison of branching strategies.

# Items	Formulation					
	Finite			Infinite		
	10	20	30	10	20	30
Branching strategy						
Closest to 0	3.80	4.20	4.00	0.70	0.30	0.26
Closest to 0.5	3.36	3.47	3.26	0.66	0.38	0.20
Closest to 1	3.33	3.43	3.20	0.63	0.34	0.20

constraint $y_{it} = 1$ on the child nodes. To obtain a high-quality and robust selection rule for branching, we performed preliminary experiments on a subset of test instances. In particular, we tested the following branching strategies on an independent set of instances: (i) Branching on the variable closest to 0, (ii) branching on the variable closest to 0.5, and (iii) branching on the variable closest to 1. Table 1 above summarizes the average error gap that we obtained with these branching strategies. Although there is no branching strategy that uniformly dominates, branching on the variable closest to 1 resulted in lower error gaps in the majority of preliminary test instances. Therefore, for the remainder of the computational experiments, we focused on that branching strategy.

5.4. Search Strategy and Heuristics

We perform a *hybrid depth-first search strategy* where, after the two child nodes of the current node are generated, linear relaxations in both child nodes are solved to optimality. Then, we first investigate the child node with the higher upper bound on the objective value of the relaxed problem.

After we complete the column generation process at a node, we perform a simple rounding heuristic in an attempt to find a feasible solution. We set a *cutoff* value between 0 and 1, and round y_{it} variables that exceed the cutoff up to 1 and down to 0 otherwise. We then solve for the optimal values of the remaining decision variables to obtain an integer solution. This procedure is repeated for different values of the cutoff parameter: Increasing from 0.1 to 1 at 0.1 increments. Similar to what was observed by Degraeve and Jans [3], we empirically find that the objective values obtained by this repeated rounding heuristic appear to follow a unimodal pattern as the cutoff value is increased. Therefore, we stop this heuristic as soon as the objective value decreases.

6. COMPUTATIONAL RESULTS

6.1. Creating Problem Instances

For our computational tests, we modified a collection of 540 widely used problem instances created by Trigeiro et al. [15] for the CLST. These problem instances for the

CLST were generated using a full factorial design on five problem characteristics: Number of items, coefficient of variation of demand across periods, time between orders, average setup times, and capacity utilization, while the number of periods was fixed to $T = 20$. In our problem instances, the second characteristic was not used because we do not have a fixed demand pattern. Instead, we created parameter values for the demand functions as follows: α_{it} fixed at 250 and β_{it} randomly generated from the uniform distribution on the set $\{2.0, 2.5, 3.0\}$. This yields a range of demand values from 0 to 250. Furthermore, we chose $\Pi_{it}(p_{it}) = p_{it}$, corresponding to a commonly used linear relationship between demands and prices. This, in turn, means that the revenue functions are quadratic: $R_{it}(\pi_{it}) = \alpha_{it}\pi_{it} - \beta_{it}\pi_{it}^2$, so that our instances of the CLSTP are mixed-integer quadratic optimization problems.

Our problem instances have the following properties with respect to the remaining four characteristics. First, we consider instances for which the number of items is $N \in \{10, 20, 30\}$. With respect to the time between orders (TBO), capacities, and setup times, we first solved all problem instances without capacity constraints but with Trigeiro’s values for the setup times. For each instance, the ratio between setup and holding costs determines the average TBO. We then classified all instances according to an average TBO of 1 (low), 2 (medium), and 4 (high) periods. Next, the capacity in each period was set to 50% (low), 60% (medium), and 70% (high) of the average capacity consumption over the planning horizon on the unconstrained solution. We then further grouped all instances according to setup times consuming 10% (low) or 30% (high) of total capacity. Finally, we randomly generated initial inventories for all items in such a way that, on average, the initial inventory satisfies about half of the time between setups. In particular, for instances with high, medium, and low TBO we generated initial inventories uniformly from the interval $[0, 400]$, $[0, 200]$, and $[0, 100]$, respectively.

6.2. Computational Tests

We tested both of our algorithms on each of the 540 problem instances under both dynamic and constant pricing strategies (for a total of 1080 problem instances). The linear and quadratic mixed-integer problems were all solved using CPLEX 11.1. In addition, we solved the formulation of the CLSTP from Section 2 of each instance directly using CPLEX as well. All experiments were performed on a 3.4 Ghz Pentium IV System with 2 GB of RAM under Windows XP. Because of the difficulty of solving especially large-scale instances to optimality, we imposed a global upper bound on the solution time of 1200 s for all instances and all solution approaches.

We summarize our results by pricing strategy (constant, dynamic), capacity values (low, medium, high), setup time

Table 2. Average percent error gaps obtained with the Finite Formulation using its own upper bound (and using the upper bound found with the Infinite Formulation).

# Items	Pricing strategy					
	Constant			Dynamic		
	10	20	30	10	20	30
Capacity						
Low	2.55 (1.89)	2.30 (1.65)	2.26 (1.68)	3.25 (1.16)	3.26 (1.17)	3.56 (1.50)
Medium	1.98 (1.47)	1.69 (1.22)	1.56 (1.11)	2.41 (0.85)	2.27 (0.84)	2.60 (1.18)
High	1.27 (0.97)	0.97 (0.72)	0.85 (0.61)	1.54 (0.56)	1.39 (0.51)	1.59 (0.75)
Setup time						
Low	1.35 (0.97)	1.07 (0.72)	1.05 (0.70)	1.76 (0.59)	1.74 (0.59)	1.89 (0.77)
High	2.51 (1.92)	2.24 (1.67)	2.06 (1.57)	3.04 (1.11)	2.87 (1.10)	3.28 (1.51)
TBO						
Low	1.45 (1.10)	1.26 (0.90)	1.14 (0.79)	1.67 (0.56)	1.78 (0.67)	2.04 (0.91)
Medium	1.82 (1.24)	1.56 (1.05)	1.47 (1.00)	2.34 (0.79)	2.25 (0.81)	2.50 (1.09)
High	2.53 (1.99)	2.15 (1.64)	2.06 (1.61)	3.22 (1.21)	2.88 (1.05)	3.21 (1.43)

values (low, high), and TBO (low, medium, high) to assess the effect of the corresponding parameters on the algorithms' performance. We define the error gap as $(UB - LB)/LB$, where LB is a lower bound, i.e., the value of the best integral solution found by the algorithm within the given time limit, and UB is the best upper bound on the optimal integer solution value to the problem instance, i.e., the maximum of all upper bounds at the leaf nodes of the branch-and-bound tree upon termination. (We would like to remark here that we have observed that the upper bound found at the root node is generally tighter for the Infinite Formulation than for the Finite Formulation; in particular, the upper bound for the latter can be up to about 5% higher than that for the former.) Tables 2 and 3 show the average percent error gaps using the Finite and Infinite Formulation, respectively, whereas Table 4 shows the average percent error gap using CPLEX, where in each case the best LB and UB found with the corresponding method was used to determine the gap. We observed that, consistently, the UB obtained by the Infinite Formulation was tighter than that obtained with the other methods. We therefore also report an improved bound on the actual gap achieved with the Finite Formulation and CPLEX that uses the best UB found using the Infinite Formulation; these are provided in parentheses in Tables 2 and 4).

From these tables, we may immediately conclude that the performance of the Infinite Formulation is far superior to that of the Finite Formulation and CPLEX in terms of solution quality obtained within the specified time limit. In fact,

Table 3. Average percent error gaps obtained with the Infinite Formulation using its own upper bound.

# Items	Pricing strategy					
	Constant			Dynamic		
	10	20	30	10	20	30
Capacity						
Low	0.79	0.37	0.25	0.65	0.33	0.21
Medium	0.72	0.29	0.20	0.48	0.23	0.12
High	0.49	0.21	0.15	0.34	0.16	0.11
Setup time						
Low	0.47	0.20	0.13	0.32	0.15	0.09
High	0.86	0.38	0.27	0.67	0.33	0.20
TBO						
Low	0.44	0.21	0.14	0.24	0.13	0.10
Medium	0.61	0.27	0.19	0.43	0.21	0.12
High	0.95	0.40	0.27	0.81	0.38	0.23

CPLEX was able to find a better integral solution than the Infinite Formulation in only 56 of the 540 instances with a dynamic pricing strategy (all with 10 items) and none of the 540 instances with a static pricing strategy. In contrast, CPLEX was able to find a better integral solution than the Finite Formulation in 490 of the 540 instances with a dynamic pricing strategy and 352 of the 540 instances with a constant pricing strategy.

Moreover, as already mentioned, the upper bound obtained using the Infinite Formulation is superior to the one obtained

Table 4. Average percent error gaps obtained with CPLEX using its own upper bounds (and using the upper bound found with the Infinite Formulation).

# Items	Pricing strategy					
	Constant			Dynamic		
	10	20	30	10	20	30
Capacity						
Low	3.85 (1.14)	5.22 (1.18)	5.52 (1.28)	3.37 (0.58)	4.34 (0.68)	4.47 (0.71)
Medium	3.75 (0.86)	4.46 (0.89)	4.66 (0.95)	3.29 (0.52)	3.83 (0.59)	3.89 (0.59)
High	2.96 (0.62)	3.48 (0.70)	3.71 (0.84)	2.80 (0.42)	3.13 (0.46)	3.09 (0.46)
Setup time						
Low	2.41 (0.61)	3.00 (0.63)	3.15 (0.67)	2.37 (0.37)	2.78 (0.41)	2.84 (0.44)
High	4.63 (1.14)	5.78 (1.20)	6.10 (1.36)	3.93 (0.64)	4.75 (0.74)	4.79 (0.73)
TBO						
Low	2.69 (0.58)	3.25 (0.62)	3.59 (0.83)	2.59 (0.36)	3.05 (0.38)	3.17 (0.44)
Medium	3.50 (0.80)	4.12 (0.82)	4.36 (0.92)	3.17 (0.48)	3.66 (0.52)	3.67 (0.52)
High	4.37 (1.24)	5.79 (1.29)	5.93 (1.27)	3.69 (0.68)	4.58 (0.83)	4.61 (0.81)

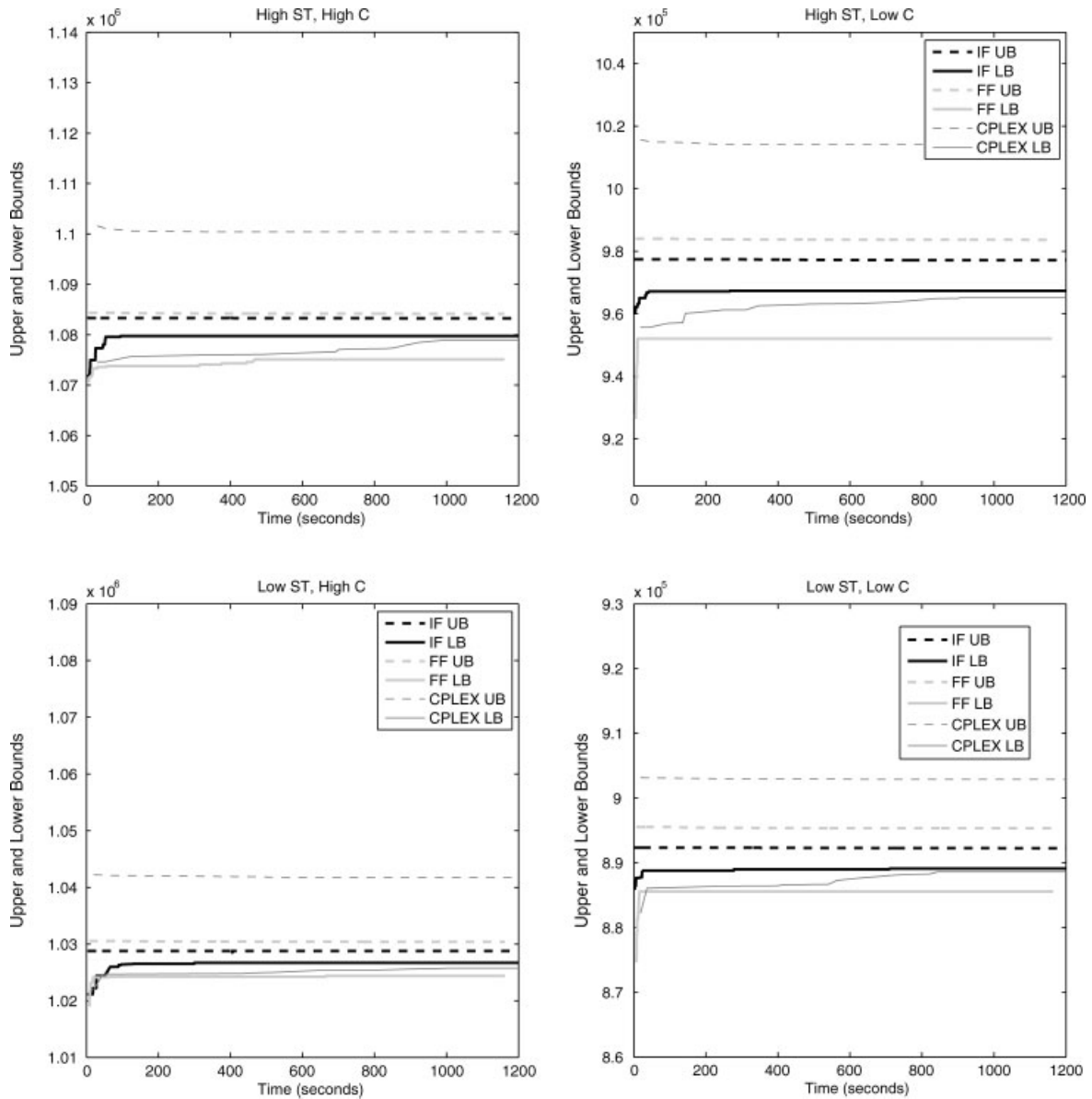


Figure 1. Comparison of branch-and-price and CPLEX performance on a representative set of test problems with 10 items, low TBO, and constant pricing.

with the other methods, which has the additional advantage that a much more accurate assessment of solution quality is obtained. The error gap tends to decrease as the number of items increases for the Infinite Formulation, as well as for the Finite Formulation under constant prices, while the reverse is true for the Finite Formulation under dynamic prices and for CPLEX. The former result was observed for the CLST by Degraeve and Jans [3] and Trigeiro et al. [15]. That result was expected since Manne [13] showed that the linear relaxation of the Dantzig-Wolfe formulation of this problem is a good approximation to the integer problem, whenever the number of items is large when compared with the number of capacity

constraints. It is interesting that this result, at least empirically, seems to extend to the branch-and-price approaches to the CLSTP. A possible explanation for the seemingly contradictory observation for the Finite Formulation under dynamic prices is that, under that formulation, the master problem is a nonlinear programming problem. Apparently, the increased difficulty of the master problem outweighs the benefits of the increased tightness of the master problem for larger problem instances (i.e., ones with larger number of items and dynamic pricing). On the basis of our results, we conjecture that the tightness of the continuous relaxation of the traditional formulation behaves in the opposite direction, which

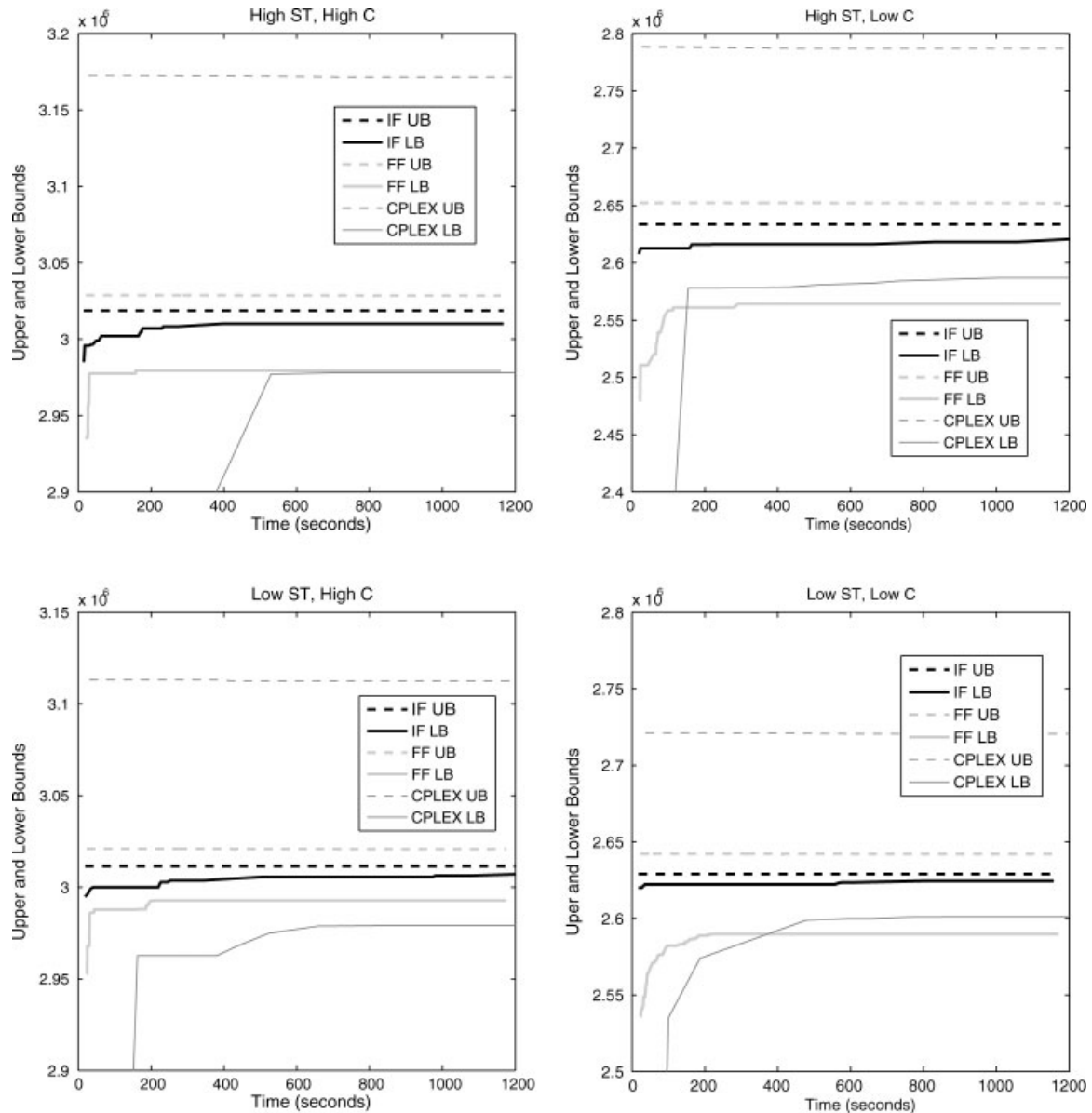


Figure 2. Comparison of branch-and-price and CPLEX performance on a representative set of test problems with 30 items, high TBO, and constant pricing.

would explain the increased error gaps as the number of items increases.

In general, the error gap increases as capacities get tighter, average setup times increase, and TBO increases. We also observe that the effects of capacity, setup time, and TBO are more pronounced when the number of items is small. Moreover, it is interesting to note that the error gaps are lower under dynamic prices than under constant prices when using the Infinite Formulation while the reverse is true when using the Finite Formulation. Finally, we observe that the advantage of our branch-and-price algorithm with the Infinite Formulation over CPLEX increases as problems become more difficult

to solve (i.e., as capacities get tighter, average setup times increase, and TBO increases). This is particularly apparent under a constant pricing strategy, high TBO values, and a larger number of items.

The aforementioned results all used a fixed running time of 1200 s. To assess the rate of convergence of our branch-and-price algorithms and CPLEX, for several representative instances, we create a plot that tracks the upper and lower bounds found by the algorithms as time progresses. Figures 1–4 show the plots on a representative set of test instances with 10 and 30 items, respectively. The test instances for the figures were selected to show the behavior of the solution

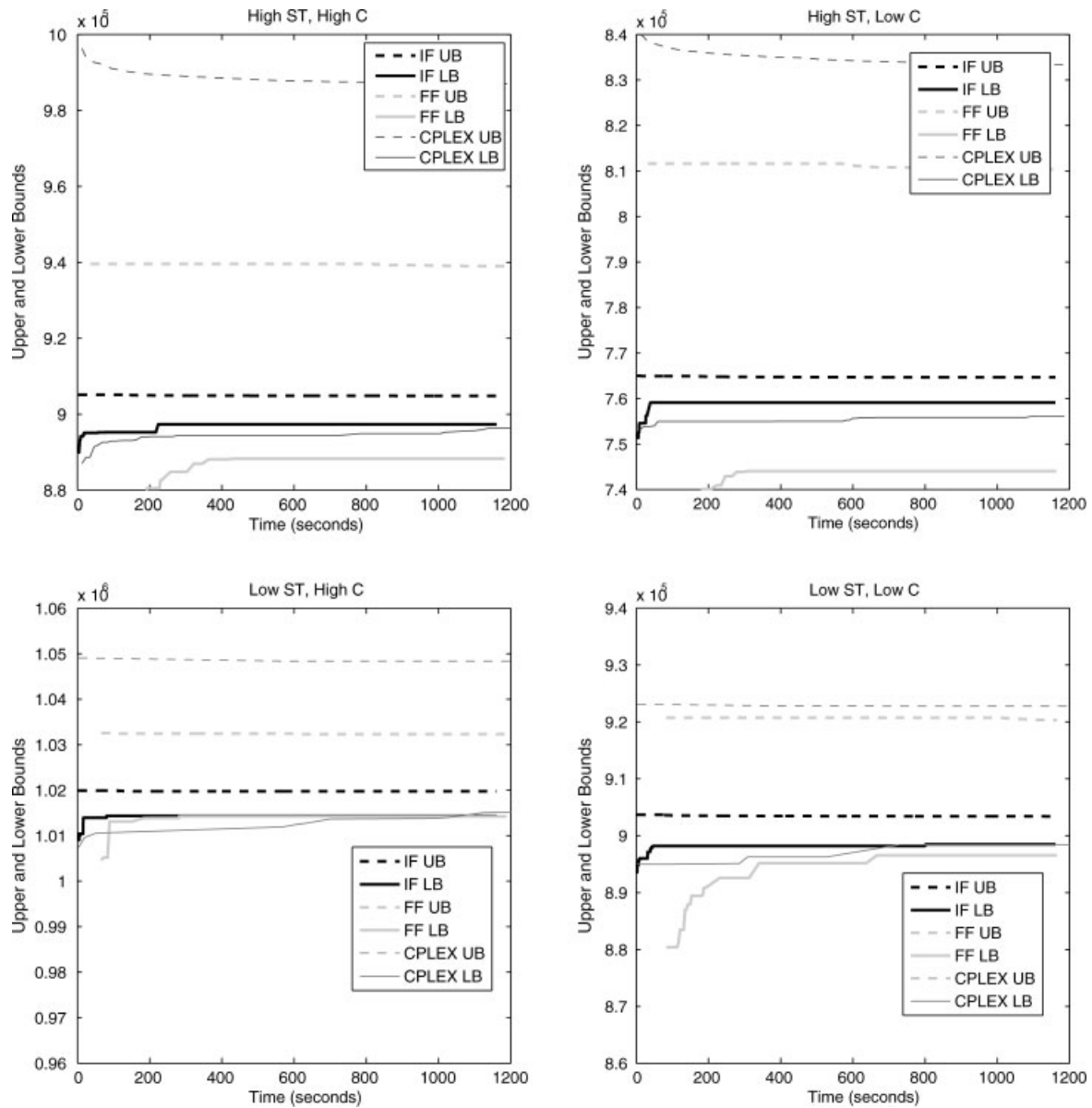


Figure 3. Comparison of branch-and-price and CPLEX performance on a representative set of test problems with 10 items, high TBO, and dynamic pricing.

approaches on a range of combinations of TBO, setup time, and capacity values. The caption with each table indicates the number of items, low/high TBO, and the pricing strategy used. Moreover, each figure is labeled with the remaining problem characteristics; e.g., “low ST, high C” represents an instance with low setup times and high capacities. In all figures, the solid lines represent lower bounds and the dashed lines upper bounds. The bounds for the branch-and-price algorithm with the Infinite Formulation (IF) and the Finite Formulation are drawn in black and grey bold lines, respectively, whereas the bounds for CPLEX are drawn in a thin line. We see from the figures that the branch-and-price

algorithm using the Infinite Formulation finds high-quality integer solutions fast and dominates CPLEX in terms of lower and upper bounds over time. These figures also suggest that the branch-and-price algorithm based on the Infinite Formulation performs much better than CPLEX on instances with a larger number of items.

Finally, to quantify the effect of the choice of pricing strategy on the optimal profit, we compared the lower bounds obtained by the Infinite Formulation with both a constant and a dynamic pricing strategy. In our test problems, the dynamic pricing strategy enjoyed an overall average additional profit of 3.54% over the profit obtained with a constant pricing

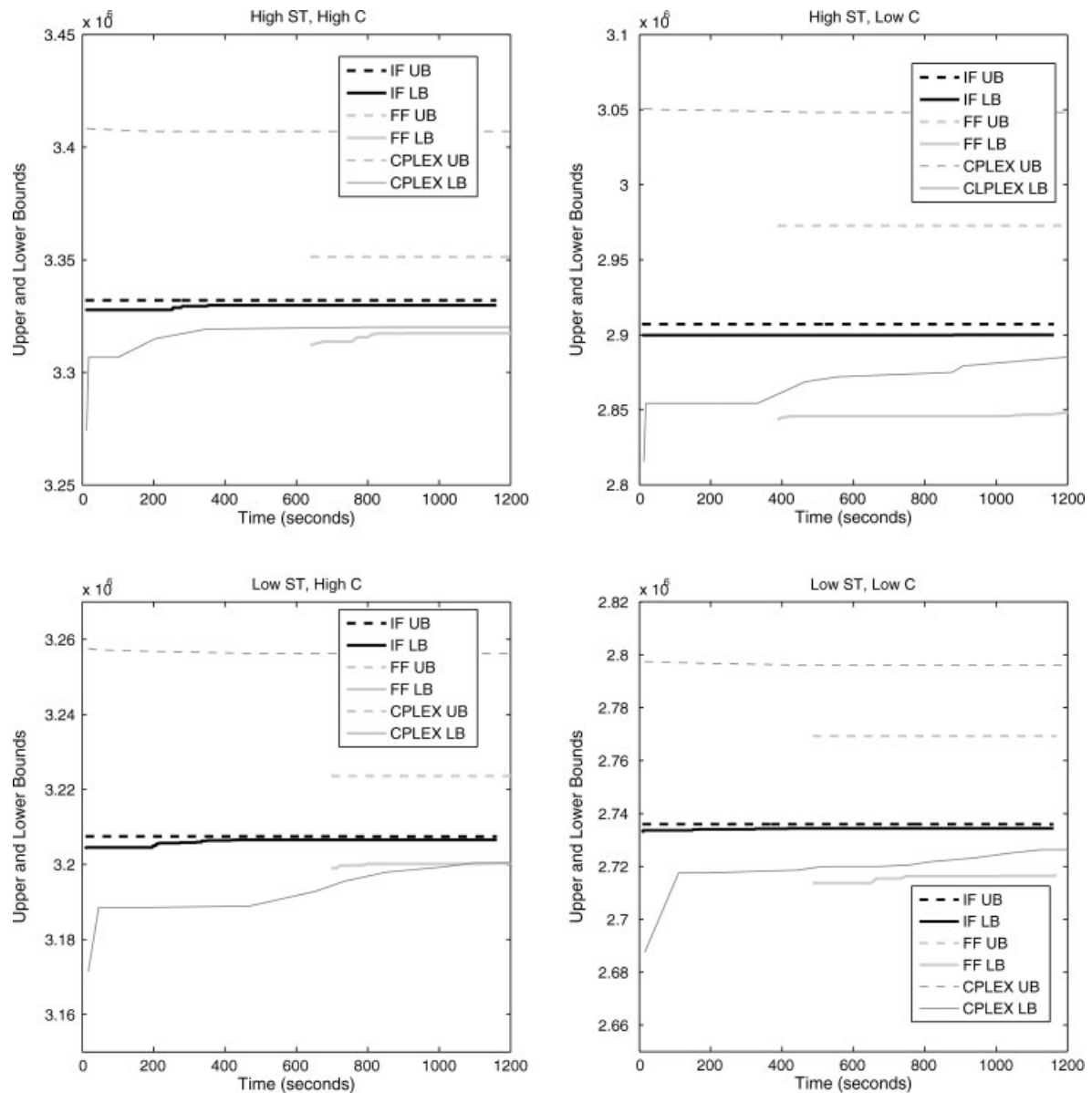


Figure 4. Comparison of branch-and-price and CPLEX performance on a representative set of test problems with 30 items, low TBO, and dynamic pricing.

strategy. Table 5 breaks this profit increase down by the different characteristics of the problem instances. In particular, we conclude that the dynamic pricing strategy is particularly profitable when capacities are low and setup times and TBO are high. This corresponds with what one might intuitively expect: For any fixed demand vector, total costs tend to be larger when setup times are larger because some of the available limited capacity is used for setups. Moreover, the resulting use of capacity for setup time reduces the number of feasible production plans. A similar argument can be made for the costliness of reduced capacities and larger TBO. In these cases, the ability to control prices and hence affecting the

demands provides a great deal of flexibility to the producer. In particular, dynamic prices can serve to better match demands to variable and limited capacities. Since constant prices provide a smaller degree of flexibility the difference between the profit under the different pricing strategies is amplified when capacity is smaller and setup times and TBO are larger.

7. CONCLUSION AND FUTURE RESEARCH

In this article, we considered the capacitated lot-sizing problem with setup times where we allow for demand flexibility through pricing decisions, considering both dynamic

Table 5. Percentage increase in profit when a dynamic pricing strategy instead of a constant pricing strategy is used.

# Items	10	20	30
Capacity			
Low	3.80	3.99	4.15
Medium	3.37	3.58	3.71
High	2.83	3.17	3.38
Setup time			
Low	3.12	3.34	3.49
High	3.49	3.79	3.99
TBO			
Low	2.94	3.27	3.54
Medium	3.35	3.49	3.68
High	3.62	3.93	3.99

and static pricing strategies. We developed two alternative Dantzig-Wolfe formulations for the problem, each of which leads to an exact branch-and-price algorithm. The Finite Formulation is guaranteed to terminate finitely, at the expense of a convex master problem. The alternative Infinite Formulation only finds a solution to within a user-specified tolerance in finite time, but has a linear master problem. In both formulations, the pricing problem generates columns by solving a single item lot-sizing subproblem with pricing decisions. For problem instances without initial inventories, efficient algorithms exist in the literature under both pricing strategies. We have developed new algorithms for the more general cases where nonzero initial inventories are present. Results of extensive computational tests showed that the Infinite Formulation outperforms both the Finite Formulation and CPLEX, finding high-quality solutions and tight error bounds in reasonable time.

Future research on the CLSTP could include a study of other pricing strategies as well as a further study of static pricing allowing for more general demand models. Finally, it would be interesting to study the polyhedral properties of the traditional formulation of the problem.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant no. DMI-0355533.

REFERENCES

- [1] G.B. Dantzig, "General convex objective forms," *Mathematical methods in the social sciences*, K.J. Arrow, S. Karlin, and P. Suppes (Editors), Stanford University Press, Stanford, California, 1960, pp. 151–158.
- [2] G.B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, NJ, 1963.
- [3] Z. Degraeve and R. Jans, A new dantzig-wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times, *Oper Res* 55 (2007), 909–920.
- [4] B.P. Dzielinski and R.E. Gomory, Optimal programming of lot sizes, inventory and labor allocations, *Manage Sci* 11 (1965), 874–890.
- [5] J. Geunes, Y. Merzifonluoğlu, and H.E. Romeijn, Capacitated procurement planning with price-sensitive demand and general concave revenue functions, *Eur J Oper Res* 194 (2009), 390–405.
- [6] J. Geunes, H.E. Romeijn, and K. Taaffe, Requirements planning with dynamic pricing and order selection flexibility, *Oper Res* 54 (2006), 394–401.
- [7] S.M. Gilbert, Coordination of pricing and multi-period procurement for constant priced goods, *Eur J Oper Res* 114 (1999), 330–337.
- [8] S.M. Gilbert, Coordination of pricing and multiple period procurement across multiple constant priced goods, *Manage Sci* 46 (2000), 1602–1616.
- [9] W. van den Heuvel and A.P.M. Wagelmans, A polynomial time algorithm for a deterministic joint pricing and inventory model, *Eur J Oper Res* 170 (2006), 463–480.
- [10] K.S. Hindi, K. Fleszar, and C. Charalambous, An effective heuristic for the clsp with set-up times, *J Oper Res Soc* 54 (2003), 490–498.
- [11] H. Kunreuther and L. Schrage, Joint pricing and inventory decisions for constant priced items, *Manage Sci* 19 (1973), 732–738.
- [12] L.S. Lasdon, *Optimization theory for large systems*, Macmillan, New York, 1970.
- [13] A.S. Manne, Programming of economic lot sizes, *Manage Sci* 4 (1958), 115–135.
- [14] J. Thomas, Price-production decisions with deterministic demand, *Manage Sci* 16 (1970), 747–750.
- [15] W.W. Trigeiro, J.L. Thomas, and J.O. McClain, Capacitated lot sizing with setup times, *Manage Sci* 35 (1989), 353–366.
- [16] A. Wagelmans, S. van Hoesel, and A. Kolen, Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case, *Oper Res* 40-S1 (1992), S145–S156.
- [17] H.M. Wagner and T.M. Whitin, Dynamic version of the economic lot size model, *Manage Sci* 5 (1958), 89–96.