The economic lot-sizing problem with an emission constraint

Mathijn J. Retel Helmrich[*,1], Raf Jans[2], Wilco van den Heuvel[1], and Albert P. M. Wagelmans[1]

[1]Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, Netherlands

[2]HEC Montréal, 3000, Chemin de la Côte-Sainte-Cathérine, Montréal, QC H3T 2A7, Canada

3 May 2012

ECONOMETRIC INSTITUTE REPORT EI 2011-41

Abstract

We consider a generalisation of the lot-sizing problem that includes an emission constraint. Besides the usual financial costs, there are emissions associated with production, keeping inventory and setting up the production process. Because the constraint on the emissions can be seen as a constraint on an alternative cost function, there is also a clear link with bi-objective optimisation. We show that lot-sizing with an emission constraint is $\mathcal{NP}$-hard and propose several solution methods. First, we present a Lagrangian heuristic to provide a feasible solution and lower bound for the problem. For costs and emissions for which the zero inventory property is satisfied, we give a pseudo-polynomial algorithm, which can also be used to identify the complete Pareto frontier of the bi-objective lot-sizing problem. Furthermore, we present a fully polynomial time approximation scheme (FPTAS) for such costs and emissions and extend it to deal with general costs and emissions. Special attention is paid to an efficient implementation with an improved rounding technique to reduce the a posteriori gap, and a combination

*Corresponding author, e-mail: retelhelmrich@ese.eur.nl

of the FPTASes and a heuristic lower bound. Extensive computational tests show that the Lagrangian heuristic gives solutions that are very close to the optimum. Moreover, the FPTASes have a much better performance in terms of their gap than the a priori imposed performance, and, especially if the heuristic's lower bound is used, they are very fast.

1 Introduction

In recent years, there has been a growing tendency to not only focus on financial costs in a production process, but also on its impact on society. This societal impact includes for instance the environmental implications, such as the emissions of pollutants during production. Particular interest is paid to the emission of greenhouse gases, such as carbon dioxide (CO_2), nitrous oxide (N_2O) and methane (CH_4). By now, there is a general consensus about the effect that these gases have on global warming. Consequently, many countries strive towards a reduction of these greenhouse gases, as formalised in treaties, such as the Kyoto Protocol (United Nations, 1998), as well as in legislation, of which the European Union Emissions Trading System (European Commission, 2010) is an important example.

The shift towards a more environmentally friendly production process can be caused by such legal restrictions, but also by a company's desire to pursue a 'greener' image by reducing its carbon footprint. As Vélazquez-Martínez et al. (2011) mention: "A substantial number of companies publicly state carbon emission reduction targets. For instance, in the 2011 Carbon Disclosure Project annual report (Carbon Disclosure Project, 2011), 926 companies publicly commit to a self-imposed carbon target, such as FedEx, UPS, Wal-Mart, AstraZeneca, PepsiCo, Coca-Cola, Danone, Volkswagen, Campbell and Ericsson."

Emissions could be reduced by for instance using less polluting machines or vehicles, or using cleaner energy sources. One should not overlook the potential benefit that changing operational decisions has on emission reduction. There is no guarantee that minimising costs of operations will also lead to low emissions. In fact, fashionable production strategies like just-in-time production, with its frequent less-than-truckload shipments and frequent change-overs on machines, may lead to emission levels that are far from optimal.

For these reasons, the classic economic lot-sizing model has been generalised. Besides the usual financial costs, there are emission levels associated with production, keeping inventory and setting up the production process. Set-up emissions can for example originate from having fixed per-truckload emissions of an order, or from a production process that needs to 'warm up', where usable products are not created until the production process has gone through a set-up phase that is already polluting. If products need to be stored in a specific way, e.g. refrigerated, then keeping inventory will also emit pollutants. The lot-sizing model that we consider in this paper minimises the (financial) costs under an emission constraint. This constraint can be seen as one global restriction over all periods. This problem was introduced by Benjaafar et al. (2011), who integrate carbon emission constraints in lot-sizing models in several ways. They consider a capacity on the total emissions over the entire problem horizon, as we do in this paper, but also a carbon tax, a capacity combined with emissions trade, or carbon offsets (where additional emission rights may be bought, but not sold). Moreover, they study the effect of collaboration between multiple firms within a serial supply chain. Several insights are derived from the models by experimenting with the problem parameters. They assume that all cost and emission functions follow a fixed-plus-linear structure, and no attention is paid to finding good solution methods yet.

In our paper, we study a lot-sizing problem with an emission constraint under concave cost and emission functions. We will see that this model is also capable of handling multiple production modes. We show that this problem is $\mathcal{NP}$-hard, even if only production emits pollutants (linearly). Moreover, we show that lot-sizing with an emission constraint and two production modes in each period is $\mathcal{NP}$-hard, even if only production emits pollutants (linearly) and either all (financial) costs or all emissions are time-invariant. Then, we develop several solution methods. First, we give a Lagrangian heuristic that finds both very good solutions and a lower bound in $\mathcal{O}(T^4)$ time, where T is the number of time periods. We also prove several structural properties of an optimal solution that we use while working towards a fully polynomial time approximation scheme (FPTAS). As a first step, a pseudo-polynomial algorithm is developed in case the costs and emissions are such that the single-sourcing (zero invertory) property is satisfied. This pseudo-polynomial algorithm is then turned into an FPTAS, which, in turn, is generalised to deal with costs and emissions that do not satisfy the single-sourcing property. We expect that this technique to construct a pseudo-polynomial algorithm and an FPTAS can be applied to more problems where one overall capacity constraint is added to a problem for which a polynomial time dynamic programme exists.

Special attention is paid to an efficient practical implementation of these algorithms. This includes a combination of the lower bound that is provided by the Lagrangian heuristic with an FPTAS, which results in excellent solutions within short computation times, as becomes clear from the extensive computational tests of all algorithms that have been carried out for this paper. Besides that, our algorithms do not only have an a priori gap (ε), but they also produce a (smaller) a posteriori gap. To reduce this gap even further, we develop an improved rounding technique, which we think can be applied to other FPTASes of the same type. Furthermore, if we compare the algorithms' solutions to the optima, we see that the gaps are even much smaller.

The model is more general than it looks at first sight, since the emission costs that we consider do not necessarily need to refer to emissions. They can be *any* kind of costs or output, other than those in the objective function, related to the three types of decisions (i.e., set-up, production and inventory). This makes the relationship with bi-objective lot-sizing clear. In multi-objective optimisation (and bi-objective optimisation in particular), one is usually interested in the frontier of Pareto optimal solutions. Theoretically, finding the optimal costs for all possible emission capacities would result in finding the Pareto frontier. The multi-objective lot-sizing problem is studied in more detail by Van den Heuvel et al. (2011), who divide the horizon in several blocks, each with its own objective function. The case with one block of length T corresponds to our problem (with fixed-plus-linear costs and emissions). In our paper, we will show that we can find the whole Pareto frontier in pseudo-polynomial time, if the costs and emissions are such that the single-sourcing (zero-inventory) property is satisfied.

Besides the works of Benjaafar et al. (2011) and Van den Heuvel et al. (2011), there are some other papers that integrate carbon emission constraints in lot-sizing problems. Absi et al. (2011) introduce lot-sizing models with emission constraints of several types: periodic, cumulative, global (as we have) and rolling. Furthermore, they consider multiple production modes, one of which is 'ecological'. As mentioned, our model can also handle multiple production modes. Vélazquez-Martínez et al. (2011) study the effect of different levels of aggregation to estimate the transportation carbon emissions in the economic lot-sizing model with backlogging. Heck and Schmidt (2010) discuss lot-sizing with an 'eco-term', which they solve heuristically with 'eco-enhanced' Wagner-Whitin and Part Period Balancing, with the possibility of 'eco-balancing'. Other papers approach the emission problem from an EOQ point of view, such as Chen et al. (2011), Hua et al. (2011) and Bouchery et al. (2010).

The remainder of this paper is organised as follows. The next section provides a formal, mathematical definition of the lot-sizing problem with a global emission constraint. In Section 3, we show that this problem, as well as a variant with two production modes, is $\mathcal{NP}$-hard under quite general conditions. In Section 4, we prove several structural properties of an optimal solution, which are used by the algorithms that are introduced in Section 5. Section 5.1 gives a Lagrangian heuristic. Sections 5.2 and 5.3 present a pseudo-polynomial algorithm, respectively FPTAS, for what we will

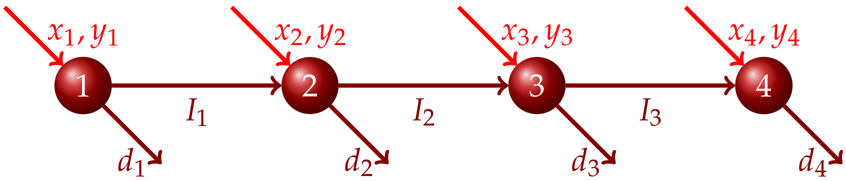

Figure 1: Graphical representation of a lot-sizing problem

define as *co-behaving* costs and emissions. An FPTAS for general costs and emissions is derived in Section 5.4. The combination of the heuristic and FPTASes is discussed in Section 5.5. Section 6 describes and gives the results of the extensive computational tests and the paper is concluded in Section 7.

2 Problem definition

The model can be formally defined as follows:

$$\min \quad \sum_{t=1}^{T} \left(p_t(x_t) + h_t(I_t)\right) \tag{1}$$

$$\begin{aligned}
\text{s.t.} \quad I_t &= I_{t-1} + x_t - d_t \quad && t = 1, \ldots, T && (2)\\
I_0 &= 0 && && (3)\\
x_t, I_t &\geq 0 && t = 1, \ldots, T && (4)\\
\sum_{t=1}^{T} \Big(\hat{p}_t(x_t) + \hat{h}_t(I_t)\Big) &\leq \hat{C} \quad , && && (5)
\end{aligned}$$

where x_t is the quantity produced in period t, and I_t is the inventory at the end of period t. The demand in period t is given by d_t, the length of the problem horizon is T, and $\hat{C}$ is the emission capacity. Furthermore, p_t and h_t are production and holding costs functions, and $\hat{p}_t$ and $\hat{h}_t$ are production and holding emission functions, respectively. We assume that all functions are concave, nondecreasing and nonnegative. This includes the well-known case with fixed set-up costs and linear production and holding costs.

Equation (2) gives the inventory balance constraints. There is no initial inventory (3); the nonnegativity constraints are given by (4), and (5) constrains the emissions over the whole problem horizon. We shall refer to problem (1)–(5) as ELSEC (Economic Lot-Sizing with an Emission Constraint).

Of course, $\hat{p}_t$ and $\hat{h}_t$ don't necessarily refer to emissions. They can be any kind of costs other than those in the objective function. Examples of what can be modelled by $\hat{p}_t$ and $\hat{h}_t$ include other types of negative externalities for society, such as other pollutants or noise resulting from production or carrying inventories. Moreover, we can impose a maximum on the total or average inventory by choosing $\hat{h}_t(I_t) = I_t$ and $\hat{p}_t(x_t) = 0$ for all t, and $\hat{C}$ equal to the total inventory or T times the average inventory. Also, we can model a lot-sizing problem with m production modes and T periods by defining an instance of ELSEC with Tm periods, where periods appear in groups of m, such that each of these periods corresponds to another production mode, with zero holding costs within such a group and where demand occurs only in the last of a group of m periods.

If the costs and emissions follow a fixed-plus-linear structure, then the model can also be formulated as the standard mixed integer linear programme (6)–(12). We shall refer to this problem as ELSEC-MILP. See Figure 1 for a graphical representation with four periods.

$$\min \quad \sum_{t=1}^{T} (K_t y_t + p_t x_t + h_t I_t) \tag{6}$$

$$\begin{aligned}
\text{s.t.} \quad I_t &= I_{t-1} + x_t - d_t & t = 1, \ldots, T \quad (7)\\
x_t &\leq y_t \sum_{s=t}^{T} d_s & t = 1, \ldots, T \quad (8)\\
I_0 &= 0 & (9)\\
x_t, I_t &\geq 0 & t = 1, \ldots, T \quad (10)\\
y_t &\in \{0, 1\} & t = 1, \ldots, T \quad (11)\\
\sum_{t=1}^{T} \left(\hat{K}_t y_t + \hat{p}_t x_t + \hat{h}_t I_t \right) &\leq \hat{C} & (12)
\end{aligned}$$

K_t and $\hat{K}_t$ are the set-up cost and emissions, respectively. Now, p_t, $\hat{p}_t$, h_t and $\hat{h}_t$ refer to the *unit* production and holding costs and emissions. y_t is a binary variable indicating a set-up in period t and constraints (8) ensure that production can only take place if there is a set-up in that period.

3 Complexity results

Van den Heuvel et al. (2011) show that some special cases of ELSEC-MILP can be solved in polynomial time. Moreover, they show that ELSEC-MILP is $\mathcal{NP}$-complete in general, even if only set-ups emit pollutants and under Wagner-Whitin (non-speculative) costs and emissions.

In this section, we will show that another special case of ELSEC-MILP is $\mathcal{NP}$-hard.

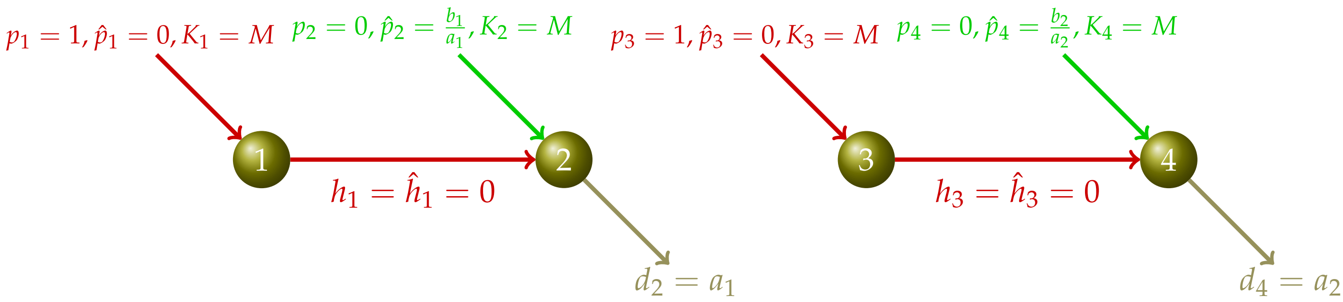

Figure 2: An instance of ELSEC-MILP that corresponds to an instance of KNAPSACK

We will see that a special case of lot-sizing with an emission constraint and two production modes is $\mathcal{NP}$-hard as well.

Theorem 1 *Lot-sizing with an emission constraint is $\mathcal{NP}$-hard, even if only production emits pollutants and these emissions are linear in the quantity produced.*

Proof We will show that KNAPSACK is a special case of ELSEC-MILP. KNAPSACK problem (decision version): given $a, b \in \mathbb{N}^n$ and $k, \hat{C} \in \mathbb{N}$, does there exist a vector $z \in \{0,1\}^n$ such that

$$\sum_{i=1}^{n} a_i z_i \geq k, \sum_{i=1}^{n} b_i z_i \leq \hat{C}?$$

Define the following instance of ELSEC-MILP (see Figure 2):

$$
\begin{array}{rclrcl}
T & = & 2n & d_t & = & \begin{cases} 0 & \text{for } t \text{ odd} \\ a_{\frac{1}{2}t} & \text{for } t \text{ even} \end{cases} \\
K_t & = & M \quad \forall t & \hat{K}_t & = & 0 \quad \forall t \\
h_t & = & \begin{cases} 0 & \text{for } t \text{ odd} \\ \infty & \text{for } t \text{ even} \end{cases} & \hat{h}_t & = & 0 \quad \forall t \\
p_t & = & \begin{cases} 1 & \text{for } t \text{ odd} \\ 0 & \text{for } t \text{ even} \end{cases} & \hat{p}_t & = & \begin{cases} 0 & \text{for } t \text{ odd} \\ \frac{b_{\frac{1}{2}t}}{a_{\frac{1}{2}t}} & \text{for } t \text{ even} \end{cases}
\end{array}
$$

where M is a very large number. Clearly, this reduction can be done in polynomial time. We will show that the answer to KNAPSACK is positive if and only if ELSEC-MILP has a solution with costs of at most $M \cdot n + \sum a_i - k$.

Suppose the answer to KNAPSACK is positive. Then if $z_i = 1$, let $x_{2i} = a_i$ and if $z_i = 0$, let $x_{2i-1} = a_i$; $x_t = 0$ otherwise. The thus created solution of ELSEC-MILP has costs:

$$M \cdot n + \sum_{i:z_i=1} x_{2i} p_{2i} + \sum_{i:z_i=0} x_{2i-1} \cdot p_{2i-1} = M \cdot n + \sum_{i:z_i=1} a_i \cdot 0 + \sum_{i:z_i=0} a_i \cdot 1$$

$$= M \cdot n + \sum_{i=1}^{n} a_i(1 - z_i) = M \cdot n + \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_i z_i \leq M \cdot n + \sum a_i - k.$$

Moreover, this solution of ELSEC-MILP has emissions:

$$\sum_{i:z_i=1} x_{2i}\hat{p}_{2i} + \sum_{i:z_i=0} x_{2i-1} \cdot \hat{p}_{2i-1} = \sum_{i:z_i=1} a_i \cdot \frac{b_i}{a_i} + \sum_{i:z_i=0} a_i \cdot 0 = \sum_{i=1}^{n} b_i z_i \leq \hat{C}.$$

Conversely, suppose ELSEC-MILP has a solution with costs of at most $M \cdot n + \sum a_i - k$. Then we know that there are at most n set-ups, otherwise the costs of ELSEC-MILP would be at least $M \cdot (n+1) > M \cdot n + \sum a_i - k$. Since $h_t = \infty$ for t even, there must be exactly one set-up in each pair of periods $(2i-1, 2i)$. Moreover, the production quantity in this period must be exactly a_i, to satisfy all demand. There is a budget of $\sum a_i - k$ left to pay for production costs. The production costs equal the sum of a_i over all i for which $x_{2i-1} = a_i$ (and $x_{2i} = 0$), so

$$\sum_{i:x_{2i-1}=a_i} a_i \cdot 1 + \sum_{i:x_{2i}=a_i} a_i \cdot 0 = \sum_{i:x_{2i-1}=a_i} a_i \leq \sum_{i=1}^{n} a_i - k.$$

It follows that

$$\sum_{i:x_{2i}=a_i} a_i \geq k.$$

Now, construct the following solution to KNAPSACK: if $x_{2i} = a_i$, then $z_i = 1$, and if $x_{2i-1} = a_i$ then $z_i = 0$. The profit of this solution equals

$$\sum_{i=1}^{n} a_i z_i = \sum_{i:x_{2i}=a_i} a_i \cdot 1 \geq k.$$

Since the solution of ELSEC-MILP is feasible (by assumption), the following holds for the emissions:

$$\sum_{i=1}^{n} b_i z_i = \sum_{i:x_{2i}=a_i} b_i = \sum_{i:x_{2i}=a_i} \frac{b_i}{a_i} a_i = \sum_{t \text{ even}} \frac{b_{\frac{1}{2}t}}{a_{\frac{1}{2}t}} x_t = \sum_{t=1}^{T} \hat{p}_t x_t = \sum_{t=1}^{T} \left(\hat{K}_t y_t + \hat{p}_t x_t + \hat{h}_t I_t \right) \leq \hat{C}.$$

$\square$

We can also view the instance from the proof as a lot-sizing problem with an emission constraint and two different production modes in each period, with a horizon of $\frac{1}{2}T$ periods. The even and odd periods then correspond to these two different production modes, and we get the following corollary.

Corollary 2 *Lot-sizing with an emission constraint and two production modes in each period is $\mathcal{NP}$-hard, even if only production emits pollutants (linearly) and either all (financial) costs or all emissions are time-invariant.*

4 Structural properties

Before we introduce our algorithms in Section 5, we prove the correctness of some structural properties of an optimal solution, which these algorithms will use.

We use the common definition of a *block* as an interval $[t, s]$ such that $I_{t-1} = I_s = 0$ and $I_\tau \neq 0 \ \forall t \leq \tau \leq s-1$. Furthermore, let a period t be called a *double-sourcing* period, if $I_{t-1} > 0$ and $x_t > 0$, that is, there is both inventory carried over from the previous period and positive production in period t. Let a period t be called a *single-sourcing* period if either $I_{t-1} = 0$ or $x_t = 0$.

Later, we will want to consider a given solution and find out what happens to the costs (and emissions) when we shift production from period i to period j and vice versa. Therefore, it will be convenient to make the following definitions. Let (x, I) be a given solution. Let $x_{i,j}$ be the quantity produced in period i that is kept in inventory until at least period j in that solution. Define $q_{i,j}$ as the *additional* production quantity in period i (compared to (x, I)) that is kept in inventory until at least period j. We can interpret x_i as the production quantity in period i in the 'old' (given) situation and $x_i + q_{i,j}$ as the production quantity in period i in the 'new' situation. Similarly, we can interpret the quantities $I_k + q_{i,j}$ as the inventories in periods k ($i \leq k \leq j-1$) in the 'new' situation. Now, define $C_{i,j}(q_{i,j}; x_i, I_i, \ldots, I_{j-1}) := p_i(x_i + q_{i,j}) + \sum_{k=i}^{j-1} h_k(I_k + q_{i,j})$. We will use $C_{i,j}(0)$ and $C_{i,j}$ as shorthand for $C_{i,j}(0; x_i, I_i, \ldots, I_{j-1})$. In this way, $C_{i,j}(0)$ gives the production costs in period i plus the holding costs incurred in periods i through $j-1$ in the 'old' situation, and $C_{i,j}(q_{i,j})$ gives the production and holding costs in the same periods in the 'new' situation. Because of concavity of p_i and h_k, it holds that $C_{i,j}$ is concave (in $q_{i,j}$) too. Note that $C_{j,j}(q_{j,j}) = p_j(x_j + q_{j,j})$. Similarly, define $\hat{C}_{i,j}(q_{i,j}; x_i, I_i, \ldots, I_{j-1}) := \hat{p}_i(x_i + q_{i,j}) + \sum_{k=i}^{j-1} \hat{h}_k(I_k + q_{i,j})$, and use $\hat{C}_{i,j}(0)$ and $\hat{C}_{i,j}$ as shorthand for $\hat{C}_{i,j}(0; x_i, I_i, \ldots, I_{j-1})$. Define

$$p_i'(x_i) := \lim_{h \downarrow 0} \frac{p_i(x_i + h) - p_i(x_i)}{h} \ ,$$

i.e., p_i' is the right derivative of p_i. Because p_i is real-valued and concave, we know that this right derivative exists for $x_i > 0$.

Similarly, let $\hat{p}_i', h_i', \hat{h}_i', C_{i,j}', \hat{C}_{i,j}'$ be the right derivatives of their respective functions. We know that the right derivative of $\hat{p}_i$ exists for $x_i > 0$, the right derivatives of h_i and $\hat{h}_i$ exist for $I_i > 0$, and the right derivatives of $C_{i,j}$ and $\hat{C}_{i,j}$ exist for $q_{i,j} + x_i > 0$ and $q_{i,j} + I_k > 0$ $(i \leq k < j)$ (i.e., the quantity that is produced *less* in period i is such that the remaining production quantity, respectively inventories are positive).

Theorem 3 *If, for each pair $i \leq j$, either $\Big(C_{i,j}'(q_{i,j}) \leq C_{j,j}'(q_{j,j})$ and $\hat{C}_{i,j}'(q_{i,j}) \leq \hat{C}_{j,j}'(q_{j,j})\Big)$ or $\Big(C_{i,j}'(q_{i,j}) \geq C_{j,j}'(q_{j,j})$ and $\hat{C}_{i,j}'(q_{i,j}) \geq \hat{C}_{j,j}'(q_{j,j})\Big)$ holds, for all (x, I) and all $(q_{i,j}, q_{j,j})$ (such* *that $q_{i,j} + x_i > 0$, $q_{j,j} + x_j > 0$ and $q_{i,j} + I_k > 0$ $(i \leq k < j)$), then there exists an optimal solution to ELSEC, such that the single-sourcing property holds in all periods.*

Proof Suppose there exists an optimal solution (x, I) with (at least) one double-sourcing period. Let v be a double-sourcing period. Suppose that period v's demand is procured from two periods, t and s, then it must be that either $v = t$ or $v = s$. Furthermore, assume that $C'_{t,v}(0) \geq C'_{s,v}(0)$ and $\hat{C}'_{t,v}(0) \geq \hat{C}'_{s,v}(0)$. (Note that this also covers the case $C'_{t,v}(0) \leq C'_{s,v}(0)$ and $\hat{C}'_{t,v}(0) \leq \hat{C}'_{s,v}(0)$, because we can switch the indices t and s.) Now, we should produce $x_{t,v}$ units in period s instead of period t, so that we obtain a solution with single-sourcing in period v. We show that this will decrease both costs and emissions. Because of concavity, it holds that

$$C_{t,v}(0) - C_{t,v}(-x_{t,v}) \geq C'_{t,v}(0)x_{t,v} \geq C'_{s,v}(0)x_{t,v} \geq C_{s,v}(x_{t,v}) - C_{s,v}(0) \ ,$$

i.e., the savings are larger than the extra expenses. Completely analogously,

$$\hat{C}_{t,v}(0) - \hat{C}_{t,v}(-x_{t,v}) \geq \hat{C}'_{t,v}(0)x_{t,v} \geq \hat{C}'_{s,v}(0)x_{t,v} \geq \hat{C}_{s,v}(x_{t,v}) - \hat{C}_{s,v}(0) \ .$$

If there are any double-sourcing periods left, then repeat the above procedure until there are only single-sourcing periods left. □

Corollary 4 *If both the financial and emission costs satisfy the* Wagner-Whitin *property (no speculative motives), then there exists an optimal solution to ELSEC, such that the single-sourcing property holds in all periods.*

Proof By definition, the Wagner-Whitin property means that it is cheapest to procure products from the most recent production period, i.e. $\left(C'_{i,j} \geq C'_{j,j} \text{ and } \hat{C}'_{i,j} \geq \hat{C}'_{j,j}\right)$ for all $i \leq j$. □

Note that in our model the single-sourcing property is the same as the zero inventory (ZIO) property, i.e., there exists an optimal solution such that $I_{t-1} = 0$ or $x_t = 0$ for all periods t. In the remainder of this paper, we will refer to all financial and emission costs that satisfy the conditions in Theorem 3 as *co-behaving*, because over time, such cost and emission functions move in the same direction, i.e., if one increases (decreases), the other increases (decreases) as well.

The following corollary is a direct consequence of Theorem 3:

Corollary 5 *If the emission cost functions are time-invariant and the holding emissions are zero, OR the financial cost functions are time-invariant and the holding costs are zero, then there exists a solution to ELSEC, such that the single-sourcing property holds in all periods.*

In general, the following property holds:

Theorem 6 *There exists an optimal solution to ELSEC, such that the single-sourcing property holds in all but (at most) one period.*

Proof See Appendix A. □

We will refer to the period in which the single-sourcing period is violated as the *double-sourcing period*. In this period, say v, it holds that both $I_{v-1} > 0$ and $x_v > 0$.

Finally, we prove the next property, which is used in Section 5.4.

Theorem 7 *There exists an optimal solution in which either the full emission capacity is used, or the single-sourcing property holds.*

Proof We need to show that if we have a solution with double-sourcing for which the emission capacity is not fully used, i.e. $\sum_{t=1}^{T}\left(\hat{p}_t(x_t) + \hat{h}_t(I_t)\right) < \hat{C}$, then there exists a solution with equal or lower costs and emissions that uses the full capacity or does not have double-sourcing in any period.

Let period v's demand be produced in periods t and s, where either $t = v$ or $s = v$. Assume that $C'_{t,v}(0) \geq C'_{s,v}(0)$, w.l.o.g. It is cheaper to move a quantity $q > 0$ from period t to period s, since because of concavity, it holds that

$$C_{t,v}(0) - C_{t,v}(-q) \geq C'_{t,v}(0)q \geq C'_{s,v}(0)q \geq C_{s,v}(q) - C_{s,v}(0) \ ,$$

i.e., the savings are larger than the extra expenses.

Try to choose $q = x_{t,v}$, so that we obtain a solution that satisfies the single-sourcing property. If the emissions of the new solution are within the emission capacity, then we are done.

Otherwise, choose $0 < q < x_{t,v}$, such that the additional emissions equal the remaining emission capacity, i.e., $\hat{C}_{s,v}(q) - \hat{C}_{s,v}(0) + \hat{C}_{t,v}(0) - \hat{C}_{t,v}(-q) = r$, where $r > 0$ is this remaining capacity. Existence of such a q follows from the mean-value theorem, since $\hat{C}_{t,v}$ and $\hat{C}_{s,v}$ are continuous on their interior domains. □

5 Algorithms

We propose several algorithms to solve ELSEC. First, we present a Lagrangian heuristic that provides an upper and lower bound for the problem. Secondly, we develop an exact algorithm that solves the co-behaving version of ELSEC in pseudo-polynomial time. We turn this algorithm into a fully-polynomial approximation scheme (FPTAS). Next, this FPTAS is extended to deal with more general cost and emission functions. Finally, we show how the FPTASes can be sped up by using a lower bound, such as the one given by the Lagrangian heuristic.

5.1 Lagrangian heuristic

In this section, we present a Lagrangian heuristic that is based on relaxation of the emission capacity constraint (5). The resulting formulation is given below. This heuristic will give us both a lower bound and a feasible solution.

$$\begin{aligned} \min \quad & \sum_{t=1}^{T} (p_t(x_t) + h_t(I_t)) + \lambda \sum_{t=1}^{T} \left(\hat{p}_t(x_t) + \hat{h}_t(I_t) - \hat{C} \right) \\ = \quad & \sum_{t=1}^{T} \left(p_t(x_t) + \lambda \hat{p}_t(x_t) + h_t(I_t) + \lambda \hat{h}_t(I_t) \right) - \lambda \hat{C} \end{aligned} \tag{13}$$

$$\begin{aligned} \text{s.t.} \quad I_t &= I_{t-1} + x_t - d_t \quad t = 1, \ldots, T && (14) \\ x_t, I_t &\geq 0 \qquad\qquad\qquad\quad t = 1, \ldots, T && (15) \\ I_0 &= 0 && (16) \\ \lambda &\geq 0 && (17) \end{aligned}$$

First, suppose that λ is given. Obviously, constraints (14)–(16) are the same as in the classic (uncapacitated, single-item) lot-sizing problem. Moreover, $p_t + \lambda \hat{p}_t$ is a concave function of x_t, because both p_t and $\hat{p}_t$ are concave, and λ is nonnegative. Similarly, $h_t + \lambda \hat{h}_t$ is a concave function of I_t. Furthermore, $\lambda \hat{C}$ is a constant, so we can ignore it when optimising. Hence, for a given λ, the relaxed problem (13)–(16) is a classic lot-sizing problem and we can solve it with Wagner and Whitin (1958)'s algorithm.

For any $\lambda \geq 0$, the optimal value of (13) gives a lower bound on ELSEC. Naturally, we are looking for the best (that is, highest) lower bound. As output, our algorithm will give an interval that contains the λ^* for which this best lower bound is attained. It is easy to see that for $\lambda = 0$, the emission constraint (5) will be violated in general. Otherwise, the problem can be solved by simply ignoring the emissions and minimising costs. If λ is increased, then step by step, the emissions will decrease and the costs will increase. For some value of λ, say λ_{UB}, the solution will satisfy the emission constraint (5) (provided that a feasible solution exists). We are interested in finding the highest value of λ, say λ_{LB}, for which the solution of (13)–(16) violates the emission constraint (5). This gives our best lower bound.

We apply Megiddo (1979)'s algorithm for combinatorial problems that involve minimisation of a rational objective function to the lot-sizing problem. Gusfield (1983) showed that this is equivalent to minimising an objective of the form $a + \lambda b$. See also Wagelmans (1990) and Megiddo (1983). These papers imply that if, for a given λ, the relaxed problem can be solved in $\mathcal{O}(A)$ (with a 'suitable' algorithm) and we can check in $\mathcal{O}(B)$ whether the relaxed constraint is violated or not, then the parametrised problem ($a + \lambda b$) can be solved in $\mathcal{O}(AB)$. For a given λ, our relaxed problem (13)–(16) can be solved in $\mathcal{O}(T^2)$ with Wagner-Whitin. Moreover, the same algorithm can be used to determine whether the emission constraint is violated or not. Although Megiddo (1979) only mentions fractions of linear functions, his algorithm can be generalised to our problem in a straightforward manner. Hence, we can solve our Lagrangian relaxation in $\mathcal{O}(T^2T^2) = \mathcal{O}(T^4)$.

The intuition behind the algorithm is as follows. We are looking for an interval such that λ^* equals one of the endpoints. At λ^*, we are indifferent between two solutions, of which one is infeasible and the other feasible. The latter will give us an upper bound. A trivial initial choice for the interval is $[0, \infty)$. We act as if we know λ^*, and solve (13)–(16) with Wagner-Whitin. View this algorithm as a decision tree. At each node of the tree, we need to make a decision, say to 'go left' or 'go right'. This decision depends on a comparison of the form $a(X^1) + \lambda b(X^1) \leq a(X^2) + \lambda b(X^2)$, where a and b are a cost and an emission function, respectively, and X^1 and X^2 are (partial) solutions. Suppose we go left if the statement is true and right otherwise. We compute for which λ we are indifferent. For this λ, we can solve the relaxed problem in $\mathcal{O}(T^2)$ with Wagner-Whitin and know whether the solution is feasible. If so, then this λ provides an upper bound on our interval; if not, it provides a lower bound. Note that for all λ inside the (updated) interval, we make the same decisions in each of the decisions nodes that we already visited. Take a λ inside this interval and check whether $a(X^1) + \lambda b(X^1) \leq a(X^2) + \lambda b(X^2)$ to know if we should go left or right. We continue in this manner until the last step of the algorithm.

Below, we give pseudocode for Megiddo (1979)'s algorithm applied to our problem.

$\lambda_{LB} := 0,\ \lambda_{UB} := \infty,\ m(T+1) := 0,\ \hat{m}(T+1) := 0$

`for` $t = T$ `until 1 step -1 do`

  `MinimumCosts` $:= \infty$, `MinimumEmissions` $:= \infty$

  `for` $s = t$ `until` T `step 1 do`

    `Costs` $:= c(t,s) + m(s+1)$

    `Emissions` $:= e(t,s) + \hat{m}(s+1)$

    `if MinimumCosts` $< \infty$ `and MinimumEmissions` $< \infty$ `and Emissions` $\neq$ `MinimumEmissions then`

      $\lambda := \frac{\texttt{MinimumCosts}-\texttt{Costs}}{\texttt{Emissions}-\texttt{MinimumEmissions}}$

      `if Feasible(`λ`) then`

        $\lambda_{UB} := \min\{\lambda, \lambda_{UB}\}$

      `else`

        $\lambda_{LB} := \max\{\lambda, \lambda_{LB}\}$

```
      end if
    end if
    if λ_UB = ∞ then
      λ := λ_LB + 1
    else
      λ := ½λ_LB + ½λ_UB
    end if
    if Costs + λ·Emissions < MinimumCosts + λ·MinimumEmissions
     then
      MinimumCosts := Costs
      MinimumEmissions := Emissions
    end if
  end for
  m(t) := MinimumCosts
  m̂(t) := MinimumEmissions
end for
```

$$
\text{Here,} \qquad \begin{aligned} c(t,s) &:= p_t\left(D_{t,s}\right) + \sum_{\tau=t}^{s-1} h_\tau\left(D_{\tau,s}\right) &(18)\\ e(t,s) &:= \hat{p}_t\left(D_{t,s}\right) + \sum_{\tau=t}^{s-1} \hat{h}_\tau\left(D_{\tau,s}\right) \quad , &(19) \end{aligned}
$$

where $D_{t,s}$ is defined as $\sum_{\tau=t}^{s} d_\tau$.

The function $\texttt{Feasible}(\lambda)$ checks if the problem is feasible for the given λ by executing the Wagner-Whitin algorithm and checking whether the emission constraint is violated or not for the obtained solution. Equations (18) and (19) give the costs, respectively emissions, of procuring all of periods t through s's demand from period t.

After executing the algorithm, we get an interval $[\lambda_{LB}, \lambda_{UB}]$ that contains λ^*. Moreover, it is known that the same solution, say $x^{\frac{1}{2}}$, would be obtained for any $\lambda \in (\lambda_{LB}, \lambda_{UB})$. Hence, there are three solutions to consider: x^{UB}, $x^{\frac{1}{2}}$ and x^{LB}, corresponding to λ_{UB}, $(\frac{1}{2}\lambda_{LB} + \frac{1}{2}\lambda_{UB})$ and λ_{LB}, respectively. Note that these solutions may coincide. By construction of the algorithm, x^{UB} must be a feasible solution (if one exists) (see pseudocode). If $x^{\frac{1}{2}}$ is also feasible, we take the best feasible solution.

Furthermore, suppose that x^* is an optimal solution of problem (13)–(16) for some value of λ. Then we can compute $\sum_{t=1}^{T}\left(p_t(x_t^*) + h_t(I_t^*)\right) + \lambda^* \sum_{t=1}^{T}\left(\hat{p}_t(x_t^*) + \hat{h}_t(I_t^*) - \hat{C}\right)$, which is a lower bound for our problem. Observe that both x_{LB} and x_{UB} are optimal solutions, for λ_{LB} and λ_{UB}, respectively. Hence, we can compute that above expression for both solutions and take the higher lower bound.

5.2 Pseudo-polynomial algorithm for co-behaving costs and emissions

Apart from the heuristic, we also give a dynamic programming algorithm that solves ELSEC to optimality in case the costs and emissions satisfy the conditions in Theorem 3. We shall see that this algorithm works in pseudo-polynomial time. We construct this algorithm in such a way that it will be easy to turn it into an FPTAS in the next section.

First, assume that demand and all cost functions are integer, i.e., $d_t \in \mathbb{N}$ and $p_t(x_t), h_t(I_t) \in \mathbb{N}$ for $x_t, I_t \in \mathbb{N}$. Note that this does not have to hold for the emission functions, $\hat{p}_t$ and $\hat{h}_t$.

The general idea of the algorithm is as follows: we minimise the emissions under a (financial) budget constraint. Because of Theorem 3, we know that the single-sourcing property holds and we can extend Wagner and Whitin's well-known algorithm for the classic lot-sizing problem (Wagner and Whitin, 1958) with an extra state variable €, which denotes the budget. More precisely, let $f(t,€)$ denote the minimum emissions for periods t until T, given budget €. We define the following recursion:

$$
\begin{aligned}
f(t,€) &= \min_{s>t:\, € \geq c(t,s)} \{e(t,s) + f(s+1, € - c(t,s))\} \quad \text{for } t \leq T && (20)\\
f(T+1,€) &= 0 \quad , && (21)
\end{aligned}
$$

where, $c(t,s)$ and $e(t,s)$ are defined as in (18) and (19), respectively. Now, $f(1,€)$ gives the minimum emissions given budget €. We first compute $f(1,€)$ for $€ = 1$. If $f(1,1) \leq \hat{C}$, i.e., the minimum emissions are less than or equal to the emission cap, then we conclude that $€ = 1$ is the optimal value. If not, then the budget is raised to 2, we compute the corresponding minimum emissions $f(1,2)$ and again compare this to the emission cap. In this way, we try budgets $€ = 1, 2, 3, ...$ and compute the corresponding $f(1,€)$ until $f(1,€) \leq \hat{C}$, i.e., the minimum emissions are less than or equal to the emission cap. The first budget € for which this holds, is the optimal value.

For each $f(t,€)$, the optimal s is stored. The production schedule corresponding to the solution found by the algorithm can then be found through a simple backtracking procedure.

Running time

It is easy to see that the running time of this dynamic programme is $\mathcal{O}\left(T^2 opt\right)$, where opt is the optimal value (of the financial budget).

Memory

This algorithm needs $\mathcal{O}(Topt)$ memory, to store all values $f(t,€)$ and the corresponding optimal s.

Finding the Pareto frontier

In the process of finding the optimal solution, we construct part of the set of Pareto efficient solutions. This is because for each budget $€ = 1,\ldots,opt$, we find the minimum emissions, $f(1,€)$. This algorithm can be used to find the whole Pareto frontier. We first minimise emissions regardless of costs. This can be done by executing the (classic) Wagner-Whitin algorithm with the emission level as the objective (instead of the financial costs). Denote the corresponding costs by $\widetilde{€}$; it is easy to see that this is polynomial in the input of a problem instance. Now, we can compute the minimum emissions, $f(1,€)$ for each budget $€ = 1,\ldots,\widetilde{€}$. This procedure gives the whole Pareto frontier for co-behaving costs and emissions in $\mathcal{O}\left(T^2\widetilde{€}\right)$ time.

5.3 FPTAS for co-behaving costs and emissions

Clearly, it is the large number of budgets € to consider that makes the algorithm in the previous section run in *pseudo* rather than *fully* polynomial time. However, it is possible to turn the pseudo-polynomial algorithm into an FPTAS by reducing the number of states of € in a smart way. Instead of all budgets $€ = 1,2,\ldots$, we now only consider budgets equal to

$$\Delta^k := \left(1+\frac{\varepsilon}{(e-1)(T+1)}\right)^k \quad , \quad k \in \mathbb{N} \quad . \tag{22}$$

(See Figure 3.) This means that in every step of the dynamic programming recursion, we have to round down the budget to the nearest value of Δ^k.

$$f(t,€) = \min_{s>t:\ €\geq c(t,s)} \{e(t,s) + f(s+1,\text{round}(€ - c(t,s)))\} \quad \text{for } t \leq T \tag{23}$$

$$f(T+1,€) = 0 \tag{24}$$

$$\text{where} \qquad \text{round}(a) := \max_{k\in\mathbb{N}}\{\Delta^k : \Delta^k \leq a\} \tag{25}$$

Analogously to what we did before, we try budget $€ = \Delta^1,\Delta^2,\Delta^3,\ldots$ until $f(1,€) \leq \hat{C}$, i.e., the minimum emissions are less than or equal to the emission cap. Again, for each $f(t,€)$, the optimal s is stored. The production schedule corresponding to the solution found by the algorithm can then be found through a simple backtracking procedure.

The approach in which an exact, but only *pseudo*-polynomial dynamic programme is transformed into a FPTAS by trimming the state space is attributable to Woeginger (2000) and Schuurman and Woeginger (2011) (see also Ibarra and Kim, 1975), as well as the idea to use a so-called trimming parameter Δ of the type $\Delta := 1 + \frac{\varepsilon}{2gT}$. The FPTAS presented in this section takes an approach that is similar to Woeginger (2000). As far as we know, the FPTAS that is presented in Section 5.4 does not fit within his framework, because it is not based on a pseudo-polynomial algorithm, but rather on a generalisation of another FPTAS.

Correctness of the approximation

We verify that the obtained solution is in fact a $(1+\varepsilon)$ approximation of the true optimum. The question is: how much of the budget is 'wasted' by repeatedly rounding off the budget?

In each production period, at most the size of one interval $[\Delta^i, \Delta^{i+1})$ is lost. In the worst case this is the largest interval. Since there are at most T production periods, the maximum rounding error equals the size of the T largest intervals. Suppose that for some budget $€ = \Delta^{k+T}$, the algorithm gives no feasible solution (i.e., $f(1, \Delta^{k+T}) > \hat{C}$). Then we know that Δ^k is a lower bound, because we could have lost at most T intervals. Now, suppose that for the next budget, the algorithm does find a feasible solution (i.e., $f(1, \Delta^{k+T+1}) \leq \hat{C}$). So because we raise the budget from Δ^{k+T} to Δ^{k+T+1} each time we compute $f(1, €)$, we may lose one more interval. Hence, the maximum total error equals the size of the $T+1$ largest intervals. That means that if the algorithm finds a solution Δ^{k+T+1}, the optimal value is at least Δ^k. We therefore need to show that

$$\left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{k+T+1} \leq \left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{k} (1+\varepsilon) \quad .$$

This holds, because

$$\left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{k+T+1} = \left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{k} \left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{T+1} \quad ,$$

so we need to show that $\left(1 + \frac{\varepsilon}{(e-1)(T+1)}\right)^{T+1} \leq (1+\varepsilon)$. This is true because

$$\left(1 + \frac{\varepsilon/(e-1)}{T+1}\right)^{T+1} \leq 1 + (e-1) \cdot \frac{\varepsilon}{e-1} = 1 + \varepsilon \quad (\text{if } 0 < \varepsilon \leq (e-1)) \quad .$$

The inequality follows from the fact that $\left(1 + \frac{z}{n}\right)^n \leq 1 + (e-1)z$, if $0 \leq z \leq 1$.

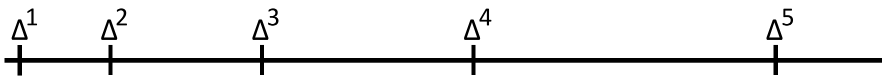

Figure 3: Budgets $\Delta^1, \Delta^2, \ldots$

Running time

The pseudo-polynomial algorithm in Section 5.2 has a running time of $\mathcal{O}(T^2 opt)$. Instead of opt intervals, the algorithm in this section has at most this many intervals:

$$\left\lceil^{1+\frac{\varepsilon}{(e-1)(T+1)}} \log(opt) \right\rceil = \left\lceil \frac{\ln(opt)}{\ln\left(1+\frac{\varepsilon}{(e-1)(T+1)}\right)} \right\rceil \leq \left\lceil \left(1+\frac{(e-1)(T+1)}{\varepsilon}\right) \ln(opt) \right\rceil \quad ,$$

so there are $\mathcal{O}\left(\frac{T \max\{\ln(opt),1\}}{\varepsilon}\right)$ budgets € to consider. Hence, the total running time is $\mathcal{O}\left(\frac{T^3 \max\{\ln(opt),1\}}{\varepsilon}\right)$, which is fully polynomial.

Memory

This algorithm needs $\mathcal{O}\left(\frac{T^2 \max\{\ln(opt),1\}}{\varepsilon}\right)$ memory, to store all values $f(t, €)$ and the corresponding optimal s.

A posteriori gap

As we have shown that the algorithm described in this section is a $(1+\varepsilon)$ approximation, we know that the optimality gap of the obtained solution is at most $100\varepsilon\%$. Previously, we have seen that Δ^k is a lower bound for the optimal value, if Δ^{k+T+1} is the (final) budget € corresponding to the algorithm's solution. Afterwards, we can compute the actual costs of this solution, which we will call v_{FPTAS}. We know that $v_{FPTAS} \leq \Delta^{k+T+1}$. That means that we can compute a smaller optimality gap as $\frac{v_{FPTAS}-\Delta^k}{\Delta^k}$.

An even better a posteriori gap can be obtained if we round down as much as possible during the execution of the algorithm. We then round down the budget according to the following rounding function:

$$\text{roundmore}\left(\Delta^i - c(t,s), t, s\right) := \max_{k \in \mathbb{N}} \left\{\Delta^k : \Delta^k \leq \Delta^{i-s+t} - c(t,s)\right\} \quad . \tag{26}$$

So we lose not just (at most) one interval in each block, but (at most) a number of intervals equal to the length of the block. It follows that the total number of intervals that we lose by rounding equals the total number of periods (T), as before.

5.4 FPTAS for general costs and emissions

As the FPTAS in the previous section is based on the single-sourcing property, it cannot be applied to the problem with general costs and emissions in a straightforward manner. However, Theorem 6 tells us that there is at most one period with double-sourcing. This leads to the following idea for a general FPTAS.

All blocks are 'normal' single-sourcing blocks, except for one *double-sourcing block*, say (t,s). The costs and emissions in the double-sourcing block depend on which period between t and s, say v, is the double-sourcing period. This implies that t and v are the two production periods in this block. The costs and emissions also depend on how much of the demand in periods v until s is produced in period t and how much in v. Note that the demand for $t,\ldots,v-1$, the earlier periods in this block, always has to be produced in period t. The costs to satisfy all demand in double-sourcing block (t,s) are between, say, a_{ts} and b_{ts}. These costs a_{ts} and b_{ts} can be computed by considering all double-sourcing periods v and calculating the costs corresponding to the situation where there is a set-up (if applicable) in both period t and v, but all demand in periods v until s is produced in either period t or period v. Now, we iterate over a 'suitable subset' of all values between a_{ts} and b_{ts}. These are the 'double-sourcing block budgets', \$. For each \$, we can compute the corresponding best v and (minimum) emissions in the double-sourcing block. For all other blocks, the single-sourcing property holds, so we can use a recursion like in the previous section.

The precise recursion is defined as follows:

$$
\begin{aligned}
g(t,€) &= \min\Bigg\{ \min_{s\geq t\,:\,€\geq c(t,s)} \{e(t,s)+g(s+1,\mathrm{round}(€-c(t,s)))\}, \\
&\qquad \min_{s\geq t\,,\,\$\in B_{ts}\,:\,€\geq\$} \{e(t,s,\$)+f(s+1,\mathrm{round}(€-\$))\}\Bigg\} &(27)\\
g(T+1,€) &= 0 &(28)\\
e(t,s,\$) &= \min_{v=t+1,\ldots,s}\{e(t,v,s,\$)\} &(29)\\
e(t,v,s,\$) &= \hat{p}_t\left(D_{t,v-1}+\alpha_{tvs\$}D_{v,s}\right)+\hat{p}_v\left((1-\alpha_{tvs\$})D_{v,s}\right)\\
&\quad +\sum_{\tau=t}^{v-1}\hat{h}_\tau\left(D_{\tau,v-1}+\alpha_{tvs\$}D_{v,s}\right)+\sum_{\tau=v}^{s}\hat{h}_\tau\left(D_{\tau,s}\right) &(30)
\end{aligned}
$$

$f(t,€)$, $c(t,s)$, $e(t,s)$ and $\mathrm{round}(\bullet)$ are exactly the same as in equations (23), (18), (19) and (25), respectively.

The interpretation of recursion (27) is: $g(t,€)$ gives the minimum emissions in periods t until T, given that there is a budget € and that there may be double-sourcing (once) in periods t until T. To find the value of $g(t,€)$, we need to determine whether the current block should have double-sourcing or not. The first line of (27) corresponds

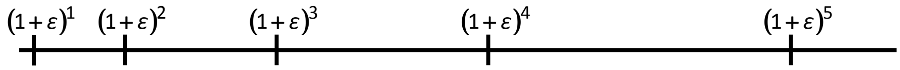

Figure 4: Budgets $(1+\varepsilon)^1, (1+\varepsilon)^2, \ldots$ for \$

to the situation in which there is no double-sourcing in the current block $[t,s]$. In that case, there may be double-sourcing in a later block and we should minimise over all possible values of the next production period, in a recursion that is similar to the $f(t,€)$ recursion (see Section 5.3). If there is double-sourcing in the current block, as in the second line of (27), then we need to minimise over s and \$, where s is the end of the current block and \$ is the amount of money that is spent in double-sourcing block (t,s). Since there cannot be another block with double-sourcing, the recursion uses the value $f(s+1,€)$ (see Section 5.3) as the minimum emissions of periods $s+1,\ldots,T$.

The minimum emissions given a budget € are given by $g(1,€)$. Try budget $€ = \Delta^1, \Delta^2, \Delta^3, \ldots$ until $g(1,€) \leq \hat{C}$, i.e., the minimum emissions are less than or equal to the emission cap, where Δ is defined as in equation (22).

The suitable subset of double-sourcing block budgets B_{ts} is defined as

$$\begin{aligned} B_{ts} &= \left\{\$: \$ = (1+\varepsilon)^k, k \in \mathbb{N}, a_{ts} \leq (1+\varepsilon)^k \leq b_{ts}\right\} \quad, && (31)\\ \text{where} \quad a_{ts} &= \min_{v=t,\ldots,s} \{c(t,v-1)+c(v,s)\} && (32)\\ \text{and} \quad b_{ts} &= \max_{v=t,\ldots,s} \{c(t,v-1)+c(v,s)\} && (33)\end{aligned}$$

That is, the double-sourcing block budget \$ is equal to $(1+\varepsilon)^k$ for some integer k and has to lie between the minimum and maximum costs in the double-sourcing block. See Figure 4.

In equation (29), $e(t,s,\$)$ gives the minimum emissions in double-sourcing block (t,s), given a budget \$. It is computed by minimising over the all possible double-sourcing periods v.

In equation (30), $e(t,v,s,\$)$ gives the emissions in double-sourcing block (t,v,s) (so given the double-sourcing period v), if a budget of \$ is spent. If the production and holding emissions are fixed-plus-linear, then this equation reduces to

$$e(t,v,s,\$) = \alpha_{tvs\$}\hat{a}_{tvs} + (1-\alpha_{tvs\$})\hat{b}_{tvs} \quad, \tag{34}$$

where $\hat{a}_{tvs}$ and $\hat{b}_{tvs}$ are the emissions to satisfy demand in the double-sourcing block, when there is a set-up (if applicable) in both period t and v, but all demand in periods v

through s is produced in period t, respectively v. $\alpha_{tvs\$}$ gives the fraction of demand in periods v through s that is produced in period t, if the budget in double-sourcing block (t, v, s) is \$; the remaining $(1 - \alpha_{tvs\$})$ is then produced in period v. If the production and holding emissions are fixed-plus-linear, then this is simply

$$\alpha_{tvs\$} = \frac{\$ - b_{tvs}}{a_{tvs} - b_{tvs}} \quad ,$$

where a_{tvs} and b_{tvs} are the costs to satisfy demand in the double-sourcing block, when there is a set-up (if applicable) in both period t and v, but all demand in periods v through s is produced in period t, respectively v. In general, $\alpha_{tvs\$}$ is the solution of

$$p_t\left(D_{t,v-1} + \alpha_{tvs\$}D_{v,s}\right) + p_v\left((1 - \alpha_{tvs\$})D_{v,s}\right) + \sum_{\tau=t}^{v-1} h_\tau\left(D_{\tau,v-1} + \alpha_{tvs\$}D_{v,s}\right) + \sum_{\tau=v}^{s} h_\tau\left(D_{\tau,s}\right) = \$. \tag{35}$$

We assume that this $\alpha_{tvs\$}$ can be found in constant time. This is the case for e.g. fixed-plus-linear costs, cost functions that are polynomials of degree at most four, and compound functions of which every piece is such a function (as long as the resulting function is concave for relevant production/inventory quantities). Otherwise, if finding an $\alpha_{tvs\$}$ takes $\mathcal{O}(A)$ time and this is more than $\mathcal{O}\left(\frac{\max\{\ln(opt),1\}}{\varepsilon}\right)$, then the time complexity becomes $\mathcal{O}\left(\frac{T^3 \max\{\ln^2(opt),1\}}{\varepsilon^2} + \frac{T^3 \max\{\ln(opt),1\}}{\varepsilon} \cdot A\right)$ (see Section 'Running time'). Note that we may approximate $\alpha_{tvs\$}$, for instance with a numerical method like bisection. However, in order for the algorithm to be accurate enough, we may not overestimate $\alpha_{tvs\$}$. (Here we assume that the lhs in (35) is an increasing function in $\alpha_{tvs\$}$. Otherwise, define $\alpha_{tvs\$}^{new} = 1 - \alpha_{tvs\$}$.)

In practice, the algorithm can be sped up, because we know that many triples (t, v, s) do not have to form a double-sourcing block in an optimal solution. This is because Theorem 3 tells us that the single-sourcing property holds for a triple (t, v, s), if it is true that $\left(C_{t,s} \leq C_{v,s} \text{ and } \hat{C}_{t,s} \leq \hat{C}_{v,s}\right)$ or $\left(C_{t,s} \geq C_{v,s} \text{ and } \hat{C}_{t,s} \geq \hat{C}_{v,s}\right)$. Therefore, it is not necessary to compute the minimum in (29) for the triples for which this holds.

Smart backtracking

The production schedule corresponding to the solution found by the algorithm can be found through a relatively simple backtracking procedure. For each $f(t, €)$, we store the optimal s, as before. For each $g(t, €)$, we store the optimal s, whether double-sourcing in block $[t, s]$ is optimal or not, and if so, which budget \$ is optimal. We could also store the optimal double-sourcing period v, but in certain cases, we can choose an approach to make a solution with lower costs by using as much of the (remaining) emission capacity as possible.

Suppose that the backtracking procedure has given the optimal production quantities in all blocks except the double-sourcing block, (t, v, s). We know that if there is double-sourcing in a period, then it is always best to use the whole emission capacity $\hat{C}$. (See Theorem 7.) However, because we have rounded the budget \$, it is very well possible that the FPTAS gives a solution in which the emissions are strictly smaller than the capacity. Therefore, we first compute the total emissions in all single-sourcing blocks. Then, we re-optimise the double-sourcing period $v = t+1, \ldots, s$ and budget \$, such that as much as possible of the remaining emission capacity is used. (This takes only $\mathcal{O}(T)$ time.)

Correctness of the approximation

As in Section 5.3, we verify that the obtained solution is in fact a $(1+\varepsilon)$ approximation of the true optimum by answering the question: how much of the budget is 'wasted' by repeatedly rounding off the budget?

Rounding values of \$ costs at most one 'big' $(1+\varepsilon)$-interval. In the remainder of the algorithm, at most $T+1$ 'small' Δ-intervals are lost. In Section 5.3, we have shown that these small intervals add up to at most one 'big' $(1+\varepsilon)$-interval. Hence, the maximum total error is $\varepsilon \cdot opt + \varepsilon(1+\varepsilon)opt = (2\varepsilon + \varepsilon^2)opt \leq 3\varepsilon \cdot opt$ (for $0 \leq \varepsilon \leq 1$). We could define $\varepsilon := \frac{\delta}{3}$ to get a $(1+\delta)$ approximation. In practice, we choose $\varepsilon = \sqrt{1+\delta} - 1 \geq \frac{\delta}{3}$. That way, ε is the positive solution of $2\varepsilon + \varepsilon^2 = \delta$.

Running time

As in the FPTAS for co-behaving costs, there are $\mathcal{O}\left(\frac{T \max\{\ln(opt), 1\}}{\varepsilon}\right)$ values for €. Similarly, we can show that there are $\mathcal{O}\left(\frac{\max\{\ln(opt), 1\}}{\varepsilon}\right)$ intervals for \$, because the number of double-sourcing block budgets \$ is at most

$$\left\lceil {}^{1+\varepsilon}\log(opt) \right\rceil = \left\lceil \frac{\ln(opt)}{\ln(1+\varepsilon)} \right\rceil \leq \left\lceil \left(1 + \frac{1}{\varepsilon}\right) \ln(opt) \right\rceil \quad .$$

In total, there are $\mathcal{O}\left(\frac{T^2 \max\{\ln(opt), 1\}}{\varepsilon}\right)$ values of both $g(t, €)$ and $f(t, €)$ that need to be computed. As in Section 5.3, it takes $\mathcal{O}(T)$ time to compute one $f(t, €)$. Computing one $g(t, €)$ takes $\mathcal{O}\left(T + T \cdot \frac{\max\{\ln(opt), 1\}}{\varepsilon}\right) = \mathcal{O}\left(\frac{T \max\{\ln(opt), 1\}}{\varepsilon}\right)$ time, because there are two minimisations in recursion (27); the first one over periods s; the second one over periods s and $\$ \in B_{ts}$. Hence, the total time needed to compute all $g(t, €)$ and $f(t, €)$ is $\mathcal{O}\left(\frac{T \max\{\ln(opt), 1\}}{\varepsilon} + \frac{T^3 \max\{\ln^2(opt), 1\}}{\varepsilon^2}\right) = \mathcal{O}\left(\frac{T^3 \max\{\ln^2(opt), 1\}}{\varepsilon^2}\right)$.

Furthermore, there are $\mathcal{O}\left(\frac{T^2 \max\{\ln(opt), 1\}}{\varepsilon}\right)$ values of $e(t, s, \$)$ that need to be computed. Computing one $e(t, s, \$)$ takes $\mathcal{O}(T)$ time, so the time needed to compute all

$e(t,s,\$)$ is $\mathcal{O}\left(\frac{T^3 \max\{\ln(opt),1\}}{\varepsilon}\right)$. Since all $e(t,s,\$)$ can be computed beforehand, it follows that the time complexity of the whole FPTAS is $\mathcal{O}\left(\frac{T^3 \max\{\ln^2(opt),1\}}{\varepsilon^2}\right)$.

Memory

As in the co-behaving case, this algorithm needs $\mathcal{O}\left(\frac{T^2 \max\{\ln(opt),1\}}{\varepsilon}\right)$ memory to store all values $f(t,€)$ and the corresponding optimal s, and all values $g(t,€)$ and the corresponding optimal s and \$. Storing all values $e(t,s,\$)$ requires $\mathcal{O}\left(\frac{T^3 \max\{\ln(opt),1\}}{\varepsilon}\right)$ memory. Hence, the total required memory is of the same order.

A posteriori gap

As we have shown that the algorithm described in this section is a $(1+\varepsilon)$ approximation, we know that the optimality gap of the obtained solution is at most $100\varepsilon\%$. Previously, we have seen that $\frac{\Delta^{k+T+1}}{(2\varepsilon+\varepsilon^2)}$ (or even: $\frac{\Delta^{k+T+1}}{(1+\varepsilon)\Delta^{T+1}} = \frac{\Delta^k}{1+\varepsilon}$) is a lower bound for the optimal value, if Δ^{k+T+1} is the (final) budget € corresponding to the algorithm's solution. Afterwards, we can compute the actual costs of this solution, which we will call v_{FPTAS}. We know that $v_{FPTAS} \leq \Delta^{k+T+1}$. That means that we can compute the optimality gap more sharply as $\frac{v_{FPTAS} - \frac{\Delta^k}{1+\varepsilon}}{\frac{\Delta^k}{1+\varepsilon}}$.

As in Section 5.3, an even better a posteriori gap can be obtained if we round down € as much as possible during the execution of the algorithm. We round down the budget € according to the roundmore function (see equation (26)). As before, it follows that the total number of Δ-intervals that we lose by rounding € equals the total number of periods (T).

What if 1 is not a trivial LB?

For the FPTAS for co-behaving costs and emissions, it was trivial that 1 was a lower bound, because demand and cost functions were assumed integer, and production was always integral, in accordance with Theorem 3. For the general FPTAS described in this section, this is no longer trivial, as production in the double-sourcing block may be non-integral. However, the instances with an optimal value lower than 1 all correspond to a very specific situation, which we can easily exclude.

In these instances, costs must equal 0 in all single-sourcing blocks and one of the sources in the double-sourcing block. Now, iterate over all possible double-sourcing intervals (at most $\frac{1}{2}T(T-1)$), such that all other costs equal 0.

Given a double-sourcing block $[t,s]$, we solve two classic lot-sizing problems: we minimise emissions in $[1,t-1]$ and in $[s+1,T]$ with an algorithm such as Wagelmans et al. (1992) or Wagner and Whitin (1958), extended with the following tie-breaking rule. See the algorithm as a decision tree. If somewhere in the tree we must choose between branches with equal emissions, then choose the branch with lower costs.

Consider all double-sourcing blocks $[t, s]$ such that the emissions in $[1, t-1] \cup [s+1, T]$ are below the capacity and the costs are zero, if any of such intervals exist. Iterate over all possible second sources v in this interval ($t < v \leq s$), such that one of the sources (t or v) has costs zero. Compute the emission capacity that remains for such a double-sourcing block (t, v, s), if any of such blocks exist. Now, we know how much should be produced in each source such that the emissions are within capacity, if this is possible at all. Compute the costs in the double-sourcing blocks for which this is possible. If there exists such a double-sourcing block with costs lower than 1, then 1 is not a lower bound and the costs of the cheapest double-sourcing block is the optimal value. Otherwise, 1 is a lower bound.

We can check this in $\mathcal{O}\left(T^3\right)$.

5.5 Using the heuristic to speed up the FPTAS

In the execution of the FPTASes in Sections 5.3 and 5.4, we encounter many small intervals. For example, we need to compute $f(t, €)$ for $€ = \Delta^1, \Delta^2, \Delta^3, \ldots$, even though the optimal value is closer to, say, Δ^{100}. In retrospect, we would not have needed intervals smaller than $\frac{\varepsilon}{(e-1)(T+1)} opt$ for €. Of course, we do not know the optimal value beforehand. However, we can compute a lower bound (LB) first, so that we know that we do not need intervals smaller than $\frac{\varepsilon}{(e-1)(T+1)} LB$ for € during the execution of the FPTAS. We replace all intervals below LB by intervals of size $\frac{\varepsilon}{(e-1)(T+1)} LB$. To see why this works, we look back at the Correctness of the approximation in Section 5.3. Again, suppose we find a solution when $€ = \Delta^{k+T+1}$ ($\geq LB$). Also, suppose we have a lower bound after executing the algorithm, say LB_{post}. In Section 5.3, this lower bound equaled Δ^k; now, it is $LB_{\text{post}} = \max\{\Delta^k, LB\}$. If $LB \geq \Delta^k$, then it follows that we have found a $(1+\varepsilon)$ approximation, because $opt - LB \leq opt - \Delta^k \leq \Delta^{k+T+1} - \Delta^k \leq \Delta^k(\Delta^{T+1} - 1) \leq \Delta^k(1 + \varepsilon - 1) \leq LB \cdot \varepsilon \leq opt \cdot \varepsilon$, where the correctness of the fourth inequality was shown in Section 5.3. Alternatively, suppose that $\Delta^k > LB$. In the worst case, we have lost the $T+1$ intervals due to rounding. In the proof in Section 5.3, we have shown that losing the $T+1$ biggest intervals still resulted in a $(1+\varepsilon)$ approximation. There, the smallest of the biggest intervals had size $\Delta^{k+1} - \Delta^k = \Delta^k(\Delta - 1) = \Delta^k \cdot \frac{\varepsilon}{(e-1)(T+1)}$. In the algorithm in this section, the intervals above LB are the same as before; the intervals below LB have size $\frac{\varepsilon}{(e-1)(T+1)} LB \leq \frac{\varepsilon}{(e-1)(T+1)} \Delta^k$. Because the $T+1$ biggest intervals that can be lost in this

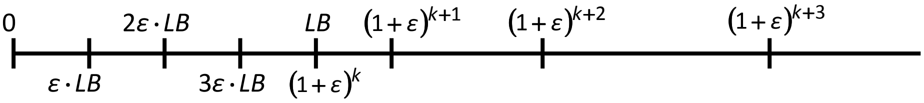

Figure 5: Intervals for \$ of size at least $\varepsilon \cdot LB$

section have the same size as or are smaller than in Section 5.3, we conclude that we still have a $(1+\varepsilon)$ approximation.

Similarly, we may use intervals of size at least $\varepsilon \cdot LB$ for \$ in the FPTAS for general costs and emissions. We replace all intervals below LB by intervals of size $\varepsilon \cdot LB$. See Figure 5 for an example with $LB = 4\varepsilon = (1+\varepsilon)^k$.

In the computational tests in the next section, we will use the Lagrangian heuristic from Section 5.1 to compute a lower bound, but of course any method to compute a nonzero lower bound would do.

Note that, because we use a lower bound in the FPTASes, we do not need integer demand and cost functions anymore.

Running time

To determine the running times of both FPTASes if we use the minimum interval size as described above, we must compute the new numbers of values for € and \$.

For the total budget €, we compute the number of values that we had in the FPTAS before, subtract the number of values that lay below LB (as these values will not be used anymore), and add the number of newly created, larger intervals that lie below LB. We get:

$$\begin{aligned}
& \left\lceil {}^{1+\frac{\varepsilon}{(e-1)(T+1)}}\log(opt) \right\rceil - \left\lfloor {}^{1+\frac{\varepsilon}{(e-1)(T+1)}}\log(LB) \right\rfloor + \left\lceil \frac{LB}{\frac{\varepsilon}{(e-1)(T+1)} LB} \right\rceil \\
\leq \;& {}^{1+\frac{\varepsilon}{(e-1)(T+1)}}\log(opt) - {}^{1+\frac{\varepsilon}{(e-1)(T+1)}}\log(LB) + \frac{(e-1)(T+1)}{\varepsilon} + 3 \\
= \;& {}^{1+\frac{\varepsilon}{(e-1)(T+1)}}\log\left(\frac{opt}{LB}\right) + \frac{(e-1)(T+1)}{\varepsilon} + 3 \quad ,
\end{aligned}$$

so there are $\mathcal{O}\left(\frac{T \max\{\ln\left(\frac{opt}{LB}\right), 1\}}{\varepsilon} + \frac{T}{\varepsilon}\right) = \mathcal{O}\left(\frac{T \max\{\ln\left(\frac{opt}{LB}\right), 1\}}{\varepsilon}\right)$ values for €, using the same argument as in Section 5.3.

For the double-sourcing block budget \$, the analysis is similar. We get:

$$\left\lceil {}^{1+\varepsilon}\log(opt) \right\rceil - \left\lfloor {}^{1+\varepsilon}\log(LB) \right\rfloor + \left\lceil \frac{LB}{\varepsilon \cdot LB} \right\rceil$$

$$
\begin{aligned}
&\leq \; ^{1+\varepsilon}\log(opt) - ^{1+\varepsilon}\log(LB) + \frac{1}{\varepsilon} + 3 \\
&= \; ^{1+\varepsilon}\log\left(\frac{opt}{LB}\right) + \frac{1}{\varepsilon} + 3 \quad ,
\end{aligned}
$$

so there are $\mathcal{O}\left(\frac{\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon} + \frac{1}{\varepsilon}\right) = \mathcal{O}\left(\frac{\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon}\right)$ values for \$, using the same argument as in Section 5.4.

This gives the following running times:

- $\mathcal{O}\left(\frac{T^3\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon}\right)$ for the FPTAS for co-behaving costs and emissions plus the running time of the algorithm that provides the lower bound. The Lagrangian heuristic from Section 5.1 that we use, for instance, has a running time of $\mathcal{O}\left(T^4\right)$, giving a total running time of $\mathcal{O}\left(\frac{T^3\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon} + T^4\right)$. This can be reduced to $\mathcal{O}\left(\frac{T^3\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon}\right)$ for fixed-plus-linear costs and emissions if an $\mathcal{O}\left((T\ln T)^2\right)$ implementation of the heuristic is used, i.e., one that is based on an $\mathcal{O}(T\ln T)$ algorithm for the classic lot-sizing problem, such as Wagelmans et al. (1992).
- $\mathcal{O}\left(\frac{T^3\max\{\ln^2\left(\frac{opt}{LB}\right),1\}}{\varepsilon^2}\right)$ for the FPTAS for general costs; again plus the running time of the algorithm that provides the lower bound.

Memory

It follows that the FPTAS for co-behaving costs and emissions needs $\mathcal{O}\left(\frac{T^2\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon}\right)$ memory and the general FPTAS needs $\mathcal{O}\left(\frac{T^3\max\{\ln\left(\frac{opt}{LB}\right),1\}}{\varepsilon}\right)$ memory.

6 Computational tests

6.1 Test set-up

The FPTASes that we developed have some nice theoretical properties regarding their running times and approximation qualities. However, we are also interested in their practical performance. Moreover, we would like to know how well the Lagrangian heuristic performs on a large number of test instances. Therefore, we have randomly generated 1800 problem instances. These instances are solved with all of the algorithms that were presented in this paper. More specifically, these are:

- the Lagrangian heuristic ('Megiddo') from Section 5.1;
- the pseudo-polynomial algorithm for co-behaving costs and emissions (PP-CB) from section 5.2, if the instance satisfies the conditions for co-behaviour in Theorem 3;
- the FPTAS for co-behaving costs and emissions (FPTAS-CB) from section 5.3, again only if the instance is co-behaving indeed;
- the FPTAS for co-behaving costs and emissions that uses the lower bound generated by Megiddo (FPTAS-CB-LB), again only if the instance is co-behaving;
- the general FPTAS (FPTAS-gen);
- the general FPTAS that uses the Megiddo lower bound (FPTAS-gen-LB);
- for comparison purposes, we included the CPLEX 10.1 solver. We used this solver on the 'natural' formulation, as defined in equations (6)-(12), as well as on the shortest path reformulation. The shortest path reformulation, as introduced by Eppen and Martin (1987), is known to have a better LP relaxation.

For each of the FPTASes, three values of ε were used: 0.10, 0.05 and 0.01. The FPTASes that use Megiddo's lower bound (FPTAS-CB-LB and FPTAS-gen-LB) were executed even when the feasible solution found by Megiddo was within $(1+\varepsilon)$ from the lower bound. This was done in order to reduce the a posteriori gap, even though it was not strictly necessary.

The values of the problem parameters were chosen in the following way. Although the algorithms are suitable for more general concave functions, all cost and emissions functions were assumed to have a fixed-plus-linear structure. This is a common cost structure in the literature. Moreover, it allowed us to also solve the instances with CPLEX, so that we can compare our algorithms' solutions with the optimal solution.

The time horizons that we considered were 25, 50 and 100 periods. Horizons as long as 100 period were considered, because the number of time periods in our model (T) may correspond to $m \cdot T'$ for instances with m production modes and T' periods.

First, we generated instances that satisfy the co-behaviour conditions in Theorem 3. Demand was generated from a discrete uniform distribution with minimum 0 and maximum 200 (and thus mean 100). Both the set-up costs and emissions were drawn from three different discrete uniform distributions: DU(500, 1500), DU(2500, 7500) and DU(5000, 15000) (with means 1000, 5000 and 10000). p_t, $\hat{p}_t$, h_t and $\hat{h}_t$ were all generated from DU(0, 20), but we only kept those $(p, \hat{p}, h, \hat{h})$ that satisfy the conditions in Theorem 3.

The second group of instances was generated from the same distributions, with the same parameters, but we only kept those $(p, \hat{p}, h, \hat{h})$ such that exactly $\lceil \frac{1}{2} T \rceil$ period pairs (t, s) are eligible for double-sourcing. That is, for $\lceil \frac{1}{2} T \rceil$ pairs the conditions in Theorem 3 were violated.

The third group of instances was different from the other data sets in the sense that periods always occurred in (consecutive) pairs, where the even periods have low production and set-up costs and high production and set-up emissions, and the odd periods have high costs and low emissions. To be precise, p_t was drawn from DU(0, 9) for t even and from DU(11, 20) for t odd; $\hat{p}_t$ was drawn from DU(11, 20) for t even and from DU(0, 9) for t odd. The low set-up costs and emissions, K_t and $\hat{K}_s$, for t even and s odd, were drawn from DU(500, 1500). The high set-up costs and emissions, for t odd and s even, were both drawn from DU(2500, 7500) and DU(5000, 10000). The holding costs and emissions between two periods within one pair were always zero. Between two pairs, they were drawn from DU(0, 20). Demand was zero in the first period of a pair, and in the second period generated from DU(0, 200). The numbers of periods we considered are 26, 50 and 100. Generating the data in this way corresponds to a problem with $\frac{1}{2} T$ periods, but with two production modes, 'cheap & dirty' and 'expensive & clean'. These instances show similarities with the instance that was used in the $\mathcal{NP}$-hardness proof (Theorem 1), so we expect that they are difficult to solve.

Ten instances were generated for every combination of the parameter settings that were described above, giving 600 data sets. Every instance thus generated was combined with three different values of the emission capacity. We let $\hat{C} = \lceil \beta \hat{C}_{\min} + (1 - \beta) \hat{C}_{\max} \rceil$, where $\beta = 0.25, 0.5, 0.75$, $\hat{C}_{\min}$ is the level of emissions when emissions are minimised, ignoring costs, and $\hat{C}_{\max}$ is the level of emissions when costs are minimised, ignoring emissions. In total, this gave $600 \cdot 3 = 1800$ instances.

All algorithms were implemented in a Java programme that was used to solve all instances on a Windows 7-based PC with an AMD Athlon II X2 B24 processor ($2 \times$ 3000 MHz) and 4 GB RAM.

6.2 Results

A summary of the results of the computational tests can be found in Table 1. Tables 2–8 in Appendix B.1 give more detailed results, for different values of the average set-up costs and emissions, or emission capacity. Four characteristics are given for each algorithm:

- the average solution time of the algorithm, where the computation time of Megiddo was included in the times of the FPTASes that used this lower bound;

- the average a posteriori gap, the percentage difference between the algorithm's solution and the lower bound that the algorithm found;
- the average true gap, the percentage difference between the algorithm's solution and the optimal value that was found by CPLEX (and PP-CB);
- the percentage of instances for which the algorithm's solution value was exactly equal to the optimal value.

Below, we will discuss the most important findings.

Tables 2, 3 and 4 give the results for the co-behaving instances, which satisfy the conditions in Theorem 3, as summarised in the columns marked 'co-bhv.' in Table 1. We see that the heuristic (Megiddo) finds solutions that are very close to the optimum. For a horizon of 25 periods, it even finds the optimum itself in over 60% of the cases, and the true gap is less than a half percent on average; its a posteriori gap is 1.5% on average. It is remarkable to see that if the horizon becomes longer (50 or 100 periods), these gaps become even smaller.

The set-up emissions ($\hat{K}$) and emissions capacity ($\hat{C}$) do not appear to have a big influence on the results, for any of our algorithms. For lower set-up costs (K), Megiddo's gaps are smaller.

Looking at the results for the FPTASes for co-behaving costs and emissions (FPTAS-CB) tells us that they give solutions that are well within the specified precision in a very short amount of time. The average computation times of FPTAS-CB-LB ranges from 0.39 seconds, for 100 periods and $\varepsilon = 0.01$, down to only 1 millisecond for 25 periods and $\varepsilon = 0.1$. FPTAS-CB-LB with $\varepsilon = 0.05$ or $\varepsilon = 0.1$ is faster than CPLEX, even on the shortest path formulation. For 25 and 50 periods, this also holds when ε is 0.01. Of course, this comes at the expense of ε-optimal solutions instead of the optimal solutions that were generated by CPLEX. Nonetheless, even when $\varepsilon = 0.1$, the optimum is found in over two-thirds of the instances, and the average true gaps are below 0.025%. For $\varepsilon = 0.01$, these are even below 0.0005%.

Comparing the FPTAS-CBs with the general FPTASes, we see that the general FPTASes have a higher computation time, as could be expected. However, the increase appears to be less than of order $\frac{T \ln(opt)}{\varepsilon}$, which is what would be expected from the difference in time complexities (see Sections 5.3 and 5.4). This is because our implementation of the FPTAS-gen checks whether double-sourcing 'makes sense', and, because these data sets satisfy the conditions in Theorem 3, this is never the case. The solutions of FPTAS-gen are even better than those of FPTAS-CB, because a smaller epsilon ($\varepsilon = \sqrt{1+\delta} - 1$) is used, which is unnecessary, because for co-behaving data, the solution never has double-sourcing.

	T	25	25	26	50			100		
	data set	co-bhv.	gen.	2 modes	co-bhv.	gen.	2 modes	co-bhv.	gen.	2 modes
Megiddo	avg. sol. time (s)	<0.001	<0.001	0.002	0.001	0.001	0.004	0.002	0.003	0.016
	avg. post. gap (%)	1.5	2.8	12	0.85	1.3	6.2	0.41	0.61	2.8
	avg. true gap (%)	0.47	1.2	6.1	0.41	0.74	3.8	0.26	0.41	2.1
	solved to opt. (%)	63	43	42	44	31	22	32	21	30
PP-CB	avg. sol. time (s)	0.24			1.8			22		
FPTAS-CB-LB(0.1)	avg. sol. time (s)	0.001			0.007			0.036		
	avg. post. gap (%)	0.81			0.44			0.17		
	avg. true gap (%)	0.021			0.024			0.015		
	solved to opt. (%)	89			79			69		
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.001			0.009			0.068		
	avg. post. gap (%)	0.55			0.34			0.16		
	avg. true gap (%)	0.0022			0.0067			0.0060		
	solved to opt. (%)	96			90			83		
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.006			0.048			0.39		
	avg. post. gap (%)	0.15			0.12			0.075		
	avg. true gap (%)	0.00044			0.00016			0.00014		
	solved to opt. (%)	98			98			98		
FPTAS-CB(0.1)	avg. sol. time (s)	0.008			0.052			0.35		
	avg. post. gap (%)	3.4			3.4			3.5		
	avg. true gap (%)	0.010			0.020			0.017		
	solved to opt. (%)	91			80			68		
FPTAS-CB(0.05)	avg. sol. time (s)	0.018			0.11			0.77		
	avg. post. gap (%)	1.7			1.7			1.7		
	avg. true gap (%)	0.0021			0.0054			0.0042		
	solved to opt. (%)	95			89			84		
FPTAS-CB(0.01)	avg. sol. time (s)	0.093			0.67			5.2		
	avg. post. gap (%)	0.33			0.34			0.34		
	avg. true gap (%)	0.000088			0.00015			0.00015		
	solved to opt. (%)	99			98			98		
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.003	0.005	0.017	0.013	0.029	0.11	0.083	0.20	0.71
	avg. post gap (%)	1.0	1.6	3.7	0.45	0.62	2.3	0.16	0.21	0.69
	avg. true gap (%)	0.0053	0.063	0.022	0.0066	0.024	0.031	0.0048	0.017	0.0080
	solved to opt. (%)	91	72	88	89	75	83	83	63	82
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.004	0.009	0.041	0.025	0.063	0.29	0.16	0.48	2.0
	avg. post gap (%)	0.92	1.4	2.3	0.44	0.61	1.9	0.16	0.21	0.69
	avg. true gap (%)	0.00082	0.041	0.028	0.0011	0.022	0.039	0.0014	0.014	0.0080
	solved to opt. (%)	97	78	97	94	76	90	91	67	88
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.016	0.082	0.57	0.13	0.69	5.5	1.1	5.7	36
	avg. post gap (%)	0.41	0.46	0.54	0.32	0.38	0.54	0.15	0.20	0.46
	avg. true gap (%)	0.000014	0.011	0.00066	0.000080	0.010	0.011	0.0000090	0.0076	0.0033
	solved to opt. (%)	100	87	97	99	82	90	99	71	88
FPTAS-gen(0.1)	avg. sol. time (s)	0.022	0.054	0.14	0.13	0.42	1.2	0.94	3.6	11
	avg. post gap (%)	6.6	6.6	6.3	6.6	6.6	6.3	6.6	6.6	6.4
	avg. true gap (%)	0.0042	0.017	0.014	0.0048	0.012	0.015	0.0046	0.0093	0.010
	solved to opt. (%)	94	81	90	89	78	85	83	65	80
FPTAS-gen(0.05)	avg. sol. time (s)	0.046	0.14	0.42	0.29	1.2	4.2	2.1	10	36
	avg. post gap (%)	3.3	3.3	3.2	3.3	3.3	3.2	3.3	3.3	3.2
	avg. true gap (%)	0.00048	0.0081	0.019	0.00084	0.0072	0.015	0.0017	0.0040	0.0038
	solved to opt. (%)	98	84	88	95	84	85	90	73	87
FPTAS-gen(0.01)	avg. sol. time (s)	0.27	2.1	8.7	1.9	20	90	14	165	691
	avg. post gap (%)	0.67	0.66	0.65	0.67	0.66	0.64	0.67	0.67	0.64
	avg. true gap (%)	0.000027	0.0014	0.00073	0.000064	0.00052	0.00099	0.000049	0.00089	0.00095
	solved to opt. (%)	99	95	95	99	94	95	98	83	92
CPLEX 10.1 Nat.	avg. sol. time (s)	0.045	0.041	0.035	0.44	0.38	0.12	–	–	–
CPLEX 10.1 SP	avg. sol. time (s)	0.030	0.031	0.053	0.069	0.076	0.14	0.23	0.27	0.55

Table 1: Summary of all results

The FPTASes that use the lower bound have a much lower computation time than the ones that do not, so using the lower bound really makes a difference. The reduction in computation time varies from about seven times faster than the (already fast) FPTAS-gen(0.1) for $T = 25$ (and FPTAS-CB(0.1) for $T = 50$), up to almost thirty times faster than FPTAS-gen(0.01) for $T = 50$ (0.69 vs. 20 seconds). The solutions of the FPTASes without lower bound have even smaller true gaps than those found by the FPTASes with lower bounds, since not using the lower bound results in using smaller intervals than necessary. The a posteriori gaps found by the FPTASes without lower bounds are larger than those found by the FPTASes with lower bounds, because the latter can compute the gap with respect to two lower bounds, Δ^{k-T-1} (see Section 5.3) and the heuristic's lower bound. Of course, the higher of the two is used. The a posteriori gaps of FPTAS-CB (without lower bound) are about two thirds less than is required by ε, and those of FPTAS-gen are about one third less (e.g., an a posteriori gap of 0.67% when $\varepsilon = 0.01$). Tables 1–8 all give the results that were obtained with the 'round-more' function (see pages 18 and 23). We can compare these with the a posteriori gaps that were obtained by the algorithms that do not use this improved lower bound, as can be found in Tables 9–15 in Appendix B.2. We see that in that case the a posteriori gaps of FPTAS-CB (without lower bound) are half of what is required by ε, and those of FPTAS-gen are only one quarter less than required by ε (e.g., an a posteriori gap of 0.75% when $\varepsilon = 0.01$).

The pseudo-polynomial algorithm (PP-CB) is still reasonably fast, but not as fast as the FPTAS-CBs. Moreover, its computation times increase as the set-up costs increase, since this means that the optimal value increases as well, and its time complexity is dependent on this optimal value (see Section 5.2).

CPLEX applied to the natural formulation is very sensitive to the size of the set-up costs. Only for the smallest set-up costs, it is sometimes slightly faster than the shortest path formulation. Moreover, for 100 periods, we were very often not able to solve the instances at all, because of memory issues. The results for CPLEX-nat are therefore not included for $T = 100$.

The results for the instances with $\left\lceil \frac{1}{2}T \right\rceil$ pairs that violate the co-behaviour property are shown in Tables 5, 6 and 7, and are summarised in the columns marked 'gen.' in Table 1. In general, we see the same patterns as for the co-behaving instances.

Megiddo still gives good solutions in the same amount of time, although the solutions are not as good as in the co-behaving case. This is because the heuristic can only find solutions that satisfy the single-sourcing property, whereas these non-co-behaving instances can have an optimal solution with a double-sourcing block (see Theorem 6). Still, the average true gap is 1.2% for 25 periods, down to less than a half percent for

100 periods.

The results for the FPTASes are similar to what we have seen before, but the computation times have increased compared to the co-behaving case, because now, we also need to iterate over the double-sourcing block budgets \$ (see Section 5.4) in the $\lceil \frac{1}{2}T \rceil$ period-pairs in which double-sourcing might be optimal. However, the solution times of FPTAS-gen-LB(0.1) are still shorter than CPLEX-SP. Moreover, the true gaps are still very close to zero for all FPTASes.

Table 8 gives the results for the instances that can be interpreted as having two production modes (cheap & dirty and expensive & clean), as summarised in the columns marked '2 modes' in Table 1. Roughly the same patterns as before are shown. However, the gaps of the heuristic, and the computation times of the FPTASes are again larger. Of course, this comes as no surprise, because we specially *designed* these problem instances to be the hardest to solve for our algorithms. The highest average solution time is obtained by FPTAS-gen with $\varepsilon = 0.01$: seven and a half minutes for $T = 100$. On the other hand, if the heuristic's lower bound is used in the FPTAS, the average computation times are below 36 seconds, even for $\varepsilon = 0.01$ and $T = 100$. If we take a higher epsilon ($\varepsilon = 0.1$), then the average solution time goes down to 0.71 seconds, while still obtaining solutions with an average true gap below 0.01%. Unfortunately, this is slightly slower than CPLEX-SP. However, for $T = 25$ or $T = 50$, FPTAS-gen-LB(0.01) is faster than CPLEX-SP. Moreover, where CPLEX requires the cost and emission functions to fit in a linear model, our algorithms are able to handle more general concave cost and emission functions.

7 Conclusions & further research

In this paper, we have considered a lot-sizing problem with a global emission constraint. Here, the emissions take the form of a second type of 'costs' on production, set-up and inventory decisions. Of course, these second costs can be any type of costs other than those in the objective function. We have shown that this problem is $\mathcal{NP}$-hard (in the weak sense) even if only production emits pollutants (linearly). From the $\mathcal{NP}$-hardness proof, we learned that our model also entails lot-sizing with emissions and multiple production modes. We have presented a Lagrangian heuristic (Megiddo), FPTASes and a pseudo-polynomial algorithm to solve the problem, and subjected these algorithms to a large number of computational tests. This has shown that Megiddo gives near-optimal solutions, and we recommend using its lower bound as input for the FPTASes. Moreover, we have seen that instances are easier to solve if the costs and emissions satisfy a co-behaviour property (see Theorem 3). This is also reflected by the time complexity of the FPTASes; for the co-behaving case, this is $\mathcal{O}\left(\frac{T^3 \max\{\ln(opt/LB),1\}}{\varepsilon}\right)$, whereas in the general case, it is $\mathcal{O}\left(\frac{T^3 \max\{\ln^2(opt/LB),1\}}{\varepsilon^2}\right)$. We have seen that, in practice, the FPTASes have a much smaller gap than the a priori imposed performance. The FPTASes that use Megiddo's lower bound (FPTAS-CB-LB and FPTAS-gen-LB) are very fast, even compared to CPLEX. In case the costs and emissions are co-behaving, they are even faster. We have seen that the instances that are the hardest to solve, are constructed in such a way that the degree of non-co-behaviour is very high. Instances with two production modes are the hardest in this regard. However, recall that our algorithms are able to solve instances with more general concave cost and emission functions.

Because we have carried out a large number of computational tests, special attention was paid to an efficient implementation of the FPTASes. We developed an improved rounding technique to reduce the a posteriori gap, and combined an FPTAS in the style of Woeginger (2000) with a lower bound, which turned out to lead to very good results. We expect that these techniques can be applied to more FPTASes of this type.

We think that it may be worthwhile to develop a Lagrangian heuristic for fixed-plus-linear costs and emissions, following Megiddo's approach, based on an $\mathcal{O}(T \ln T)$ algorithm for the classic lot-sizing problem, such as Wagelmans et al. (1992). Futhermore, we expect that the technique to construct a pseudo-polynomial algorithm and an FPTAS can be applied to more problems where one capacity constraint (on a 'second objective function') is added to a problem for which a polynomial time dynamic programme exists. In our opinion, another interesting line of future research into lot-sizing with emission constraints involves extending the lot-sizing model to a production-distribution system with emissions.

Acknowledgements

Mathijn J. Retel Helmrich's and Wilco van den Heuvel's research was supported by the Netherlands Organisation for Scientific Research (NWO).

References

N. Absi, S. Dauzère-Pérèz, S. Kedad-Sidhoum, B. Penz, and C. Rapine. Lot-sizing with carbon emission constraints. Technical report, 2011.

S. Benjaafar, Y. Li, and M. Daskin. Carbon footprint and the management of supply chains: Insights from simple models. Technical report, University of Minnesota, 2011.

Y. Bouchery, A. Ghaffari, Z. Jemai, and Y. Dallery. How to incorporate sustainability issues into traditional operations management models? The sustainable order quantity example. Technical report, Ecole Centrale Paris, Laboratoire Génie Industriel, 2010.

Carbon Disclosure Project. CDP global 500 report 2011, accelerating low carbon growth, 2011. `https://www.cdproject.net/CDPResults/CDP-G500-2011-Report.pdf`.

X. Chen, S. Benjaafar, and A. Elomri. The carbon-constrained EOQ. Technical report, University of Minnesota, 2011.

G. D. Eppen and R. K. Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6):832–848, 1987.

European Commission. EU Emissions Trading System (EU ETS), 2010. `http://ec.europa.eu/clima/policies/ets/index_en.htm`. Accessed on 6 October 2011.

D. Gusfield. Parametric combinatorial computing and a problem in a program module allocation. *Journal of the Association for Computing Machinery*, 30:551–563, 1983.

M. Heck and G. Schmidt. Lot-size planning with non-linear cost functions supporting environmental sustainability. *International Journal of Green Computing*, 1(1):34–39, 2010.

G. Hua, T. C. E. Cheng, and S. Wang. Managing carbon footprints in inventory management. *International Journal of Production Economics*, 132:178–185, 2011.

O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22(4):463–468, 1975.

N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, 1979.

N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the Association for Computing Machinery*, 30(4):852–865, 1983.

P. Schuurman and G. J. Woeginger. Approximation schemes – A tutorial. In R. H. Moehring, C. N. Potts, A. S. Schulz, G. J. Woeginger, and L. A. Wolsey, editors, *Lectures On Scheduling*. 2011. To appear.

United Nations. Kyoto Protocol to the United Nations Framework Convention on Climate Change, 1998. `http://unfccc.int/resource/docs/convkp/kpeng.pdf`.

W. van den Heuvel, H. E. Romeijn, D. Romero Morales, and A. P. M. Wagelmans. Finding pareto optimal solutions for the multi-objective economic lot-sizing problem. In *Proceedings 2nd International Workshop on Lot Sizing 2011, Istanbul, Turkey, 24–26 August 2011*, 2011.

J. C. Vélazquez-Martínez, J. C. Fransoo, E. E. Blanco, and J. Mora-Vargas. The impact of carbon footprinting aggregation on realizing emission reduction targets. Technical report, 2011.

A. P. M. Wagelmans. *Sensitivity Analysis in Combinatorial Optimization*. PhD thesis, Erasmus University Rotterdam, 1990.

A. P. M. Wagelmans, C. P. M. van Hoesel, and A. Kolen. Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, 40:S145–S156, 1992.

H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.

G. J. Woeginger. When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing*, 12(1):57–74, 2000.

A Proof of Theorem 6

Theorem 6 *There exists an optimal solution to ELSEC, such that the single-sourcing property holds in all but (at most) one period.*

Proof Suppose there exists an optimal solution with (at least) two periods with two arcs with positive inflow. We will show that there must exist a solution with single-sourcing in all but at most one period, at equal or lower costs.

First, suppose that period v's demand is procured from periods t and s (i.e., v is a double-sourcing period), and $C'_{t,v}(0) \geq C'_{s,v}(0)$ and $\hat{C}'_{t,v}(0) \geq \hat{C}'_{s,v}(0)$. (Note that this also covers the case $C'_{t,v}(0) \leq C'_{s,v}(0)$ and $\hat{C}'_{t,v}(0) \leq \hat{C}'_{s,v}(0)$, because we can switch the indices t and s.) It was shown in the proof of Theorem 3 that there must exist a solution with at most one period with double-sourcing and lower or equal costs and emissions.

Now, suppose that both periods with double-sourcing, say v_1 and v_2, are in separate blocks. The case with three or more sources in one block is treated later.

Suppose that period v_1's demand is procured from periods t_1 and s_1 and that period v_2's demand is procured from periods t_2 and s_2. Let $v_i := \max\{s_i, t_i\}$, for $i = 1, 2$.

We may assume that $C'_{t_1,v_1}(0) \geq C'_{s_1,v_1}(0)$, $\hat{C}'_{t_1,v_1}(0) < \hat{C}'_{s_1,v_1}(0)$, $C'_{t_2,v_2}(0) \geq C'_{s_2,v_2}(0)$ and $\hat{C}'_{t_2,v_2}(0) < \hat{C}'_{s_2,v_2}(0)$, w.l.o.g., because we may swap t_1 and s_1, or t_2 and s_2.

Now, define the following notation:

$$\frac{C'_{i,j}(0) - C'_{k,j}(0)}{\hat{C}'_{k,j}(0) - \hat{C}'_{i,j}(0)} \quad ,$$

which denotes the financial savings per additional unit of emissions, if we produce (some of) period j's demand in period k instead of period i, near $q_{i,j} = 0$ and $q_{j,j} = 0$ (given that $j = i$ or $j = k$). Suppose

$$\frac{C'_{t_1,v_1}(0) - C'_{s_1,v_1}(0)}{\hat{C}'_{s_1,v_1}(0) - \hat{C}'_{t_1,v_1}(0)} \geq \frac{C'_{t_2,v_2}(0) - C'_{s_2,v_2}(0)}{\hat{C}'_{s_2,v_2}(0) - \hat{C}'_{t_2,v_2}(0)},$$

again w.l.o.g., because we can swap the indices 1 and 2.

We show that it is cheaper and cleaner to move items from period t_1 to s_1 and from s_2 to t_2 until nothing is produced in period t_1 or s_2. We decide to move a quantity $q_1 > 0$ from period t_1 to s_1 and to move a quantity $q_2 > 0$ from period s_2 to t_2. Let $q_2 := \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} q_1$. Moreover, we can choose q_1 such that $q_1 = x_{t_1,v_1}$ or $q_2 = x_{s_2,v_2}$. In other words: such that one of the two blocks has only one source.

First, we show that the costs of the thus constructed solution are lower or equal.

$$\begin{aligned}
& C_{t_1,v_1}(0) - C_{t_1,v_1}(-q_1) + C_{s_2,v_2}(0) - C_{s_2,v_2}(-q_2) \\
\geq\ & C'_{t_1,v_1}(0) q_1 + C'_{s_2,v_2}(0) q_2 \\
=\ & \left(C'_{t_1,v_1} + C'_{s_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} \right) q_1 \\
\geq\ & \left(C'_{s_1,v_1} + C'_{t_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} \right) q_1 \\
=\ & C'_{s_1,v_1}(0) q_1 + C'_{t_2,v_2}(0) q_2 \\
\geq\ & C_{s_1,v_1}(q_1) - C_{s_1,v_1}(0) + C_{t_2,v_2}(q_2) - C_{t_2,v_2}(0)
\end{aligned}$$

That is, the savings are larger than the extra expenses. The first and last inequality follow from concavity. The middle inequality is true, because $q_1 > 0$ and we know that

$$\begin{aligned}
\frac{C'_{t_1,v_1} - C'_{s_1,v_1}}{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}} &\geq \frac{C'_{t_2,v_2} - C'_{s_2,v_2}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} \\
\Rightarrow C'_{t_1,v_1} - C'_{s_1,v_1} &\geq \left(C'_{t_2,v_2} - C'_{s_2,v_2}\right) \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} \\
\Rightarrow C'_{t_1,v_1} + C'_{s_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} &\geq C'_{s_1,v_1} + C'_{t_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}}
\end{aligned}$$

In a similar way, we show that the emissions are lower or equal.

$$\begin{aligned}
& \hat{C}_{t_1,v_1}(0) - \hat{C}_{t_1,v_1}(-q_1) + \hat{C}_{s_2,v_2}(0) - \hat{C}_{s_2,v_2}(-q_2) \\
\geq\ & \hat{C}'_{t_1,v_1}(0)q_1 + \hat{C}'_{s_2,v_2}(0)q_2 \\
=\ & \left(\hat{C}'_{t_1,v_1} + \hat{C}'_{s_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}}\right) q_1 \\
=\ & \left(\hat{C}'_{s_1,v_1} + \hat{C}'_{t_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}}\right) q_1 \\
=\ & \hat{C}'_{s_1,v_1}(0)q_1 + \hat{C}'_{t_2,v_2}(0)q_2 \\
\geq\ & \hat{C}_{s_1,v_1}(q_1) - \hat{C}_{s_1,v_1}(0) + \hat{C}_{t_2,v_2}(q_2) - \hat{C}_{t_2,v_2}(0)
\end{aligned}$$

The middle equality follows from:

$$\begin{aligned}
\hat{C}'_{t_1,v_1} - \hat{C}'_{s_1,v_1} &= -(\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}) = \left(\hat{C}'_{t_2,v_2} - \hat{C}'_{s_2,v_2}\right) \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} \\
\Rightarrow \hat{C}'_{t_1,v_1} + \hat{C}'_{s_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} &= \hat{C}'_{s_1,v_1} + \hat{C}'_{t_2,v_2} \frac{\hat{C}'_{s_1,v_1} - \hat{C}'_{t_1,v_1}}{\hat{C}'_{s_2,v_2} - \hat{C}'_{t_2,v_2}} .
\end{aligned}$$

Suppose that we have a solution with one block with three production periods. Let P denote the set of production periods in this block and let u (v) be the first (last) production period in this block. We will show that there must exist a solution with only two production periods in this block and equal or lower costs and emissions, following a similar reasoning.

We may assume, w.l.o.g., that

$$\begin{aligned}
p'_t(x_t) + \sum_{k=t}^{v-1} h'_k(I_k) \ &\geq\ p'_s(x_s) + \sum_{k=s}^{v-1} h'_k(I_k) \ \geq\ p'_r(x_r) + \sum_{k=r}^{v-1} h'_k(I_k) \quad \text{and} \\
\hat{p}'_t(x_t) + \sum_{k=t}^{v-1} \hat{h}'_k(I_k) \ &<\ \hat{p}'_s(x_s) + \sum_{k=s}^{v-1} \hat{h}'_k(I_k) \ <\ \hat{p}'_r(x_r) + \sum_{k=r}^{v-1} \hat{h}'_k(I_k) \quad ,
\end{aligned}$$

where $t, s, r \in P, t \neq s \neq r \neq t$. We will compare the financial savings per additional unit of emissions, if we produce (some of) period v's demand in period s instead of period t, with the financial savings per additional unit of emissions, if we produce (some of) period v's demand in period r instead of period s (near x_t, x_s, x_r and $I_k \forall k \in \{\min\{t,s,r\},\ldots,v\}$).

We distinguish between two cases:

Case 1: We assume that

$$\frac{p'_t(x_t) + \sum_{k=t}^{v-1} h'_k(I_k) - p'_s(x_s) - \sum_{k=s}^{v-1} h'_k(I_k)}{\hat{p}'_s(x_s) + \sum_{k=s}^{v-1} \hat{h}'_k(I_k) - \hat{p}'_t(x_t) - \sum_{k=t}^{v-1} \hat{h}'_k(I_k)} \geq \frac{p'_s(x_s) + \sum_{k=s}^{v-1} h'_k(I_k) - p'_r(x_r) - \sum_{k=r}^{v-1} h'_k(I_k)}{\hat{p}'_r(x_r) + \sum_{k=r}^{v-1} \hat{h}'_k(I_k) - \hat{p}'_s(x_s) - \sum_{k=s}^{v-1} \hat{h}'_k(I_k)}.$$

(Note that both fractions are nonnegative.) We show that it is cheaper and cleaner to move items from period t to s and from r to s until nothing is produced in period t or r. We decide to move a quantity $q_1 > 0$ from period t to s and to move a quantity $q_2 > 0$ from period r to s. Let $q_2 := \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} q_1$. Moreover, we can choose q_1 such that $q_1 = x_{t,v}$ or $q_2 = x_{r,v}$. In other words: such that there are only two sources in this block.

Case 2: Assume that

$$\frac{p'_t(x_t) + \sum_{k=t}^{v-1} h'_k(I_k) - p'_s(x_s) - \sum_{k=s}^{v-1} h'_k(I_k)}{\hat{p}'_s(x_s) + \sum_{k=s}^{v-1} \hat{h}'_k(I_k) - \hat{p}'_t(x_t) - \sum_{k=t}^{v-1} \hat{h}'_k(I_k)} < \frac{p'_s(x_s) + \sum_{k=s}^{v-1} h'_k(I_k) - p'_r(x_r) - \sum_{k=r}^{v-1} h'_k(I_k)}{\hat{p}'_r(x_r) + \sum_{k=r}^{v-1} \hat{h}'_k(I_k) - \hat{p}'_s(x_s) - \sum_{k=s}^{v-1} \hat{h}'_k(I_k)}.$$

(Note that both fractions are nonnegative.) We show that it is cheaper and cleaner to move items from period s to t and from s to r until nothing is produced in period s. We decide to move a quantity $-q_1 > 0$ from period s to t and to move a quantity $-q_2 > 0$ from period s to r. Again, let $q_2 := \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} q_1$. Moreover, we can choose q_1 such that $-q_1 - q_2 = x_{s,v}$. In other words: such that there are only two sources in this block.

Note that in both cases, we move a quantity q_1 from period t to s and a quantity q_2 from period r to s, but q_1 and q_2 may both be negative depending on the case we are in. Regardless of which case we are in, define $I^*_k := I_k - q_1\delta_{kt} - q_2\delta_{kr} + (q_1 + q_2)\delta_{ks}$, where $\delta_{ij} = \begin{cases} 1 & \text{if } i \geq j \\ 0 & \text{otherwise} \end{cases}$.

Before we show that the costs and emissions of the thus constructed solution are lower or equal, we make two claims:

Claim 8

$$p_t(x_t - q_1) - p_t(x_t) + p_r(x_r - q_2) - p_r(x_r) + p_s(x_s + q_1 + q_2) - p_s(x_s) + \sum_{k=u}^{v-1} (h_k(I^*_k) - h_k(I_k))$$

$$\leq -\left(p_t'(x_t) + \sum_{k=t}^{v-1} h_k'(I_k)\right) q_1 - \left(p_r'(x_r) + \sum_{k=r}^{v-1} h_k'(I_k)\right) q_2 + \left(p_s'(x_s) + \sum_{k=s}^{v-1} h_k'(I_k)\right)(q_1+q_2)$$

Proof This follows from concavity and the fact that we can rewrite $\sum_{k=u}^{v-1}\left(h_k(I_k^*) - h_k(I_k)\right)$. Note that the holding emissions ($\hat{h}$) can be rewritten in the same manner.

Suppose $u = t < s < r = v$. This also proves the case where $r < s < t$, because, in the proof, we can switch r and t, and their corresponding q_1 and q_2.

$$\begin{aligned}
\sum_{k=u}^{v-1}\left(h_k(I_k^*) - h_k(I_k)\right) &= \sum_{k=t}^{s-1}(h_k(I_k - q_1) - h_k(I_k)) + \sum_{k=s}^{v-1}(h_k(I_k + q_2) - h_k(I_k)) \\
&\leq -\sum_{k=t}^{s-1} h_k'(I_k)q_1 + \sum_{k=s}^{v-1} h_k'(I_k)q_2 - \sum_{k=s}^{v-1} h_k'(I_k)q_1 + \sum_{k=s}^{v-1} h_k'(I_k)q_1 \\
&= -\sum_{k=t}^{v-1} h_k'(I_k)q_1 + \sum_{k=s}^{v-1} h_k'(I_k)(q_1+q_2)
\end{aligned}$$

The term $\sum_{k=r}^{v-1} h_k'(I_k)q_2$ is absent, since $r = v$.

Suppose $u = t < r < s = v$. This also proves the case where $r < t < s$.

$$\begin{aligned}
\sum_{k=u}^{v-1}\left(h_k(I_k^*) - h_k(I_k)\right) &= \sum_{k=t}^{r-1}(h_k(I_k - q_1) - h_k(I_k)) + \sum_{k=r}^{v-1}(h_k(I_k - q_1 - q_2) - h_k(I_k)) \\
&\leq -\sum_{k=t}^{r-1} h_k'(I_k)q_1 - \sum_{k=r}^{v-1} h_k'(I_k)(q_1+q_2) \\
&= -\sum_{k=t}^{v-1} h_k'(I_k)q_1 - \sum_{k=r}^{v-1} h_k'(I_k)q_2
\end{aligned}$$

Suppose $u = s < t < r = v$. This also proves the case where $s < r < t$.

$$\begin{aligned}
\sum_{k=u}^{v-1}\left(h_k(I_k^*) - h_k(I_k)\right) &= \sum_{k=s}^{t-1}(h_k(I_k + q_1 + q_2) - h_k(I_k)) + \sum_{k=t}^{v-1}(h_k(I_k + q_2) - h_k(I_k)) \\
&\leq \sum_{k=s}^{t-1} h_k'(I_k)(q_1+q_2) + \sum_{k=t}^{v-1} h_k'(I_k)q_2 + \sum_{k=t}^{v-1} h_k'(I_k)q_1 - \sum_{k=t}^{v-1} h_k'(I_k)q_1 \\
&= \sum_{k=s}^{v-1} h_k'(I_k)(q_1+q_2) - \sum_{k=t}^{v-1} h_k'(I_k)q_1
\end{aligned}$$

□

Claim 9

$$\left(C_{s,v}' - C_{t,v}' + \left(C_{s,v}' - C_{r,v}'\right)\frac{\hat{C}_{s,v}' - \hat{C}_{t,v}'}{\hat{C}_{r,v}' - \hat{C}_{s,v}'}\right) q_1 \leq 0$$

Proof In Case 1: $q_1 > 0$ and by assumption, we know that:

$$\begin{aligned}
\frac{C'_{t,v} - C'_{s,v}}{\hat{C}'_{s,v} - \hat{C}'_{t,v}} &\geq \frac{C'_{s,v} - C'_{r,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow \frac{C'_{s,v} - C'_{t,v}}{\hat{C}'_{s,v} - \hat{C}'_{t,v}} &\leq \frac{C'_{r,v} - C'_{s,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow C'_{s,v} - C'_{t,v} &\leq \left(C'_{r,v} - C'_{s,v}\right) \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow C'_{s,v} - C'_{t,v} + \left(C'_{s,v} - C'_{r,v}\right) \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} &\leq 0
\end{aligned}$$

In Case 2: $q_1 < 0$ and by assumption, we know that:

$$\begin{aligned}
\frac{C'_{t,v} - C'_{s,v}}{\hat{C}'_{s,v} - \hat{C}'_{t,v}} &< \frac{C'_{s,v} - C'_{r,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow \frac{C'_{s,v} - C'_{t,v}}{\hat{C}'_{s,v} - \hat{C}'_{t,v}} &> \frac{C'_{r,v} - C'_{s,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow C'_{s,v} - C'_{t,v} &> \left(C'_{r,v} - C'_{s,v}\right) \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} \\
\Rightarrow C'_{s,v} - C'_{t,v} + \left(C'_{s,v} - C'_{r,v}\right) \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} &> 0
\end{aligned}$$

□

Now, we show that the costs of the constructed solution are lower or equal:

$$p_t(x_t - q_1) - p_t(x_t) + p_r(x_r - q_2) - p_r(x_r) + p_s(x_s + q_1 + q_2) - p_s(x_s) + \sum_{k=u}^{v-1} \left(h_k(I_k^*) - h_k(I_k)\right)$$

$$\begin{aligned}
&\leq -\left(p'_t(x_t) + \sum_{k=t}^{v-1} h'_k(I_k)\right) q_1 - \left(p'_r(x_r) + \sum_{k=r}^{v-1} h'_k(I_k)\right) q_2 + \left(p'_s(x_s) + \sum_{k=s}^{v-1} h'_k(I_k)\right) (q_1 + q_2) \\
&= -C'_{t,v} q_1 - C'_{r,v} q_2 + C'_{s,v} \cdot (q_1 + q_2) \\
&= -C'_{t,v} q_1 - C'_{r,v} \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} q_1 + C'_{s,v} \left(q_1 + \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}} q_1\right) \\
&= \left(C'_{s,v} - C'_{t,v} + \left(C'_{s,v} - C'_{r,v}\right) \frac{\hat{C}'_{s,v} - \hat{C}'_{t,v}}{\hat{C}'_{r,v} - \hat{C}'_{s,v}}\right) q_1 \\
&\leq 0 \quad ,
\end{aligned}$$

where the first inequality follows from Claim 8 and the last inequality from Claim 9.

In a similar way, we show that the emissions are lower or equal.

$$\hat{p}_t(x_t - q_1) - \hat{p}_t(x_t) + \hat{p}_r(x_r - q_2) - \hat{p}_r(x_r) + \hat{p}_s(x_s + q_1 + q_2) - \hat{p}_s(x_s) + \sum_{k=u}^{v-1} \left(\hat{h}_k(I_k^*) - \hat{h}_k(I_k)\right)$$

$$\begin{aligned}
&\leq -\left(\hat{p}_t'(x_t) + \sum_{k=t}^{v-1} \hat{h}_k'(I_k)\right) q_1 - \left(\hat{p}_r'(x_r) + \sum_{k=r}^{v-1} \hat{h}_k'(I_k)\right) q_2 + \left(\hat{p}_s'(x_s) + \sum_{k=s}^{v-1} \hat{h}_k'(I_k)\right) (q_1 + q_2) \\
&= -\hat{C}_{t,v}' q_1 - \hat{C}_{r,v}' q_2 + \hat{C}_{s,v}' \cdot (q_1 + q_2) \\
&= -\hat{C}_{t,v}' q_1 - \hat{C}_{r,v}' \frac{\hat{C}_{s,v}' - \hat{C}_{t,v}'}{\hat{C}_{r,v}' - \hat{C}_{s,v}'} q_1 + C_{s,v}' \left(q_1 + \frac{\hat{C}_{s,v}' - \hat{C}_{t,v}'}{\hat{C}_{r,v}' - \hat{C}_{s,v}'} q_1\right) \\
&= \left(\hat{C}_{s,v}' - \hat{C}_{t,v}' + (\hat{C}_{s,v}' - \hat{C}_{r,v}') \frac{\hat{C}_{s,v}' - \hat{C}_{t,v}'}{\hat{C}_{r,v}' - \hat{C}_{s,v}'}\right) q_1 \\
&= (\hat{C}_{s,v}' - \hat{C}_{t,v}' - \hat{C}_{s,v}' + \hat{C}_{t,v}') q_1 \\
&= 0 \quad ,
\end{aligned}$$

where the first inequality follows from the analogy of Claim 8 for emissions instead of costs.

We conclude that there exists an optimal solution to ELSEC, such that the single-sourcing property holds in all but (at most) one period. □

B Tables of results

B.1 Results with improved lower bound

Tables 2–8 present the results of the computational tests of the algorithms that use the improved lower bound, as described in Sections 5.3 and 5.4.

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	<0.001	0.001	<0.001	0.001	<0.001	0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
	avg. post. gap (%)	0.85	0.67	0.73	1.8	1.6	1.9	2.5	1.6	1.9	1.9	1.5	1.1
	avg. true gap (%)	0.29	0.21	0.18	0.47	0.51	0.48	0.88	0.50	0.69	0.53	0.52	0.36
	solved to opt. (%)	57	63	63	80	67	67	53	70	50	60	60	70
PP-CB	avg. sol. time (s)	0.12	0.12	0.13	0.25	0.26	0.22	0.34	0.34	0.34	0.26	0.23	0.22
FPTAS-CB-LB(0.1)	avg. sol. time (s)	0.001	0.001	<0.001	0.002	<0.001	0.002	0.001	<0.001	0.003	0.001	0.001	0.001
	avg. post. gap (%)	0.58	0.51	0.57	0.87	0.88	0.89	1.1	0.91	0.98	0.97	0.83	0.63
	avg. true gap (%)	0.012	0.051	0.036	0.0032	0.0045	0	0.010	0.015	0.060	0.015	0.015	0.035
	solved to opt. (%)	83	70	83	93	97	100	90	93	90	92	90	84
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.002	0.001	<0.001	0.002	0.001	0.002	0.002	0.001	0.003	0.001	0.002	0.001
	avg. post. gap (%)	0.50	0.40	0.44	0.54	0.57	0.56	0.71	0.59	0.59	0.63	0.58	0.43
	avg. true gap (%)	0.0033	0.011	0.0013	0.0032	0	0	0.00048	0	0	0.0017	0.00059	0.0041
	solved to opt. (%)	90	90	93	93	100	100	97	100	100	97	97	94
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.005	0.004	0.007	0.008	0.005	0.004	0.008	0.007	0.005	0.007	0.005	0.005
	avg. post. gap (%)	0.17	0.13	0.18	0.13	0.14	0.15	0.16	0.14	0.14	0.16	0.15	0.14
	avg. true gap (%)	0.0013	0	0.0013	0	0.00073	0	0.00048	0.00012	0	0.00022	0.00038	0.00072
	solved to opt. (%)	97	100	93	100	97	100	97	97	100	99	98	97
FPTAS-CB(0.1)	avg. sol. time (s)	0.006	0.008	0.007	0.009	0.009	0.008	0.008	0.008	0.010	0.009	0.008	0.008
	avg. post. gap (%)	3.8	3.8	3.8	3.1	3.3	3.2	3.2	3.1	3.1	3.4	3.4	3.4
	avg. true gap (%)	0.0061	0.029	0.020	0.00012	0.0019	0	0.0048	0.0050	0.026	0.0072	0.0061	0.018
	solved to opt. (%)	87	80	83	97	90	100	93	93	93	93	92	87
FPTAS-CB(0.05)	avg. sol. time (s)	0.017	0.016	0.016	0.020	0.018	0.017	0.018	0.018	0.019	0.019	0.018	0.016
	avg. post. gap (%)	1.9	1.8	1.8	1.5	1.7	1.6	1.6	1.6	1.5	1.7	1.7	1.7
	avg. true gap (%)	0.0068	0.0082	0.0013	0.00012	0.00073	0	0.0020	0.00012	0	0.0029	0.0025	0.0010
	solved to opt. (%)	87	87	93	97	97	100	97	97	100	96	94	94
FPTAS-CB(0.01)	avg. sol. time (s)	0.086	0.084	0.085	0.099	0.095	0.097	0.099	0.094	0.099	0.097	0.093	0.090
	avg. post. gap (%)	0.38	0.36	0.38	0.31	0.33	0.32	0.32	0.31	0.31	0.33	0.33	0.34
	avg. true gap (%)	0	0	0.00067	0.00012	0	0	0	0	0	0.00022	0	0.000041
	solved to opt. (%)	100	100	97	97	100	100	100	100	100	99	100	99
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.003	0.002	0.002	0.003	0.003	0.003	0.002	0.003	0.004	0.002	0.003	0.002
	avg. post gap (%)	0.58	0.49	0.55	1.4	1.0	1.4	1.5	1.1	1.2	1.3	0.98	0.73
	avg. true gap (%)	0.0061	0.013	0.0095	0.0032	0.0056	0	0.0055	0.0050	0	0.0042	0.0039	0.0079
	solved to opt. (%)	87	80	87	93	90	100	93	93	100	93	94	87
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.002	0.002	0.004	0.003	0.003	0.005	0.007	0.003	0.004	0.004	0.003	0.004
	avg. post gap (%)	0.58	0.47	0.55	1.2	0.99	1.1	1.2	1.0	1.1	1.1	0.94	0.67
	avg. true gap (%)	0.0032	0	0.0038	0.00012	0.00028	0	0	0	0	0.00063	0.00023	0.0016
	solved to opt. (%)	90	100	90	97	97	100	100	100	100	98	99	94
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.014	0.012	0.011	0.019	0.014	0.016	0.018	0.020	0.019	0.017	0.015	0.015
	avg. post gap (%)	0.39	0.31	0.34	0.42	0.44	0.42	0.50	0.42	0.45	0.47	0.43	0.33
	avg. true gap (%)	0	0	0	0.00012	0	0	0	0	0	0	0	0.000041
	solved to opt. (%)	100	100	100	97	100	100	100	100	100	100	100	99
FPTAS-gen(0.1)	avg. sol. time (s)	0.021	0.020	0.021	0.022	0.024	0.022	0.023	0.022	0.021	0.022	0.022	0.021
	avg. post gap (%)	6.8	6.8	6.8	6.5	6.6	6.5	6.5	6.5	6.4	6.6	6.6	6.6
	avg. true gap (%)	0.0032	0.015	0.017	0.00012	0	0	0.0024	0	0	0.0021	0.0056	0.0050
	solved to opt. (%)	90	87	80	97	100	100	93	100	100	97	93	92
FPTAS-gen(0.05)	avg. sol. time (s)	0.046	0.043	0.043	0.050	0.046	0.047	0.047	0.045	0.049	0.049	0.045	0.045
	avg. post gap (%)	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.2	3.2	3.3	3.3	3.3
	avg. true gap (%)	0.0013	0.00077	0.0013	0.00012	0.00073	0	0	0	0	0.00022	0.00023	0.00098
	solved to opt. (%)	97	97	93	97	97	100	100	100	100	99	99	96
FPTAS-gen(0.01)	avg. sol. time (s)	0.26	0.25	0.25	0.29	0.28	0.28	0.28	0.28	0.29	0.28	0.27	0.26
	avg. post gap (%)	0.69	0.68	0.69	0.65	0.66	0.66	0.66	0.65	0.65	0.67	0.67	0.66
	avg. true gap (%)	0	0	0	0.00012	0	0	0	0.00012	0	0	0	0.000081
	solved to opt. (%)	100	100	100	97	100	100	100	97	100	100	100	98
CPLEX 10.1 Nat.	avg. sol. time (s)	0.034	0.025	0.025	0.052	0.050	0.053	0.056	0.061	0.055	0.048	0.045	0.044
CPLEX 10.1 SP	avg. sol. time (s)	0.036	0.030	0.024	0.031	0.026	0.026	0.036	0.031	0.030	0.030	0.032	0.028

Table 2: 25 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	<0.001	<0.001	<0.001	0.001	<0.001	0.001	0.002	0.002	0.001	0.001	<0.001	0.001
	avg. post. gap (%)	0.42	0.45	0.36	0.89	0.98	0.98	1.2	1.3	1.1	1.2	0.74	0.63
	avg. true gap (%)	0.20	0.21	0.17	0.44	0.46	0.43	0.67	0.64	0.43	0.54	0.34	0.34
	solved to opt. (%)	50	30	30	53	40	47	47	53	47	37	44	51
PP-CB	avg. sol. time (s)	0.84	0.81	0.89	1.8	1.8	1.7	2.7	2.8	2.7	1.9	1.7	1.6
FPTAS-CB-LB(0.10)	avg. sol. time (s)	0.005	0.005	0.006	0.006	0.009	0.006	0.007	0.008	0.007	0.007	0.005	0.008
	avg. post. gap (%)	0.25	0.30	0.24	0.41	0.52	0.54	0.54	0.52	0.60	0.59	0.41	0.31
	avg. true gap (%)	0.040	0.055	0.048	0.012	0.0023	0.012	0.0094	0	0.035	0.016	0.029	0.026
	solved to opt. (%)	73	43	47	90	97	87	93	100	80	87	76	74
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.007	0.008	0.006	0.010	0.007	0.009	0.011	0.012	0.011	0.010	0.009	0.008
	avg. post. gap (%)	0.22	0.27	0.20	0.33	0.42	0.40	0.42	0.40	0.42	0.43	0.33	0.26
	avg. true gap (%)	0.0098	0.021	0.0071	0.0067	0.0023	0.0029	0.00033	0	0.010	0.0046	0.0072	0.0083
	solved to opt. (%)	87	67	80	93	97	93	97	100	93	94	88	87
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.038	0.039	0.039	0.045	0.049	0.048	0.057	0.059	0.059	0.051	0.048	0.046
	avg. post. gap (%)	0.13	0.13	0.13	0.11	0.12	0.11	0.11	0.11	0.1	0.13	0.12	0.10
	avg. true gap (%)	0	0.00043	0.00086	0	0	0.00015	0	0	0	0.00025	0	0.00024
	solved to opt. (%)	100	93	93	100	100	97	100	100	100	98	100	97
FPTAS-CB(0.10)	avg. sol. time (s)	0.046	0.045	0.046	0.055	0.056	0.053	0.056	0.056	0.056	0.054	0.051	0.051
	avg. post. gap (%)	3.9	3.9	3.8	3.3	3.4	3.3	3.2	3.2	3.2	3.4	3.4	3.4
	avg. true gap (%)	0.038	0.031	0.036	0.0035	0.025	0.037	0	0	0.0083	0.017	0.025	0.018
	solved to opt. (%)	63	63	57	97	77	70	100	100	93	82	79	79
FPTAS-CB(0.05)	avg. sol. time (s)	0.092	0.093	0.093	0.11	0.11	0.11	0.12	0.12	0.12	0.11	0.11	0.10
	avg. post. gap (%)	1.9	1.9	1.9	1.7	1.7	1.6	1.6	1.6	1.6	1.7	1.7	1.7
	avg. true gap (%)	0.0080	0.014	0.018	0.0013	0	0.0061	0.0018	0	0	0.0033	0.0067	0.0062
	solved to opt. (%)	93	67	60	97	100	90	97	100	100	92	89	87
FPTAS-CB(0.01)	avg. sol. time (s)	0.58	0.57	0.57	0.69	0.71	0.71	0.73	0.74	0.75	0.71	0.67	0.64
	avg. post. gap (%)	0.38	0.38	0.37	0.33	0.33	0.32	0.31	0.31	0.31	0.34	0.34	0.34
	avg. true gap (%)	0	0.00084	0.00036	0	0	0.00015	0	0	0	0.000051	0.000088	0.00031
	solved to opt. (%)	100	87	97	100	100	97	100	100	100	99	99	96
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.011	0.013	0.010	0.014	0.013	0.012	0.013	0.017	0.015	0.015	0.013	0.011
	avg. post gap (%)	0.23	0.26	0.21	0.45	0.52	0.54	0.56	0.61	0.63	0.64	0.40	0.30
	avg. true gap (%)	0.013	0.016	0.013	0.0061	0	0.00015	0.0011	0	0.010	0.0034	0.0070	0.0094
	solved to opt. (%)	83	70	67	93	100	97	97	100	93	94	88	84
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.019	0.019	0.020	0.022	0.025	0.024	0.031	0.029	0.031	0.026	0.024	0.024
	avg. post gap (%)	0.22	0.25	0.20	0.45	0.52	0.54	0.56	0.61	0.63	0.64	0.40	0.29
	avg. true gap (%)	0.0027	0.0019	0.0026	0	0	0.00015	0	0	0.0030	0.00069	0.0015	0.0012
	solved to opt. (%)	90	80	83	100	100	97	100	100	97	96	96	91
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.10	0.10	0.11	0.12	0.13	0.13	0.15	0.16	0.15	0.14	0.13	0.12
	avg. post gap (%)	0.21	0.24	0.18	0.31	0.38	0.34	0.39	0.40	0.39	0.39	0.31	0.24
	avg. true gap (%)	0	0.00057	0	0	0	0.00015	0	0	0	0.000051	0.000050	0.00014
	solved to opt. (%)	100	90	100	100	100	97	100	100	100	99	99	98
FPTAS-gen(0.1)	avg. sol. time (s)	0.12	0.11	0.12	0.14	0.14	0.14	0.14	0.14	0.15	0.14	0.13	0.13
	avg. post gap (%)	6.8	6.8	6.8	6.6	6.6	6.6	6.5	6.5	6.5	6.6	6.6	6.6
	avg. true gap (%)	0.0058	0.013	0.017	0	0.0020	0.0034	0.0018	0	0	0.0037	0.0037	0.0069
	solved to opt. (%)	83	70	57	100	97	93	97	100	100	91	90	84
FPTAS-gen(0.05)	avg. sol. time (s)	0.25	0.25	0.26	0.30	0.30	0.30	0.31	0.32	0.31	0.30	0.29	0.28
	avg. post gap (%)	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	avg. true gap (%)	0.0027	0.00073	0.0039	0	0	0.00015	0	0	0	0.00099	0.00035	0.0012
	solved to opt. (%)	90	87	80	100	100	97	100	100	100	96	98	91
FPTAS-gen(0.01)	avg. sol. time (s)	1.7	1.6	1.7	1.9	2.0	2.0	2.0	2.1	2.1	2.0	1.9	1.8
	avg. post gap (%)	0.69	0.69	0.68	0.66	0.66	0.66	0.66	0.66	0.66	0.67	0.67	0.67
	avg. true gap (%)	0	0.00058	0	0	0	0	0	0	0	0	0.000050	0.00014
	solved to opt. (%)	100	90	100	100	100	100	100	100	100	100	99	98
CPLEX 10.1 Nat.	avg. sol. time (s)	0.066	0.075	0.064	0.38	0.32	0.26	0.86	0.97	0.97	0.63	0.48	0.22
CPLEX 10.1 SP	avg. sol. time (s)	0.061	0.063	0.065	0.070	0.069	0.072	0.076	0.067	0.075	0.073	0.069	0.064

Table 3: 50 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	0.001	0.001	0.002	0.003	0.003	0.002	0.003	0.004	0.005	0.003	0.003	0.002
	avg. post. gap (%)	0.21	0.22	0.22	0.40	0.41	0.48	0.55	0.60	0.64	0.64	0.39	0.21
	avg. true gap (%)	0.15	0.16	0.15	0.24	0.22	0.31	0.31	0.36	0.44	0.42	0.25	0.11
	solved to opt. (%)	23	27	30	37	33	30	40	40	30	21	28	48
PP-CB	avg. sol. time (s)	14	14	14	22	21	21	31	30	30	23	22	21
FPTAS-CB-LB(0.10)	avg. sol. time (s)	0.029	0.029	0.029	0.037	0.036	0.036	0.042	0.044	0.046	0.038	0.035	0.036
	avg. post. gap (%)	0.090	0.080	0.090	0.17	0.20	0.20	0.25	0.24	0.20	0.23	0.16	0.12
	avg. true gap (%)	0.029	0.018	0.021	0.0083	0.016	0.021	0.016	0.0035	0.0033	0.010	0.014	0.022
	solved to opt. (%)	43	60	50	73	73	67	73	93	87	77	70	60
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.054	0.051	0.057	0.071	0.067	0.070	0.079	0.083	0.080	0.071	0.068	0.064
	avg. post. gap (%)	0.060	0.060	0.090	0.16	0.19	0.18	0.23	0.22	0.20	0.21	0.15	0.11
	avg. true gap (%)	0.0029	0.0055	0.017	0.0027	0.0057	0.0045	0.0097	0.0050	0.00079	0.0031	0.0059	0.0089
	solved to opt. (%)	83	77	57	80	93	87	87	90	90	88	80	80
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.29	0.28	0.29	0.39	0.38	0.40	0.48	0.49	0.48	0.41	0.38	0.37
	avg. post. gap (%)	0.060	0.060	0.070	0.090	0.080	0.090	0.080	0.070	0.080	0.086	0.075	0.065
	avg. true gap (%)	0	0	0.000080	0.00064	0	0.00023	0	0	0.00034	0.00023	0.00018	0.000012
	solved to opt. (%)	100	100	97	93	100	97	100	100	93	98	97	99
FPTAS-CB(0.10)	avg. sol. time (s)	0.29	0.29	0.29	0.36	0.36	0.37	0.40	0.41	0.41	0.37	0.35	0.34
	avg. post. gap (%)	3.9	3.9	3.9	3.4	3.4	3.3	3.2	3.2	3.2	3.5	3.5	3.5
	avg. true gap (%)	0.039	0.035	0.027	0.0077	0.015	0.011	0.011	0.0038	0.0053	0.019	0.014	0.018
	solved to opt. (%)	37	50	47	80	73	70	77	93	83	69	66	69
FPTAS-CB(0.05)	avg. sol. time (s)	0.62	0.61	0.63	0.80	0.78	0.80	0.90	0.91	0.90	0.81	0.77	0.75
	avg. post. gap (%)	1.9	1.9	1.9	1.7	1.7	1.7	1.6	1.6	1.6	1.7	1.7	1.7
	avg. true gap (%)	0.0083	0.010	0.0063	0.0021	0.00084	0.0043	0.0037	0.00053	0.0014	0.0044	0.0039	0.0042
	solved to opt. (%)	80	67	67	83	93	83	93	97	90	84	83	83
FPTAS-CB(0.01)	avg. sol. time (s)	4.3	4.1	4.3	5.5	5.3	5.5	6.1	6.2	6.1	5.5	5.2	5.1
	avg. post. gap (%)	0.37	0.38	0.38	0.34	0.34	0.33	0.32	0.31	0.32	0.34	0.34	0.34
	avg. true gap (%)	0	0.00081	0.00010	0	0	0.00014	0	0	0.00030	0.00015	0.000013	0.00028
	solved to opt. (%)	100	97	90	100	100	97	100	100	97	97	99	98
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.068	0.066	0.068	0.084	0.081	0.085	0.098	0.10	0.099	0.086	0.082	0.082
	avg. post gap (%)	0.07	0.06	0.08	0.16	0.19	0.18	0.25	0.24	0.20	0.22	0.15	0.10
	avg. true gap (%)	0.013	0.0054	0.0063	0.0010	0.00066	0.0016	0.0089	0.0035	0.0028	0.0030	0.0054	0.0059
	solved to opt. (%)	57	77	67	93	93	90	90	93	87	87	80	82
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.13	0.13	0.13	0.17	0.16	0.17	0.19	0.20	0.19	0.17	0.16	0.16
	avg. post gap (%)	0.06	0.06	0.08	0.16	0.19	0.18	0.24	0.24	0.20	0.22	0.15	0.10
	avg. true gap (%)	0.0021	0.0032	0.0017	0.0013	0.00058	0.0017	0.00035	0.0012	0.00034	0.00076	0.0019	0.0015
	solved to opt. (%)	90	87	83	93	93	90	97	97	93	94	87	93
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.88	0.84	0.86	1.2	1.1	1.2	1.4	1.4	1.4	1.2	1.1	1.1
	avg. post gap (%)	0.06	0.06	0.07	0.16	0.19	0.18	0.23	0.23	0.20	0.21	0.15	0.099
	avg. true gap (%)	0	0	0.000078	0	0	0	0	0	0	0	0.000013	0.000013
	solved to opt. (%)	100	100	93	100	100	100	100	100	100	100	99	99
FPTAS-gen(0.1)	avg. sol. time (s)	0.78	0.76	0.78	0.96	0.95	0.98	1.1	1.1	1.1	0.98	0.93	0.91
	avg. post gap (%)	6.8	6.8	6.8	6.6	6.6	6.6	6.5	6.5	6.5	6.6	6.6	6.6
	avg. true gap (%)	0.0058	0.0053	0.014	0.0035	0.0036	0.0064	0.0013	0.00097	0.00034	0.0038	0.0053	0.0046
	solved to opt. (%)	77	83	53	83	90	77	93	97	93	86	81	82
FPTAS-gen(0.05)	avg. sol. time (s)	1.8	1.7	1.8	2.2	2.2	2.2	2.5	2.5	2.5	2.3	2.1	2.1
	avg. post gap (%)	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	avg. true gap (%)	0.0027	0.0043	0.0043	0.0016	0.00029	0.0011	0.0011	0	0.00030	0.0016	0.0019	0.0017
	solved to opt. (%)	87	77	77	87	97	93	93	100	97	89	88	92
FPTAS-gen(0.01)	avg. sol. time (s)	11	11	11	14	14	14	15	16	15	14	14	13
	avg. post gap (%)	0.69	0.69	0.69	0.67	0.67	0.66	0.66	0.66	0.66	0.67	0.67	0.67
	avg. true gap (%)	0	0	0.00018	0	0	0.00023	0	0	0.000037	0.000007	0.00013	0.000012
	solved to opt. (%)	100	100	87	100	100	97	100	100	97	99	96	99
CPLEX 10.1 SP	avg. sol. time (s)	0.18	0.19	0.20	0.23	0.25	0.24	0.24	0.25	0.25	0.26	0.22	0.18

Table 4: 100 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	<0.001	<0.001	0.001	<0.001	<0.001	0.001	0.001	0.001	0.001	<0.001	<0.001	0.001
	avg. post gap (%)	2.5	2.4	2.0	2.4	3.1	3.4	3.7	2.9	2.6	3.4	2.5	2.3
	avg. true gap (%)	1.5	1.4	1.2	0.95	0.98	1.7	1.2	0.74	0.75	1.5	0.84	1.2
	solved to opt. (%)	23	23	23	70	63	27	63	50	47	33	44	52
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.006	0.004	0.004	0.004	0.007	0.004	0.006	0.006	0.007	0.005	0.004	0.006
	avg. post gap (%)	1.1	1.0	1.1	1.5	1.9	1.7	2.4	2.1	1.8	1.9	1.7	1.2
	avg. true gap (%)	0.20	0.073	0.20	0.015	0.0072	0.027	0.024	0.012	0.0044	0.081	0.074	0.034
	solved to opt. (%)	43	30	43	87	93	80	90	93	87	64	71	80
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.009	0.007	0.009	0.006	0.007	0.012	0.012	0.009	0.010	0.011	0.009	0.007
	avg. post gap (%)	0.97	1.0	1.0	1.3	1.5	1.5	1.7	1.7	1.5	1.6	1.4	1.0
	avg. true gap (%)	0.10	0.066	0.17	0.00044	0.0072	0.014	0.012	0	0.0021	0.067	0.026	0.030
	solved to opt. (%)	47	47	43	97	93	83	97	100	93	70	78	86
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.096	0.072	0.084	0.070	0.082	0.093	0.091	0.079	0.076	0.098	0.082	0.068
	avg. post gap (%)	0.38	0.51	0.42	0.43	0.49	0.48	0.47	0.53	0.41	0.50	0.49	0.39
	avg. true gap (%)	0.014	0.044	0.030	0	0	0.011	0	0	0.0011	0.016	0.012	0.0046
	solved to opt. (%)	73	53	67	100	100	90	100	100	97	82	86	92
FPTAS-gen(0.1)	avg. sol. time (s)	0.065	0.055	0.061	0.048	0.055	0.053	0.049	0.046	0.051	0.060	0.053	0.048
	avg. post gap (%)	6.7	6.8	6.8	6.5	6.6	6.5	6.5	6.5	6.5	6.6	6.6	6.6
	avg. true gap (%)	0.045	0.042	0.047	0.0057	0.0017	0.0081	0	0	0.0032	0.024	0.020	0.0071
	solved to opt. (%)	57	43	57	97	97	90	100	100	90	78	81	87
FPTAS-gen(0.05)	avg. sol. time (s)	0.19	0.15	0.17	0.12	0.14	0.14	0.13	0.12	0.12	0.16	0.14	0.13
	avg. post gap (%)	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	avg. true gap (%)	0.020	0.023	0.0015	0	0	0.010	0	0	0.0044	0.0064	0.013	0.0053
	solved to opt. (%)	73	50	67	100	100	80	100	100	87	83	81	88
FPTAS-gen(0.01)	avg. sol. time (s)	3.2	2.4	2.9	1.7	2.0	2.1	1.7	1.6	1.7	2.5	2.1	1.9
	avg. post gap (%)	0.66	0.67	0.67	0.66	0.66	0.66	0.66	0.65	0.65	0.66	0.66	0.66
	avg. true gap (%)	0.00081	0.048	0.0074	0	0	0	0	0	0	0.0026	0.0012	0.00059
	solved to opt. (%)	93	83	77	100	100	100	100	100	100	91	96	98
CPLEX 10.1 Nat.	avg. sol. time (s)	0.028	0.025	0.025	0.044	0.046	0.053	0.050	0.053	0.045	0.045	0.042	0.035
CPLEX 10.1 SP	avg. sol. time (s)	0.036	0.029	0.025	0.032	0.031	0.031	0.036	0.030	0.030	0.036	0.029	0.028

Table 5: 25 periods with 13 pairs that violate the co-behaviour property

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	<0.001	<0.001	<0.001	0.001	0.001	0.001	0.001	0.001	0.001	<0.001	0.001	0.001
	avg. post gap (%)	1.2	0.93	0.91	1.4	1.4	1.7	1.4	1.3	1.7	1.6	1.5	0.90
	avg. true gap (%)	0.83	0.60	0.59	0.87	0.78	1.0	0.76	0.50	0.69	0.86	0.88	0.48
	solved to opt. (%)	6.7	20	20	27	33	47	53	50	27	30	21	43
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.027	0.027	0.023	0.030	0.027	0.027	0.034	0.034	0.033	0.032	0.029	0.027
	avg. post gap (%)	0.44	0.38	0.36	0.58	0.60	0.67	0.66	0.84	1.0	0.79	0.63	0.43
	avg. true gap (%)	0.090	0.050	0.042	0.016	0.0077	0.0029	0	0.0058	0.0055	0.032	0.020	0.021
	solved to opt. (%)	27	47	53	87	87	83	100	93	97	64	76	84
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.058	0.058	0.055	0.060	0.061	0.064	0.072	0.069	0.074	0.070	0.063	0.058
	avg. post gap (%)	0.44	0.39	0.36	0.58	0.60	0.67	0.66	0.83	0.99	0.79	0.63	0.42
	avg. true gap (%)	0.085	0.046	0.037	0.012	0.0066	0.0029	0	0.078	0	0.030	0.021	0.014
	solved to opt. (%)	27	37	60	87	90	83	100	97	100	66	76	86
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.66	0.68	0.64	0.64	0.64	0.64	0.76	0.75	0.81	0.77	0.68	0.62
	avg. post gap (%)	0.33	0.32	0.30	0.42	0.36	0.40	0.43	0.42	0.47	0.45	0.40	0.30
	avg. true gap (%)	0.033	0.023	0.019	0.0067	0.0053	0.0022	0	0	0	0.018	0.0084	0.0036
	solved to opt. (%)	47	53	60	93	93	93	100	100	100	70	83	93
FPTAS-gen(0.1)	avg. sol. time (s)	0.44	0.45	0.43	0.42	0.40	0.43	0.40	0.41	0.40	0.46	0.41	0.39
	avg. post gap (%)	6.8	6.8	6.7	6.6	6.6	6.6	6.5	6.5	6.4	6.6	6.6	6.6
	avg. true gap (%)	0.045	0.026	0.0083	0.0011	0.011	0.0060	0.0019	0.0050	0	0.014	0.012	0.0086
	solved to opt. (%)	37	47	70	97	80	83	97	93	100	71	79	84
FPTAS-gen(0.05)	avg. sol. time (s)	1.3	1.4	1.3	1.2	1.1	1.2	1.1	1.1	1.1	1.3	1.2	1.1
	avg. post gap (%)	3.4	3.4	3.4	3.3	3.3	3.3	3.2	3.3	3.2	3.3	3.3	3.3
	avg. true gap (%)	0.025	0.014	0.010	0.0083	0.0054	0.0022	0	0	0	0.012	0.0058	0.0036
	solved to opt. (%)	57	63	63	90	90	93	100	100	100	76	86	91
FPTAS-gen(0.01)	avg. sol. time (s)	25	26	24	19	18	20	17	16	16	22	20	18
	avg. post gap (%)	0.67	0.68	0.68	0.66	0.66	0.67	0.65	0.65	0.65	0.66	0.66	0.66
	avg. true gap (%)	0.0011	0.0015	0.0010	0.00039	0	0.00066	0	0	0	0.00065	0.00069	0.00021
	solved to opt. (%)	80	87	87	97	100	97	100	100	100	90	94	98
CPLEX 10.1 Nat.	avg. sol. time (s)	0.049	0.049	0.049	0.27	0.25	0.21	0.88	1.0	0.63	0.51	0.43	0.19
CPLEX 10.1 SP	avg. sol. time (s)	0.066	0.065	0.065	0.080	0.079	0.078	0.067	0.075	0.072	0.074	0.075	0.066

Table 6: 50 periods with 25 pairs that violate the co-behaviour property

	K	1000			5000			10000			$\hat{C}$		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000	25%	50%	75%
Megiddo	avg. sol. time (s)	0.001	0.002	0.002	0.002	0.002	0.005	0.004	0.004	0.006	0.004	0.003	0.002
	avg. post gap (%)	0.55	0.36	0.44	0.75	0.81	0.66	0.52	0.74	0.63	0.84	0.58	0.40
	avg. true gap (%)	0.44	0.24	0.34	0.48	0.60	0.44	0.31	0.45	0.39	0.60	0.39	0.24
	solved to opt. (%)	3.3	13	0	23	20	37	33	33	27	12	23	28
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.17	0.18	0.17	0.21	0.21	0.20	0.22	0.23	0.23	0.22	0.20	0.19
	avg. post gap (%)	0.14	0.15	0.16	0.28	0.23	0.22	0.21	0.28	0.24	0.26	0.20	0.18
	avg. true gap (%)	0.033	0.031	0.053	0.0097	0.017	0.0023	0.00050	0.0011	0.0047	0.024	0.012	0.015
	solved to opt. (%)	7	27	7	83	80	93	93	90	83	56	66	67
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.40	0.42	0.40	0.49	0.50	0.48	0.53	0.55	0.57	0.53	0.47	0.45
	avg. post gap (%)	0.13	0.14	0.15	0.28	0.23	0.22	0.21	0.28	0.24	0.26	0.20	0.17
	avg. true gap (%)	0.022	0.026	0.048	0.0091	0.016	0.0011	0.00050	0.0011	0.000077	0.023	0.0098	0.0082
	solved to opt. (%)	13	37	13	87	83	93	93	90	97	56	69	78
FPTAS-gen-LB(0.01)	avg. sol. time (s)	4.8	5.4	4.9	5.6	5.8	5.6	6.0	6.4	6.5	6.3	5.6	5.2
	avg. post gap (%)	0.12	0.13	0.13	0.28	0.21	0.22	0.21	0.26	0.24	0.24	0.19	0.17
	avg. true gap (%)	0.013	0.019	0.026	0.0090	0.00045	0.00032	0.00050	0.00061	0.000077	0.012	0.0049	0.0062
	solved to opt. (%)	20	40	20	90	90	97	93	97	97	59	76	80
FPTAS-gen(0.1)	avg. sol. time (s)	3.3	3.6	3.5	3.7	3.7	3.7	3.6	3.8	3.9	4.0	3.6	3.4
	avg. post gap (%)	6.8	6.8	6.8	6.6	6.6	6.5	6.5	6.5	6.5	6.6	6.6	6.6
	avg. true gap (%)	0.019	0.019	0.025	0.012	0.0022	0.0011	0.0017	0.00065	0.0029	0.013	0.0076	0.0069
	solved to opt. (%)	13	37	13	80	83	93	90	93	80	52	70	72
FPTAS-gen(0.05)	avg. sol. time (s)	10	11	11	10	11	10	10	11	11	11	10	9.7
	avg. post gap (%)	3.4	3.4	3.39	3.3	3.3	3.29	3.27	3.27	3.25	3.3	3.3	3.3
	avg. true gap (%)	0.0081	0.012	0.012	0.0013	0.00065	0.00032	0.00017	0.00065	0.00075	0.0062	0.0035	0.0022
	solved to opt. (%)	23	43	33	90	87	97	93	93	93	62	77	79
FPTAS-gen(0.01)	avg. sol. time (s)	172	196	184	160	163	159	140	154	158	180	162	153
	avg. post gap (%)	0.68	0.68	0.68	0.66	0.66	0.66	0.66	0.66	0.65	0.66	0.67	0.67
	avg. true gap (%)	0.0025	0.00097	0.0026	0.00013	0.00045	0.00032	0.00015	0.00061	0.00024	0.0013	0.00084	0.00055
	solved to opt. (%)	37	87	57	97	90	97	97	97	93	77	87	87
CPLEX 10.1 SP	avg. sol. time (s)	0.24	0.22	0.22	0.32	0.28	0.30	0.25	0.27	0.28	0.33	0.24	0.22

Table 7: 100 periods with 50 pairs that violate the co-behaviour property

	T	26		50		100	
	$\bar{K}_{even}$ and $\bar{K}_{odd}$	5000	10000	5000	10000	5000	10000
Megiddo	avg. sol. time (s)	0.002	0.002	0.002	0.006	0.013	0.018
	avg. post. gap (%)	11	13	5.8	6.6	2.6	3.1
	avg. true gap (%)	5.3	6.8	3.9	3.7	1.9	2.3
	solved to opt. (%)	37	47	17	27	23	37
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.018	0.016	0.10	0.12	0.68	0.73
	avg. post. gap (%)	3.8	3.6	1.9	2.7	0.64	0.74
	avg. true gap (%)	0.038	0.0060	0.032	0.030	0.014	0.0018
	solved to opt. (%)	80	97	73	93	67	97
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.047	0.035	0.28	0.31	2.0	2.1
	avg. post. gap (%)	2.3	2.3	1.7	2.0	0.64	0.74
	avg. true gap (%)	0.049	0.0060	0.048	0.030	0.014	0.0018
	solved to opt. (%)	80	97	73	93	67	97
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.71	0.44	5.2	5.8	35	37
	avg. post. gap (%)	0.51	0.57	0.53	0.54	0.45	0.46
	avg. true gap (%)	0.0013	0	0.021	0	0.0047	0.0018
	solved to opt. (%)	93	100	80	100	80	97
FPTAS-gen(0.1)	avg. sol. time (s)	0.17	0.11	1.3	1.2	12	11
	avg. post. gap (%)	6.3	6.3	6.4	6.3	6.4	6.4
	avg. true gap (%)	0.027	0.0014	0.028	0.0011	0.011	0.0090
	solved to opt. (%)	83	97	73	97	70	90
FPTAS-gen(0.05)	avg. sol. time (s)	0.52	0.32	4.3	4.0	37	34
	avg. post. gap (%)	3.2	3.2	3.2	3.2	3.2	3.2
	avg. true gap (%)	0.032	0.0060	0.028	0.0011	0.0077	0
	solved to opt. (%)	80	97	73	97	73	100
FPTAS-gen(0.01)	avg. sol. time (s)	11	5.9	94	87	726	656
	avg. post. gap (%)	0.65	0.65	0.65	0.64	0.65	0.64
	avg. true gap (%)	0.0015	0	0.0020	0	0.0019	0
	solved to opt. (%)	90	100	90	100	83	100
CPLEX 10.1 Nat.	avg. sol. time (s)	0.037	0.032	0.11	0.13		
CPLEX 10.1 SP	avg. sol. time (s)	0.065	0.042	0.13	0.14	0.55	0.56

Table 8: Two production modes

B.2 Results without improved lower bound

Tables 9–15 present the results of the computational tests of the algorithms that do not use the improved lower bound, as described in Sections 5.3 and 5.4.

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-CB-LB(0.1)	avg. sol. time (s)	0.001	0.001	0.002	0.001	<0.001	<0.001	0.001	0.001	0.001
	avg. post gap (%)	0.62	0.49	0.58	1.3	1.0	1.3	1.5	1.1	1.2
	avg. true gap (%)	0.041	0.014	0.032	0.00012	0.0017	0	0.011	0.00012	0.0023
	solved to opt. (%)	77	87	83	97	93	100	83	97	97
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.002	0.002	0.001	0.002	0.002	0.003	0.001	0.002	0.003
	avg. post gap (%)	0.58	0.48	0.54	1.1	0.97	1.1	1.2	1.0	1.1
	avg. true gap (%)	0.0036	0.0062	0.00068	0	0.0031	0	0.0034	0.00012	0.0023
	solved to opt. (%)	97	90	97	100	87	100	93	97	97
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.006	0.003	0.005	0.004	0.007	0.004	0.006	0.008	0.006
	avg. post gap (%)	0.30	0.24	0.29	0.38	0.38	0.37	0.44	0.38	0.41
	avg. true gap (%)	0	0.00077	0	0	0.00073	0	0.00048	0.00012	0
	solved to opt. (%)	100	97	100	100	97	100	97	97	100
FPTAS-CB(0.1)	avg. sol. time (s)	0.007	0.008	0.008	0.009	0.007	0.009	0.009	0.011	0.008
	avg. post. gap (%)	5.0	5.0	5.1	5.5	5.4	5.5	5.5	5.6	5.6
	avg. true gap (%)	0.034	0.026	0.015	0.00012	0.0024	0.0049	0.0089	0.00012	0.0074
	solved to opt. (%)	83	87	83	97	90	93	87	97	93
FPTAS-CB(0.05)	avg. sol. time (s)	0.015	0.015	0.016	0.018	0.019	0.018	0.018	0.017	0.019
	avg. post. gap (%)	2.5	2.5	2.5	2.7	2.7	2.7	2.7	2.8	2.8
	avg. true gap (%)	0.0063	0.0095	0.0091	0	0.0017	0	0.0091	0.00012	0.0023
	solved to opt. (%)	90	90	90	100	93	100	87	97	97
FPTAS-CB(0.01)	avg. sol. time (s)	0.087	0.082	0.083	0.098	0.092	0.092	0.094	0.091	0.091
	avg. post. gap (%)	0.49	0.5	0.5	0.53	0.53	0.53	0.55	0.55	0.54
	avg. true gap (%)	0	0.0015	0	0.00012	0.00073	0	0.00048	0.00012	0
	solved to opt. (%)	100	93	100	97	97	100	97	97	100
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.002	0.002	0.002	0.002	<0.001	0.002	0.002	0.002	0.001
	avg. post gap (%)	0.58	0.47	0.56	1.4	1.0	1.4	1.6	1.1	1.2
	avg. true gap (%)	0.0029	0.0015	0.015	0	0.0024	0	0.0060	0.00012	0.0023
	solved to opt. (%)	97	93	87	100	90	100	90	97	97
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.003	0.002	0.002	0.002	0.004	0.005	0.005	0.005	0.003
	avg. post gap (%)	0.58	0.47	0.54	1.3	1.0	1.2	1.3	1.1	1.2
	avg. true gap (%)	0	0.00077	0.00068	0	0.0016	0	0	0.00012	0
	solved to opt. (%)	100	97	97	100	93	100	100	97	100
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.016	0.012	0.015	0.016	0.015	0.016	0.018	0.018	0.020
	avg. post gap (%)	0.43	0.34	0.37	0.51	0.54	0.53	0.62	0.51	0.55
	avg. true gap (%)	0	0	0	0	0	0	0.00048	0.00012	0
	solved to opt. (%)	100	100	100	100	100	100	97	97	100
FPTAS-gen(0.1)	avg. sol. time (s)	0.018	0.020	0.019	0.021	0.021	0.020	0.022	0.021	0.021
	avg. post gap (%)	7.4	7.4	7.5	7.6	7.6	7.7	7.7	7.7	7.7
	avg. true gap (%)	0.0062	0.0015	0.012	0.00012	0.0024	0	0.0055	0.00046	0.0023
	solved to opt. (%)	90	93	87	97	90	100	90	93	97
FPTAS-gen(0.05)	avg. sol. time (s)	0.043	0.039	0.042	0.044	0.041	0.043	0.047	0.048	0.045
	avg. post gap (%)	3.7	3.7	3.7	3.8	3.8	3.8	3.9	3.8	3.9
	avg. true gap (%)	0.0029	0.00077	0.0013	0.00012	0.0024	0	0.00048	0.00046	0
	solved to opt. (%)	97	97	93	97	90	100	97	93	100
FPTAS-gen(0.01)	avg. sol. time (s)	0.25	0.25	0.24	0.28	0.27	0.27	0.28	0.27	0.28
	avg. post gap (%)	0.74	0.75	0.75	0.77	0.76	0.77	0.77	0.77	0.77
	avg. true gap (%)	0	0	0.00067	0	0	0	0	0.00012	0
	solved to opt. (%)	100	100	97	100	100	100	100	97	100

Table 9: 25 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-CB-LB(0.1)	avg. sol. time (s)	0.002	0.002	0.004	0.007	0.005	0.005	0.008	0.007	0.004
	avg. post gap (%)	0.24	0.28	0.25	0.46	0.53	0.55	0.56	0.61	0.64
	avg. true gap (%)	0.023	0.034	0.054	0.016	0.0080	0.0050	0.0074	0	0.020
	solved to opt. (%)	67	60	37	87	87	93	93	100	87
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.009	0.009	0.007	0.008	0.008	0.009	0.010	0.011	0.011
	avg. post gap (%)	0.23	0.25	0.21	0.45	0.52	0.54	0.56	0.61	0.63
	avg. true gap (%)	0.0081	0.0038	0.010	0.00062	0.0048	0	0.00033	0	0.0030
	solved to opt. (%)	80	87	70	97	93	100	97	100	97
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.034	0.033	0.039	0.041	0.045	0.041	0.052	0.052	0.051
	avg. post gap (%)	0.19	0.22	0.17	0.29	0.35	0.32	0.37	0.39	0.37
	avg. true gap (%)	0.00035	0.00027	0	0	0.00061	0	0	0	0
	solved to opt. (%)	93	97	100	100	97	100	100	100	100
FPTAS-CB(0.1)	avg. sol. time (s)	0.044	0.044	0.045	0.051	0.051	0.053	0.053	0.056	0.056
	avg. post. gap (%)	5.1	5.2	5.2	5.5	5.5	5.5	5.6	5.7	5.6
	avg. true gap (%)	0.028	0.035	0.047	0.0062	0.0079	0.0078	0	0	0.0047
	solved to opt. (%)	53	50	33	93	90	90	100	100	97
FPTAS-CB(0.05)	avg. sol. time (s)	0.091	0.089	0.089	0.10	0.11	0.11	0.12	0.11	0.11
	avg. post. gap (%)	2.5	2.5	2.5	2.7	2.7	2.7	2.8	2.8	2.8
	avg. true gap (%)	0.015	0.0086	0.014	0.0013	0.0032	0.00015	0	0	0
	solved to opt. (%)	70	83	73	97	93	97	100	100	100
FPTAS-CB(0.01)	avg. sol. time (s)	0.56	0.57	0.56	0.68	0.69	0.69	0.72	0.72	0.73
	avg. post. gap (%)	0.50	0.50	0.50	0.54	0.53	0.54	0.55	0.55	0.55
	avg. true gap (%)	0.00020	0.0080	0	0	0	0	0	0	0
	solved to opt. (%)	97	97	100	100	100	100	100	100	100
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.010	0.009	0.009	0.014	0.010	0.011	0.015	0.013	0.012
	avg. post gap (%)	0.23	0.25	0.21	0.45	0.52	0.55	0.56	0.61	0.63
	avg. true gap (%)	0.0072	0.0051	0.017	0.0041	0.0026	0.0026	0	0	0.0018
	solved to opt. (%)	80	83	63	97	97	97	100	100	97
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.017	0.019	0.018	0.019	0.022	0.020	0.026	0.026	0.027
	avg. post gap (%)	0.22	0.25	0.20	0.45	0.52	0.54	0.56	0.61	0.62
	avg. true gap (%)	0.00091	0.0019	0.0057	0	0	0	0	0	0
	solved to opt. (%)	90	90	80	100	100	100	100	100	100
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.095	0.093	0.098	0.11	0.12	0.12	0.14	0.15	0.14
	avg. post gap (%)	0.21	0.24	0.19	0.35	0.45	0.40	0.46	0.46	0.46
	avg. true gap (%)	0	0	0	0	0	0	0	0	0
	solved to opt. (%)	100	100	100	100	100	100	100	100	100
FPTAS-gen(0.1)	avg. sol. time (s)	0.11	0.12	0.12	0.13	0.13	0.14	0.14	0.14	0.14
	avg. post gap (%)	7.5	7.5	7.5	7.7	7.7	7.7	7.7	7.7	7.7
	avg. true gap (%)	0.0047	0.015	0.014	0.0013	0	0.0024	0	0	0
	solved to opt. (%)	83	80	63	97	100	93	100	100	100
FPTAS-gen(0.05)	avg. sol. time (s)	0.25	0.25	0.25	0.28	0.29	0.30	0.31	0.30	0.30
	avg. post gap (%)	3.7	3.7	3.7	3.8	3.8	3.8	3.9	3.9	3.9
	avg. true gap (%)	0.00074	0.0019	0.0049	0	0	0	0	0	0
	solved to opt. (%)	93	87	83	100	100	100	100	100	100
FPTAS-gen(0.01)	avg. sol. time (s)	1.6	1.6	1.6	1.8	1.9	1.9	2.0	2.0	2.0
	avg. post gap (%)	0.75	0.75	0.75	0.77	0.77	0.77	0.77	0.78	0.77
	avg. true gap (%)	0.00026	0.00031	0	0	0	0	0	0	0
	solved to opt. (%)	97	93	100	100	100	100	100	100	100

Table 10: 50 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-CB-LB(0.1)	avg. sol. time (s)	0.025	0.028	0.025	0.031	0.031	0.033	0.038	0.038	0.036
	avg. post gap (%)	0.078	0.078	0.094	0.17	0.20	0.18	0.25	0.24	0.20
	avg. true gap (%)	0.017	0.019	0.020	0.012	0.013	0.0092	0.010	0.0042	0.0049
	solved to opt. (%)	60	47	43	57	67	80	80	90	83
FPTAS-CB-LB(0.05)	avg. sol. time (s)	0.047	0.047	0.047	0.062	0.061	0.059	0.071	0.074	0.070
	avg. post gap (%)	0.064	0.069	0.081	0.16	0.19	0.18	0.24	0.24	0.20
	avg. true gap (%)	0.0027	0.0098	0.0072	0.0046	0.0033	0.00023	0.0039	0	0.0025
	solved to opt. (%)	87	60	57	77	83	97	87	100	83
FPTAS-CB-LB(0.01)	avg. sol. time (s)	0.27	0.26	0.27	0.36	0.35	0.37	0.44	0.45	0.44
	avg. post gap (%)	0.061	0.059	0.074	0.16	0.19	0.17	0.23	0.22	0.20
	avg. true gap (%)	0	0	0.00032	0.00035	0.00064	0	0	0 0	0.00055
	solved to opt. (%)	100	100	87	97	93	100	100	100	90
FPTAS-CB(0.1)	avg. sol. time (s)	0.28	0.27	0.28	0.34	0.34	0.35	0.38	0.38	0.38
	avg. post. gap (%)	5.2	5.2	5.1	5.5	5.6	5.6	5.6	5.6	5.6
	avg. true gap (%)	0.027	0.043	0.041	0.0050	0.026	0.0081	0.0020	0.0064	0.0014
	solved to opt. (%)	50	40	43	77	50	87	93	87	87
FPTAS-CB(0.05)	avg. sol. time (s)	0.60	0.58	0.60	0.76	0.74	0.76	0.85	0.86	0.85
	avg. post. gap (%)	2.5	2.5	2.5	2.7	2.7	2.7	2.8	2.8	2.8
	avg. true gap (%)	0.0071	0.0097	0.011	0.0023	0.0016	0.0024	0.033	0	0.00055
	solved to opt. (%)	80	73	60	87	83	87	87	100	90
FPTAS-CB(0.01)	avg. sol. time (s)	4.1	3.9	4.1	5.2	5.1	5.2	5.8	5.8	5.8
	avg. post. gap (%)	0.50	0.50	0.50	0.54	0.54	0.54	0.55	0.55	0.55
	avg. true gap (%)	0	0.0015	0.00058	0	0.000092	0.00037	0.00035	0	0.000037
	solved to opt. (%)	100	90	83	100	97	93	97	100	97
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.060	0.059	0.059	0.075	0.071	0.072	0.086	0.090	0.083
	avg. post gap (%)	0.071	0.068	0.082	0.16	0.19	0.18	0.24	0.24	0.20
	avg. true gap (%)	0.0097	0.0092	0.0087	0.0018	0.0011	0.00037	0.00041	0.0033	0.0012
	solved to opt. (%)	63	67	53	87	87	93	97	93	87
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.12	0.11	0.12	0.15	0.14	0.15	0.17	0.17	0.17
	avg. post gap (%)	0.063	0.060	0.076	0.16	0.19	0.18	0.24	0.24	0.20
	avg. true gap (%)	0.0014	0.0011	0.0030	0.00067	0.0012	0.00037	0.00036	0	0.00055
	solved to opt. (%)	90	93	73	93	87	93	97	100	90
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.80	0.76	0.79	1.0	1.0	1.1	1.3	1.3	1.3
	avg. post gap (%)	0.061	0.059	0.074	0.16	0.19	0.18	0.24	0.23	0.20
	avg. true gap (%)	0	0	0.00012	0	0.000092	0	0	0	0.000037
	solved to opt. (%)	100	100	90	100	97	100	100	100	97
FPTAS-gen(0.1)	avg. sol. time (s)	0.75	0.74	0.76	0.93	0.92	0.94	1.0	1.0	1.0
	avg. post gap (%)	7.5	7.5	7.5	7.7	7.7	7.7	7.7	7.7	7.7
	avg. true gap (%)	0.011	0.0070	0.013	0.0021	0.0012	0.0015	0.0040	0	0.00055
	solved to opt. (%)	63	77	60	87	87	93	90	100	90
FPTAS-gen(0.05)	avg. sol. time (s)	1.7	1.6	1.7	2.1	2.0	2.1	2.3	2.3	2.3
	avg. post gap (%)	3.7	3.7	3.7	3.8	3.8	3.8	3.9	3.9	3.9
	avg. true gap (%)	0.0015	0.0022	0.0024	0.00035	0.0021	0.00044	0.00058	0	0.00025
	solved to opt. (%)	87	87	83	97	83	97	97	100	93
FPTAS-gen(0.01)	avg. sol. time (s)	11	10	11	13	13	13	15	15	15
	avg. post gap (%)	0.75	0.75	0.75	0.77	0.77	0.77	0.77	0.77	0.77
	avg. true gap (%)	0	0	0.00012	0	0	0.00014	0	0	0.000037
	solved to opt. (%)	100	100	93	100	100	97	100	100	97

Table 11: 100 periods, satisfies conditions in Theorem 3

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.004	0.005	0.003	0.004	0.001	0.005	0.004	0.003	0.003
	avg. post gap (%)	1.1	0.99	1.1	1.4	2.0	1.7	2.5	2.1	1.8
	avg. true gap (%)	0.20	0.058	0.19	0.00044	0.0017	0.021	0	0.0060	0.0083
	solved to opt. (%)	43	37	40	97	97	83	100	97	87
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.007	0.006	0.007	0.007	0.007	0.010	0.009	0.010	0.007
	avg. post gap (%)	1.0	1.0	1.0	1.4	1.7	1.6	2.0	1.9	1.6
	avg. true gap (%)	0.10	0.068	0.16	0.00044	0	0.0090	0	0	0.0010
	solved to opt. (%)	50	43	40	97	100	87	100	100	97
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.090	0.067	0.079	0.063	0.075	0.086	0.083	0.071	0.071
	avg. post gap (%)	0.43	0.58	0.49	0.54	0.61	0.61	0.58	0.66	0.51
	avg. true gap (%)	0.014	0.033	0.030	0	0	0.011	0	0	0
	solved to opt. (%)	73	57	67	100	100	90	100	100	100
FPTAS-gen(0.1)	avg. sol. time (s)	0.063	0.053	0.059	0.047	0.053	0.053	0.047	0.044	0.046
	avg. post gap (%)	7.4	7.5	7.4	7.6	7.7	7.6	7.7	7.7	7.7
	avg. true gap (%)	0.031	0.051	0.043	0.00044	0.0051	0.0091	0	0.0060	0.0033
	solved to opt. (%)	60	43	63	97	93	83	100	97	90
FPTAS-gen(0.05)	avg. sol. time (s)	0.18	0.15	0.16	0.12	0.13	0.14	0.12	0.11	0.12
	avg. post gap (%)	3.7	3.7	3.7	3.8	3.8	3.8	3.9	3.9	3.9
	avg. true gap (%)	0.025	0.022	0.014	0.00044	0	0.010	0	0	0
	solved to opt. (%)	73	63	70	97	100	80	100	100	100
FPTAS-gen(0.01)	avg. sol. time (s)	3.2	2.4	2.8	1.7	2.0	2.1	1.7	1.5	1.7
	avg. post gap (%)	0.73	0.74	0.73	0.76	0.77	0.77	0.77	0.77	0.77
	avg. true gap (%)	0.00081	0.0028	0.0057	0	0	0.0012	0	0	0
	solved to opt. (%)	93	83	80	100	100	97	100	100	100

Table 12: 25 periods with 13 pairs that violate the co-behaviour property

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.025	0.024	0.024	0.026	0.025	0.024	0.028	0.031	0.029
	avg. post gap (%)	0.44	0.38	0.37	0.57	0.60	0.67	0.66	0.83	1.0
	avg. true gap (%)	0.085	0.049	0.042	0.012	0.013	0.0029	0.0046	0.0014	0
	solved to opt. (%)	23	40	47	87	77	83	93	97	100
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.054	0.056	0.052	0.054	0.054	0.055	0.062	0.062	0.065
	avg. post gap (%)	0.44	0.39	0.36	0.57	0.59	0.67	0.66	0.83	1.0
	avg. true gap (%)	0.085	0.051	0.039	0.0093	0.0054	0.0041	0	0.00078	0
	solved to opt. (%)	23	40	50	90	90	80	100	97	100
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.62	0.63	0.60	0.59	0.59	0.60	0.71	0.69	0.76
	avg. post gap (%)	0.35	0.34	0.32	0.48	0.42	0.47	0.51	0.54	0.59
	avg. true gap (%)	0.033	0.022	0.019	0.0067	0.0054	0.0022	0	0	0
	solved to opt. (%)	47	57	60	93	90	93	100	100	100
FPTAS-gen(0.1)	avg. sol. time (s)	0.43	0.44	0.42	0.40	0.38	0.41	0.39	0.39	0.38
	avg. post gap (%)	7.5	7.5	7.5	7.7	7.7	7.7	7.7	7.7	7.7
	avg. true gap (%)	0.044	0.020	0.016	0.0027	0.0053	0.0029	0.0018	0	0.0054
	solved to opt. (%)	37	53	50	93	87	83	97	100	93
FPTAS-gen(0.05)	avg. sol. time (s)	1.3	1.3	1.2	1.1	1.1	1.2	1.1	1.1	1.0
	avg. post gap (%)	3.7	3.7	3.7	3.8	3.8	3.8	3.9	3.9	3.9
	avg. true gap (%)	0.0058	0.013	0.0097	0.0067	0.0054	0.0041	0	0.00078	0
	solved to opt. (%)	67	57	67	93	90	80	100	97	100
FPTAS-gen(0.01)	avg. sol. time (s)	25	25	24	18	17	19	17	16	15
	avg. post gap (%)	0.75	0.75	0.75	0.77	0.77	0.77	0.77	0.77	0.77
	avg. true gap (%)	0.0020	0.0015	0.0011	0.00040	0	0.00066	0	0	0
	solved to opt. (%)	77	87	83	97	100	97	100	100	100

Table 13: 50 periods with 25 pairs that violate the co-behaviour property

	K	1000			5000			10000		
	$\hat{K}$	1000	5000	10000	1000	5000	10000	1000	5000	10000
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.16	0.16	0.15	0.18	0.18	0.18	0.19	0.20	0.21
	avg. post gap (%)	0.13	0.14	0.16	0.28	0.23	0.22	0.21	0.28	0.24
	avg. true gap (%)	0.027	0.027	0.054	0.011	0.015	0.0029	0.0012	0.00065	0.0024
	solved to opt. (%)	17	30	7	80	80	93	90	93	83
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.37	0.39	0.36	0.44	0.46	0.43	0.48	0.50	0.52
	avg. post gap (%)	0.13	0.14	0.15	0.28	0.23	0.22	0.21	0.28	0.24
	avg. true gap (%)	0.022	0.024	0.042	0.0093	0.015	0.00083	0.0012	0.0011	0.00024
	solved to opt. (%)	13	37	17	83	87	93	87	90	93
FPTAS-gen-LB(0.01)	avg. sol. time (s)	4.6	5.0	4.6	5.3	5.5	5.2	5.5	6.0	6.1
	avg. post gap (%)	0.12	0.13	0.13	0.28	0.21	0.22	0.21	0.27	0.24
	avg. true gap (%)	0.013	0.019	0.024	0.0092	0.00045	0.00032	0.00017	0.00061	0.000077
	solved to opt. (%)	17	40	20	87	90	97	93	97	97
FPTAS-gen(0.1)	avg. sol. time (s)	3.2	3.5	3.4	3.6	3.6	3.5	3.5	3.7	3.8
	avg. post gap (%)	7.5	7.5	7.5	7.7	7.7	7.7	7.7	7.7	7.7
	avg. true gap (%)	0.012	0.025	0.025	0.00091	0.0020	0.00083	0.0036	0.00061	0.0035
	solved to opt. (%)	10	30	17	87	87	93	80	97	87
FPTAS-gen(0.05)	avg. sol. time (s)	9.7	11	10	10	10	10	9.6	10	11
	avg. post gap (%)	3.8	3.8	3.8	3.8	3.8	3.8	3.9	3.9	3.9
	avg. true gap (%)	0.0099	0.012	0.0093	0.00088	0.00045	0.00032	0.00052	0.00061	0.00024
	solved to opt. (%)	20	40	37	90	90	97	90	97	93
FPTAS-gen(0.01)	avg. sol. time (s)	169	193	181	157	160	156	137	151	155
	avg. post gap (%)	0.75	0.75	0.75	0.77	0.77	0.77	0.77	0.78	0.77
	avg. true gap (%)	0.0025	0.00016	0.0024	0.00017	0.00045	0.00032	0.00017	0.00061	0.000077
	solved to opt. (%)	40	93	57	93	90	97	93	97	97

Table 14: 100 periods with 50 pairs that violate the co-behaviour property

	T	26		50		100	
	$\bar{K}_{even}$ and $\bar{K}_{odd}$	5000	10000	5000	10000	5000	10000
FPTAS-gen-LB(0.1)	avg. sol. time (s)	0.018	0.010	0.092	0.098	0.59	0.63
	avg. post. gap (%)	4.3	4.1	1.9	2.8	0.65	0.74
	avg. true gap (%)	0.038	0.0060	0.028	0.030	0.016	0.0018
	solved to opt. (%)	80	97	73	93	67	97
FPTAS-gen-LB(0.05)	avg. sol. time (s)	0.042	0.030	0.25	0.28	1.8	1.9
	avg. post. gap (%)	2.8	2.7	1.8	2.4	0.64	0.74
	avg. true gap (%)	0.038	0.0060	0.044	0.029	0.014	0.0018
	solved to opt. (%)	80	97	73	97	70	97
FPTAS-gen-LB(0.01)	avg. sol. time (s)	0.66	0.41	5.0	5.5	33	35
	avg. post. gap (%)	0.66	0.73	0.71	0.74	0.54	0.57
	avg. true gap (%)	0.0013	0	0.021	0	0.0047	0.0018
	solved to opt. (%)	93	100	80	100	80	97
FPTAS-gen(0.1)	avg. sol. time (s)	0.16	0.11	1.2	1.1	11	10
	avg. post. gap (%)	7.6	7.6	7.7	7.7	7.8	7.8
	avg. true gap (%)	0.038	0.0060	0.050	0	0.014	0.0029
	solved to opt. (%)	83	97	70	100	70	93
FPTAS-gen(0.05)	avg. sol. time (s)	0.50	0.31	4.1	3.9	36	33
	avg. post. gap (%)	3.8	3.8	3.8	3.8	3.9	3.9
	avg. true gap (%)	0.032	0.0060	0.028	0	0.0077	0
	solved to opt. (%)	80	97	77	100	73	100
FPTAS-gen(0.01)	avg. sol. time (s)	11	5.7	91	84	705	639
	avg. post. gap (%)	0.77	0.77	0.77	0.77	0.77	0.77
	avg. true gap (%)	0.0013	0	0.0020	0	0.0017	0
	solved to opt. (%)	93	100	90	100	87	100

Table 15: Two production modes