



HAL
open science

A perfect information lower bound for robust lot-sizing problems

Marcio C Santos, Michael Poss, Dritan C Nace

► **To cite this version:**

Marcio C Santos, Michael Poss, Dritan C Nace. A perfect information lower bound for robust lot-sizing problems. 2016. hal-01409515v1

HAL Id: hal-01409515

<https://hal.science/hal-01409515v1>

Preprint submitted on 5 Dec 2016 (v1), last revised 31 Jan 2018 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A perfect information lower bound for robust lot-sizing problems[☆]

Marcio C. Santos^{a,*}, Michael Poss^{b,**}, Dritan Nace^a

^aUMR CNRS 7253 Heudiasyc, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, 60200 Compiègne, France

^bUMR CNRS 5506 LIRMM, Université de Montpellier 2, 161 rue Ada, 34392 Montpellier Cedex 5, France.

Abstract

Robust multi-stage optimization is hard computationally and only small problems can be solved exactly. These problems are typically addressed heuristically through decision rules, which provide upper bounds for the optimal solution costs of the problems. We investigate in this paper lower bounds inspired by the perfect information model used in stochastic programming. Specifically, we study the uncapacitated robust lot-sizing problem, showing that different versions of the problem become tractable whenever the non-anticipativity constraints are relaxed. Hence, we can solve the resulting problem efficiently, obtaining a lower bound for the optimal solution cost of the original problem. We compare numerically the solution time and the quality of the new lower bound with the dual affine decision rules that have been proposed by Kuhn et al. (2011).

Keywords: multi-stage robust optimization, perfect information, lot-sizing problem

1. Introduction

Lot-sizing optimization problems appear in a wide range of applications where products have to be made to attend demands along with a planning horizon. In these problems, the future demands are usually not known with precision until the current period is reached. To be relevant in practice, optimization models must model the uncertainty explicitly so that the decisions are taken so as to optimize the benefit over the whole planning horizon. Different models of uncertainty exist in the literature, each of which having its advantages and drawbacks. In this paper, we focus on robust lot-sizing, therefore assuming that the demand uncertainty is modeled by a convex set. Thus, the objective is to optimize the cost of the production plan in the worst-case scenario. This model, used in [2, 15, 9, 14, 25, 6, 23], among others, is relevant when historical data are not accurate enough to draw probabilistic distributions of the uncertain demands.

Robust optimization is often known as being a natural approach to uncertainty since, for instance, a robust linear program with polyhedral uncertainty sets can be reformulated as a robust linear program whose dimension does not grow much with respect to the dimensions of the deterministic problem [8]. While this is true for *static problems*, where decisions are taken before revealing the uncertain parameters, the situations with *adjustable problems* is far more complex. *Adjustable robust optimization* problems suppose that the uncertainty is revealed as time goes by and one can adjust the values of some of the decision variables according to the

[☆]This work was supported by the French ministries MAE and MENESR

*Corresponding author

**Principal corresponding author

Email addresses: marcio.costa-santos@hds.utc.fr (Marcio C. Santos), michael.poss@lirmm.fr (Michael Poss), dritan.nace@hds.utc.fr (Dritan Nace)

Preprint submitted to Elsevier

December 5, 2016

current knowledge of the uncertain parameters. Hence, the adjustable optimization variables become functions of the uncertain parameters. Robust lot-sizing problems can be modeled either as static or adjustable robust optimization problems. In the static version, it is assumed that the production plan is fixed before the planning horizon starts. In contrast, the adjustable version supposes that the production and other decisions can be adjusted according to the values taken by past demands, thus becoming *functions* of the past demands. In this paper, we consider the adjustable situation.

Adjustable robust optimization is known for being \mathcal{NP} -hard, even in the case of a linear program with only two decision stages [6]. In spite of its theoretical difficulty, the problem can be solved exactly by decomposition approaches whenever some assumptions hold [4, 16, 29]. These approaches consider finite subsets of the uncertainty set and dynamically increase the number of considered scenarios by solving separation problems. The numerical tractability of the resulting algorithms highly depends on the complexity of the separation problem. For instance, the separation problem for the robust vehicle routing problem can be solved in polynomial time [1], while the one related to facility location or network design problems require solving MILP with big-M coefficients [4, 16, 29]. These decomposition approaches do not extend to multi-stage problems, because of the non-anticipativity constraints present in these problems. Stated simply, non-anticipativity constraints model the fact that optimization variables can only depend on past realizations of the uncertain parameters; they cannot adjust their decision to unknown realizations.

Given the difficulty of adjustable multi-stage robust problems, many researchers have developed heuristic approaches that try to provide feasible solutions for these problems. The bottom line of all these approaches is to restrict the set of feasible functions for the adjustable variables. The seminal paper in this line of research is [6] which restricts adjustable variables to affine functions of the uncertainties, which they call *affine decision rules*. Subsequent authors have studied more complex decision rules that offer more flexibility than affine decision rules while providing more or less tractable optimization problems. Among others, [20] propose to define affine decision rules built from extended descriptions of the uncertainty set and [22] introduce complex piece-wise linear decision rules defined through the lifting of the uncertainty set. More complex decision rules have also been considered, such as [10] which proposes piece-wise decision rules modeled with the help of binary optimization variables, or [9, 28] which dynamically partition the uncertainty set and selects constant policies for each element of the partition.

The heuristic solutions typically yield upper bounds for minimization problems. Since the optimal solution of the underlying optimization problem is unknown, one needs lower bounds to evaluate the quality of the aforementioned upper bounds. Up to our knowledge, the literature is scarce when it comes to proposing lower bounds. Specifically, we are aware of only two previous works. First, authors have considered the simple lower bound that consists of selecting a finite subset of the uncertainty set and solving exactly the resulting finite linear program. This approach is used by [9], among others, in the course of their partitioning algorithm. Second, [25] have introduced dual affine decision rules, which provide lower bounds for multistage problems. While that work targets more specifically multistage stochastic optimization, their approach is also applicable to robust optimization.

The contributions of this paper follow that line of research by providing another way to compute lower bounds for multistage robust optimization problems. Our approach relaxes the non-anticipativity constraints of the problem, thus yielding a relaxation of the original problem. This relaxation is well-known in the stochastic programming literature as the *perfect information* relaxation. The first mention of the associated optimization problem in the stochastic programming literature, called the expected value of the perfect information, can be traced back to [3]. It has then been further studied in several papers e.g. [19] and has become a well-known concept in stochastic programming [17]. However, up to our knowledge it has never been used

in robust optimization. The interest of studying the perfect information relaxation in robust optimization is two-fold. On the one hand, it helps the decision maker in assessing how much gain could be obtained by reducing the uncertainty on the uncertain parameters (e.g. [26]). On the other hand, it can be used as a lower bound for the optimal value of the true uncertain problem. We argue that this second aspect is particularly relevant in robust optimization in which few papers have been proposing lower bounds. As we show in the paper, the perfect information relaxation of the robust lot-sizing problems can be solved efficiently either through polynomial-time algorithms or MILP reformulations. Our experiments realized on lot-sizing instances inspired by the literature seem to indicate that the perfect information relaxation can be very tight. Throughout the paper, we pay a particular attention to the budget uncertainty set introduced by [13] and widely used in the mixed-integer linear robust optimization literature.

The remaining sections of this paper are organized as follows. In Section 2, we formally present the problems we tackle and the uncertainty model used. In Section 3, we present the bounds known in the literature, as well as the problem obtained by relaxing the non-anticipativity constraints. In section 4, which provides the main methodological contributions of our paper, we present combinatorial algorithms and (integer or continuous) linear programming formulations to solve the relaxations based on the perfect information. In Section 5, we present numerical experiments to evaluate the quality of the bounds empirically. We conclude the paper in Section 6.

2. Problem description

2.1. Deterministic model

We describe below the problem studied in this paper. Let $H = \{1, \dots, n\}$ denote the planning horizon composed of n periods; c_i , h_i and p_i denote the *production*, *holding* and *backlogging* unitary cost at period i , respectively. The objective of the lot-sizing problem, denoted by \mathcal{LS} , is to provide a *production plan* (information about the amount produced, stored and backlogged in each period $i \in H$) that fulfills the *client demand* d_i at each period i , either by producing at that period or by producing in an earlier/later period, in which case we must pay a fee represented by the holding/backlogging cost. Also, we have to pay a fixed cost g_i in each period i where production takes place. To keep notations simple, we consider herein problems with a single item and a single producer; one can readily generalize our approach to problems with multiple items and producers. The mathematical formulation of problem \mathcal{LS} follows.

$$\begin{aligned}
 (\mathcal{LS}) \quad & \min \quad \kappa \\
 & s.t \quad \kappa \geq \sum_{i \in H} (c_i x_i + g_i y_i + h_i s_i + p_i r_i) & (1) \\
 & \quad s_{i+1} = x_i - d_i + s_i - r_{i-1} + r_i & \forall i \in H, & (2) \\
 & \quad x_i \leq M y_i & \forall i \in H, & (3) \\
 & \quad y \in \{0, 1\}^n, x, s, r \geq 0.
 \end{aligned}$$

Constraint (1) imposes that κ be not smaller than the cost of the production plan represented by x , which is formulated in the right-hand side of the constraint. Constraints (2) are equilibrium constraints linking the production, stock and backlog variables, respectively x_i , s_i and r_i . Constraints (3) state that if we produce in a period, then we must pay a fixed cost (setup cost g_i). The other constraints state that y (setup variables) is a binary vector while the other variables are non-negative, M is a large predefined value which is larger than the sum of all clients demands. Notice that the objective function could be substituted with the right-hand side of constraint (1) in the above formulation. However, we prefer to keep the formulation as

it is presented here because it can be translated more naturally to the robust context described in the next subsection.

2.2. Robust model

We assumed so far that demands are known with precision when solving problem \mathcal{LS} described earlier. This is unrealistic in many applications where one has to face uncertainty around the exact values of the demands. To model this issue, we consider an *uncertainty polytope* Ξ and we suppose that the demand at time period i is defined by the affine function $d_i(\xi)$ defined over Ξ :

$$d_i(\xi) = \bar{d}_i + \sum_{j \in H} \hat{D}_{ij} \xi_j, \quad (4)$$

where \bar{d}_i can be seen as the mean value of the clients demands for time period i among demands. We also assume that the demand functions are non-negative, more precisely $d_i(\xi) \geq 0$ for all $i \in H$ and $\xi \in \Xi$.

In the robust context, backlog and holding costs depend on the specific scenario ξ . Hence, they are represented by functions $s_i(\xi)$ and $r_i(\xi)$ for each time period i . The situation is more complex with production and setup costs. One could suppose that these features are independent of ξ , which would model the fact that all *decisions* must be taken at the beginning of the planning horizon, see for instance [2, 15, 14]. In this paper, we consider a more subtle approach where the productions and setups can be adjusted according to *past* demand realizations. Hence, these decisions are modeled by functions $x_i(\xi)$ and $y_i(\xi)$ for each time period i . Notice that, for each time period i , these functions must depend only on the demand revealed up to time period i . This is modeled by the *non-anticipativity* constraints

$$\begin{aligned} x_i(\xi) &= x_i(\xi') & \forall \xi, \xi' \in \Xi, \text{Proj}_{[1\dots i]}(\xi) &= \text{Proj}_{[1\dots i]}(\xi') \\ y_i(\xi) &= y_i(\xi') & \forall \xi, \xi' \in \Xi, \text{Proj}_{[1\dots i]}(\xi) &= \text{Proj}_{[1\dots i]}(\xi'). \end{aligned}$$

where $\text{Proj}_{[1\dots i]}(\xi)$ denotes the projection of ξ on its first i components. Said differently, the non-anticipativity constraints model the fact that x and y do not depend on future knowledge of the uncertainty. The mathematical formulation for the robust model follows.

$$\begin{aligned} (\mathcal{P}) \quad & \min & & \kappa \\ & s.t & & \kappa \geq \sum_{i \in H} (c_i x_i(\xi) + g_i y_i(\xi) + h_i s_i(\xi) + p_i r_i(\xi)) & \forall \xi \in \Xi & (5) \\ & & & s_{i+1}(\xi) = x_i(\xi) - d_i(\xi) + s_i(\xi) - r_{i-1}(\xi) + r_i(\xi) & \forall i \in H, \forall \xi \in \Xi & (6) \\ & & & x_i(\xi) \leq M y_i(\xi) & \forall i \in H, \forall \xi \in \Xi & (7) \\ & & & x_i(\xi) = x_i(\xi') & \forall \xi, \xi' \in \Xi, \text{Proj}_{[1\dots i]}(\xi) = \text{Proj}_{[1\dots i]}(\xi') & \forall i \in H, & (8) \\ & & & y_i(\xi) = y_i(\xi') & \forall \xi, \xi' \in \Xi, \text{Proj}_{[1\dots i]}(\xi) = \text{Proj}_{[1\dots i]}(\xi') & \forall i \in H, & (9) \\ & & & y(\xi) \in \{0, 1\}^n, x(\xi), s(\xi), r(\xi) \geq 0 & \forall \xi \in \Xi. \end{aligned}$$

Constraints (5)–(7) play the same role as constraints (1)–(3) for each element $\xi \in \Xi$. Constraints (8) and (9) are the non-anticipativity constraints mentioned previously. Notice that the above problem contains an infinite number of constraints and variables.

In this paper, we study bounding procedures for problem \mathcal{P} as well as for the following two simplifications of problem \mathcal{P} . The first one considers that the setup decisions must be taken before knowing anything about the demand; that is, y becomes a vector of optimization variables that are independent of ξ . The second one looks at the problem without setup costs,

which can be modeled by setting all components of y and g to 1 and 0, respectively. We denote these simplifications as \mathcal{P}_1 and \mathcal{P}_0 , respectively. Each of the three models is relevant for specific applications. For instance, \mathcal{P}_0 is close to the classical supply chain model addressed in most papers from the robust lot-sizing literature (e.g. [14, 6, 23]). In contrast, models \mathcal{P}_1 and \mathcal{P} are relevant for applications that involve fixed costs for the production due, for instance, to machine configurations.

In general, we assume that Ξ can be any non-empty polytope, described by the matrix W with m rows and $|H| = n$ columns

$$\Xi = \{\xi \mid W\xi \leq q\}. \quad (10)$$

In addition to general polytopes, we will also take a closer look at the complexity of the optimization problems when using the *budgeted uncertainty polytope* introduced in [13]. Given a positive real Γ , the budgeted polytope is defined as

$$\Xi_\Gamma = \left\{ \xi \mid \sum_{i \in H} |\xi_i| \leq \Gamma, \quad -1 \leq \xi_i \leq 1, \forall i \in H \right\}. \quad (11)$$

3. Bounds

We present in Subsections 3.1 and 3.2 approaches from the literature that provide upper and lower bounds for the optimal solution of the adjustable robust problems. Notice that these approaches cannot be applied to robust multi-stage optimization problems that contain adjustable integer variables, such as \mathcal{P} . Hence, in the following two sections, we assume that y does not depend on ξ , either because it is a nonadjustable vector of optimization variables (as in \mathcal{P}_0) or because each of its components has been fixed to 1 (as in \mathcal{P}_1).

3.1. Affine decision rules

The classical upper bound for multi-stage robust optimization problems is based on the so-called affine decision rules. The main idea of the approach is to impose that functions s_i, r_i and x_i depend affinely on ξ . Formally, these restrictions are modeled with constraints

$$x_i(\xi) = x_i^0 + \sum_{j=1}^i x_i^j \xi_j, \quad (12)$$

$$s_i(\xi) = s_i^0 + \sum_{j=1}^n s_i^j \xi_j, \quad (13)$$

$$r_i(\xi) = r_i^0 + \sum_{j=1}^n r_i^j \xi_j. \quad (14)$$

where x_i^0, s_i^0, r_i^0 and x_i^j, s_i^j, r_i^j for $i, j \in H$ are optimization variables. The right-hand side of (12) involves only the components of $\text{Proj}_{[1\dots i]}(\xi) = \text{Proj}_{[1\dots i]}(\xi)$. Hence, the equation models implicitly the non-anticipativity constraints introduced in the previous section. Substituting $x_i(\xi), s_i(\xi)$, and $r_i(\xi)$ with the rhs of (5)–(7) for each $i \in H$, we obtain an upper bound for \mathcal{P}_0 and \mathcal{P}_1 , see Appendix A. Then, one can apply classical tools from robust optimization to the formulation from Appendix A to reformulate the upper bound as a compact linear program.

3.2. Dual affine decision rules

Recently, the authors of [25] have proposed lower bounds for problems \mathcal{P}_0 and \mathcal{P}_1 , which they call *dual affine decision rules*. To be more precise, their approach is developed to provide lower bounds for multi-stage stochastic linear programs. To apply the technique to robust multi-stage programs, one needs to introduce artificial probability weights for the scenarios in Ξ . These probability weights are then used to formulate a lower bounding problem where the robust constraints are relaxed to expectation constraints. Then, a subtle reformulation allows them to provide a compact linear mixed integer formulation for the lower bounding problem. The reformulation is based on the use of convex duality and probability theory. One of the main difficulties of the method relies in the computation of the expectation matrix $\mathcal{M} = \mathbb{E}(\xi^T \xi)$. The approach is sketched in Appendix B, we redirect the interested reader to [25] for full details.

3.3. Perfect information relaxation

The major impediment to the efficient solutions of problems \mathcal{P} , \mathcal{P}_0 and \mathcal{P}_1 lies in the presence of the non-anticipativity constraints. Expressing non-anticipativity constraints is not easy in general and strongly depends on the particular structure of the considered set Ξ . In what follows, we propose a lower bounding problem for \mathcal{P} that relaxes the non-anticipativity constraints from \mathcal{P} , which we call the *problem with perfect information*. Unlike the affine decision rules and the dual version presented in the previous sections, the perfect information relaxation can be applied to \mathcal{P} regardless to the dependency of y on ξ . The associated optimization problems are denoted by \mathcal{PI} , \mathcal{PI}_1 , and \mathcal{PI}_0 , for \mathcal{P} , \mathcal{P}_1 and \mathcal{P}_0 respectively.

$$\begin{aligned}
 (\mathcal{PI}) \quad & \min \quad \kappa \\
 & s.t \quad \kappa \geq \sum_{i \in H} (c_i x_i(\xi) + g_i y_i(\xi) + h_i s_i(\xi) + p_i r_i(\xi)) \quad \forall \xi \in \Xi \\
 & \quad s_{i+1}(\xi) = x_i(\xi) - d_i(\xi) + s_i(\xi) - r_{i-1}(\xi) + r_i(\xi) \quad \forall i \in H, \forall \xi \in \Xi \\
 & \quad x_i(\xi) \leq y_i(\xi) M \quad \forall i \in H, \forall \xi \in \Xi \\
 & \quad y(\xi) \in \{0, 1\}^n, x(\xi), s(\xi), r(\xi) \geq 0 \quad \forall \xi \in \Xi.
 \end{aligned}$$

Hence, a formulation for problem \mathcal{PI}_1 can be obtained from the above formulation by removing the dependency of ξ from y , while a formulation for \mathcal{PI}_0 is obtained by removing the variables y and constraints associated with it. This approach is well-known in stochastic optimization to examine the quality of proposed solutions. In particular, it is used to compute the so-called *expected value of perfect information* which defines the maximum price that one would be ready to pay to obtain perfect information about the actual scenario, see [17].

We discuss in the next section how to solve problems \mathcal{PI}_0 , \mathcal{PI}_1 and \mathcal{PI} .

4. Solving the problem with perfect information

Let us first introduce some useful definitions. The *cumulative cost* w_{ij} represents the unitary cost of producing at time period i to satisfy the demand of time period j :

$$w_{ij} = c_i + \sum_{l=i}^{j-1} h_l + \sum_{l=j}^{i-1} p_l.$$

Notice that the above is well-defined, since for a fixed period i , $i \neq j$, only one of the two summations is not empty. For the sake of simplicity, we denote by ω_j the *minimum cumulative cost* for period j , the smallest among values $\{w_{ij}, i \in H\}$. Finally, we denote by $opt(X)$ the optimal solution cost of any optimization problem X .

In the following, we discuss how to solve the optimization problems obtained by relaxing the non-anticipativity constraints. We first focus on problem \mathcal{PI}_0 , then we address problem \mathcal{PI}_1 , and we finish with problem \mathcal{PI} . For each problem, we present a generic solution algorithm that can handle general uncertainty polytopes and more efficient algorithms that are tailored for the budgeted uncertainty polytope.

4.1. No setup

We first deal with the robust problem without setup costs \mathcal{PI}_0 , which can be formulated as follows

$$\begin{aligned}
(\mathcal{PI}_0) \quad & \min \quad \kappa \\
& s.t \quad \kappa \geq \sum_{i \in H} (c_i x_i(\xi) + h_i s_i(\xi) + p_i r_i(\xi)) \quad \forall \xi \in \Xi \\
& \quad \quad s_{i+1}(\xi) = x_i(\xi) - d_i(\xi) + s_i(\xi) - r_{i-1}(\xi) + r_i(\xi) \quad \forall i \in H, \forall \xi \in \Xi \\
& \quad \quad x(\xi), s(\xi), r(\xi) \geq 0 \quad \forall i \in H, \forall \xi \in \Xi.
\end{aligned}$$

We show that \mathcal{PI}_0 is equivalent to problem

$$\max_{\xi \in \Xi} \sum_{i \in H} \omega_i d_i(\xi),$$

where ω_i is the minimum cumulative cost of period i as defined previously. Hence, the result shows that the complexity of \mathcal{PI}_0 is related to the complexity of optimizing an affine function over Ξ .

Theorem 1. *Let Ξ be any uncertainty set. Then,*

$$\text{opt}(\mathcal{PI}_0) = \max_{\xi \in \Xi} \sum_{i \in H} \omega_i d_i(\xi)$$

Proof. Because there is no capacity constraint, one readily verifies the following. For each time period i , there exists a unique time period j for which all demand of i is produced, which corresponds to the period that provides the minimum cumulative cost to serve the demand of time period i . Hence, given any $\xi \in \Xi$, we have to pay the total cost

$$\sum_{i \in H} \omega_i d_i(\xi)$$

Then, the absence of setup costs implies that the time period yielding the minimum production cost does not depend on ξ , proving the result. \square

Theorem 1 implies that \mathcal{PI}_0 is polynomially solvable in the input since linear programming is polynomially solvable. We show in the next result that we can get faster algorithms for Ξ_Γ when Γ is integer.

Corollary 1. *Let $\Xi = \Xi_\Gamma$ and define the subset $\mathbf{\Gamma}(\omega^T \hat{D}) \subseteq H$ that contains the indices of the Γ largest elements of the vector $(\sum_{i \in H} \omega_i \hat{D}_{ij}, j \in H)$, and $\mathbf{\Gamma}'(\omega^T \hat{D})$ that denotes the $(\Gamma + 1)$ -largest element of that vector. The following holds:*

$$\text{opt}(\mathcal{PI}_0) = \sum_{i \in H} \omega_i \bar{d}_i + \sum_{i \in \mathbf{\Gamma}(\omega^T \hat{D})} \omega^T \hat{D}_i + (\Gamma - \lfloor \Gamma \rfloor) \omega^T \hat{D}_{\mathbf{\Gamma}'(\omega^T \hat{D})}.$$

Moreover, $\text{opt}(\mathcal{PI}_0)$ can be computed in $O(n^2)$.

Proof. We obtain immediately from Theorem 1 that

$$\begin{aligned}
opt(\mathcal{PI}_0) &= \max_{\xi \in \Xi_\Gamma} \sum_{i \in H} \omega_i d_i(\xi) \\
&= \max_{\xi \in \Xi_\Gamma} \sum_{i \in H} \left(\omega_i \bar{d}_i + \sum_{j \in H} \omega_i \hat{D}_{ij} \xi_j \right) \\
&= \sum_{i \in H} \omega_i \bar{d}_i + \max_{\xi \in \Xi_\Gamma} \sum_{i, j \in H} \omega_i \hat{D}_{ij} \xi_j \\
&= \sum_{i \in H} \omega_i \bar{d}_i + \max_{\substack{\sum_{i \in H} \xi_i \leq \Gamma \\ -1 \leq \xi_i \leq 1}} \sum_{i, j \in H} \omega_i \hat{D}_{ij} \xi_j \\
&= \sum_{i \in H} \omega_i \bar{d}_i + \sum_{i \in \Gamma(\omega^T \hat{D})} \omega^T \hat{D}_i + (\Gamma - \lfloor \Gamma \rfloor) \omega^T \hat{D}_{\Gamma'(\omega^T \hat{D})}
\end{aligned}$$

Regarding the complexity, notice that we must calculate all cumulative costs, which takes $O(n^2)$. Then, we must calculate the minimum cumulative cost for every period, which takes $O(n^2)$. Finally, we must compute the vector $(\sum_{i \in H} \omega_i \hat{D}_{ij}, j \in H)$, which takes $O(n^2)$ and chooses the Γ larger values of it, which takes $O(\Gamma \log n)$. Hence, the complexity of this strategy is $O(3n^2 + \Gamma \log n) = O(n^2)$. \square

4.2. Non-adjustable setup

In this section, we address problem \mathcal{PI}_1 , which can be formulated as

$$\begin{aligned}
(\mathcal{PI}_1) \quad \min \quad & \kappa \\
s.t \quad & \kappa \geq \sum_{i \in H} (c_i x_i(\xi) + g_i y_i + h_i s_i(\xi) + p_i r_i(\xi)) \quad \forall \xi \in \Xi \\
& s_{i+1}(\xi) = x_i(\xi) - d_i(\xi) + s_i(\xi) - r_{i-1}(\xi) + r_i(\xi) \quad \forall i \in H, \forall \xi \in \Xi \\
& x_i(\xi) \leq y_i M \quad \forall i \in H, \forall \xi \in \Xi \\
& x(\xi), s(\xi), r(\xi) \geq 0 \quad \forall \xi \in \Xi \\
& y \in \{0, 1\}^n.
\end{aligned}$$

As in the previous section, we first address \mathcal{PI}_1 for general polytopes and show that the problem is \mathcal{NP} -hard and can be reformulated as mixed-integer linear program. We present then a polynomial approach that is applicable to Ξ assuming that \hat{D} is a diagonal matrix. We address first the complexity of \mathcal{PI}_1 , using a reduction from the partition problem.

Theorem 2. *The problem \mathcal{PI}_1 is \mathcal{NP} -hard.*

Proof. Let $R = \{1, \dots, r\}$ and consider a set of positive integers $\{a_i, i \in R\}$. The partition problem looks for subset $S \subset R$ such that

$$\sum_{i \in S} a_i = \sum_{i \in R \setminus S} a_i. \tag{15}$$

Finding out whether such a subset S exists is \mathcal{NP} -complete, see for instance [21]. We define the corresponding instance of \mathcal{PI}_1 by considering $H = \{1', 1'', \dots, r', r''\}$, setting the production costs c to 0, the fixed costs g to K (to be defined later), the holding and backlogging costs between i' and i'' to 1 for each $i \in R$, while those between periods i'' and $(i+1)'$ are set to a very large

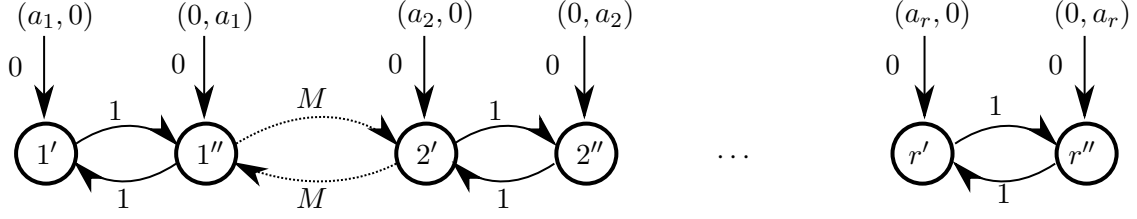


Figure 1: Reduction of the partition problem to \mathcal{PT}_1 .

number M (to be defined later), for each $i = 1, \dots, r-1$. Finally, $\bar{d} = 0$, \hat{D} is the identity matrix, and $\Xi \subset \mathcal{R}_+^{2r}$ is defined as the convex hull of the two vectors $\xi' = (a_1, 0, a_2, 0, \dots, a_r, 0)$ and $\xi'' = (0, a_1, 0, a_2, \dots, 0, a_r)$, see Figure 1. One readily verifies that the above polytope Ξ can be obtained through $4r + 2$ linear inequalities: the first $4r$ inequalities characterize the line joining ξ' and ξ'' while the last two inequalities bound the line to obtain the required segment. It is well-known (e.g. [4]) that we can restrict ourselves to the two extreme points of Ξ when analyzing the optimal solution of the problem and its cost.

Let $A = \sum_{i=1}^r a_i$. Choosing M large enough and $K = \max_{i=1, \dots, r} a_i + 1$, we prove below that there exists a subset S of R that satisfies (15) if and only if the optimal solution cost of the above instance of \mathcal{PT}_1 is equal to $rK + \frac{A}{2}$. Let us first show that, in any optimal solution to \mathcal{PT}_1 , either $y_{i'} = 1$ and $y_{i''} = 0$, or $y_{i'} = 0$ and $y_{i''} = 1$, for each $i \in R$. Specifically, setting $y_{i'} = 1$ and $y_{i''} = 0$ yields a cost of a_i for scenario ξ' and 0 for scenario ξ'' , in addition to the fixed cost K . Similarly, setting $y_{i''} = 1$ and $y_{i'} = 0$ yields a cost of 0 for scenario ξ' and a_i for scenario ξ'' , in addition to the fixed cost K . Hence, both approaches cost $K + a_i$ in the worst-case. This is always less costly than setting $y_{i'} = y_{i''} = 0$, which costs at least Ma_i in both scenarios, or setting $y_{i'} = y_{i''} = 1$, which costs $2K$ in both scenarios.

Let $S \subseteq R$ denote the elements for which $y_{i'} = 1$, so that $R \setminus S$ contains the elements of R for which $y_{i''} = 1$. We see that the cost of the solution described by S is equal to

$$rK + \max \left(\sum_{i \in S} a_i, \sum_{i \in R \setminus S} a_i \right).$$

Hence, the cost of S is minimized if and only if (15) holds, proving the result \square

The above result is in line with classical results on robust combinatorial optimization problems (e.g. [24]) which show that the robust counterparts of polynomially solvable optimization problems turn \mathcal{NP} -hard for arbitrary uncertainty sets. This being said, it is possible to solve \mathcal{LS}_1 for general uncertainty polytopes through a mixed-integer linear programming reformulation. The first step to obtain the reformulations relies on reformulating \mathcal{LS}_1 as static robust optimization problem, hence, removing the adjustable variables from the problem.

Lemma 1. *Problem \mathcal{PT}_1 can be reformulated as*

$$\min \left(\max_{\xi \in \Xi} \sum_{i,j \in H} w_{ij} v_{ij} d_j(\xi) + \sum_{i \in H} g_i y_i \right)$$

$$s.t. \quad \sum_{i \in H} v_{ij} = 1 \quad \forall j \in H \quad (16)$$

$$v_{ij} \leq y_i \quad \forall i, j \in H \quad (17)$$

$$y \in \{0, 1\}^n, v \geq 0. \quad (18)$$

Proof. Let us define the binary variable $v_{ij}(\xi)$ that is equal to 1 iff the demand of period j is produced at period i . Using a reasoning similar to the one used in the proof of Theorem 1, we can reformulate \mathcal{PT}_1 as

$$\begin{aligned} \min_{y \in \{0,1\}^n} \max_{\xi \in \Xi} \min_{v(\xi)} \sum_{i,j \in H} w_{ij} v_{ij}(\xi) d_j(\xi) + \sum_{i \in H} g_i y_i \\ \text{s.t. } \sum_{i \in H} v_{ij}(\xi) = 1 \quad \forall j \in H, \forall \xi \in \Xi \end{aligned} \quad (19)$$

$$\begin{aligned} v_{ij}(\xi) \leq y_i \quad \forall i, j \in H, \forall \xi \in \Xi \quad (20) \\ v \geq 0, \end{aligned}$$

where the first term of the objective function represents the cost of attending the demand of all periods while the second term represents the fixed costs. Notice then that, for any $y \in \{0, 1\}^n$, we always have that $v_{ij}(\xi) = 1$ iff

$$w_{ij} = \min_{i' \in H} \{w_{i'j} : y_{i'} = 1\},$$

which does not depend on ξ . Hence, $v_{ij}(\xi) = v_{ij}(\xi')$ for each pair $\xi, \xi' \in \Xi$ and we can remove the dependency on ξ from $v_{ij}(\xi)$. Moreover, we can remove the binary restrictions on v because constraints (19) and (20) are formed by a totally unimodular matrix. Applying von Neumann's minmax theorem, we can exchange the maximization over ξ and the minimization over v obtaining the desired result. \square

Let $\theta \in \mathbb{R}^m$ denote the dual variables associated to the constraints that characterize Ξ . Dualizing the inner maximization problem obtained in Lemma 1, we obtain immediately a MILP reformulation for \mathcal{PT}_1 .

Corollary 2. *Let Ξ be the uncertainty polytope defined in (10). The problem \mathcal{PT}_1 can be solved by the following mixed integer linear program*

$$\begin{aligned} \text{opt}(\mathcal{PT}_1) = \min \sum_{l=1}^m q_l \theta_l + \sum_{i,j \in H} v_{ij} w_{ij} \bar{d}_j + \sum_{i \in H} g_i y_i \\ \text{s.t. } \sum_{l=1}^m W_{lk} \theta_l \geq \sum_{i,j \in H} v_{ij} w_{ij} \hat{D}_{jk} \quad \forall k \in H, \\ \sum_{i \in H} v_{ij} = 1 \quad \forall j \in H \\ v_{ij} \leq y_i \quad \forall i, j \in H \\ y \in \{0, 1\}^n, v \geq 0. \end{aligned}$$

We devote the rest of the section to the study of uncertainty set Ξ_Γ , further assuming that \hat{D} is a diagonal matrix. We show that, under these assumptions, problem \mathcal{PT}_1 can be solved in polynomial time. Our approach requires the following extension of a classical result from Bertsimas and Sim [11] that shows how some robust combinatorial optimization problems can be solved by solving a polynomial number of times the deterministic counterpart. The result considers generic optimization problems defined by the optimization variables $x \in \{0, 1\}^L$ and the uncertainty polytope $\Xi_\Gamma \subset \mathbb{R}^K$, where K and L are arbitrary index sets. Its proof is similar to the proof of Theorem 2 from [18] and is therefore deferred to Appendix C.

Lemma 2. Consider the index sets K and L , and consider the subset $K(l) \subseteq K$ for each $l \in L$. Let $\mathcal{X} \subseteq \{0, 1\}^{|L|}$ be the feasibility set and $\alpha \in \mathcal{R}_+^{|L|}$ and $\beta \in \mathcal{R}_+^{|L| \times |K|}$ be the cost vectors that characterize the robust optimization problem

$$\min_{x \in \mathcal{X}} \max_{\xi \in \Xi_\Gamma} \sum_{l \in L} \left(\alpha_l + \sum_{k \in K(l)} \beta_{lk} \xi_k \right) x_l. \quad (21)$$

Define the subset $L(k) = \{l \in L : k \in K(l)\}$ for each $k \in K$. If any element $x \in \mathcal{X}$ satisfies the constraints

$$\sum_{l \in L(k)} x_l = 1, \quad k \in K,$$

then the optimal solution cost to problem (21) is given by

$$\min_{\{(k', l') : k' \in K, l' \in L(k)\} \cup \{(0, 0)\}} G_{k'l'}$$

where

$$G_{k'l'} = \Gamma \beta_{k'l'} + \min_{x \in \mathcal{X}} \sum_{l \in L} \left(\alpha_l + \sum_{k \in K(l)} \max(0, \beta_{lk} - \beta_{k'l'}) \right) x_l$$

for each $\{(k', l') : k' \in K, l' \in L(k)\}$ and where $\beta_{00} = 0$.

To use Lemma 2, we should reformulate problem \mathcal{PT}_1 as a special case of (21). In this aim, we use the shortest path formulation for the deterministic problem \mathcal{LS} , presented in [27], and modify that formulation to handle the robust aspect as required. This is presented in the following theorem.

Theorem 3. Let $\Xi = \Xi_\Gamma$ and assume that \hat{D} is a diagonal matrix. Then, problem \mathcal{PT}_1 can be solved by solving $O(n^2)$ shortest path problems in an acyclic digraph that contains n nodes.

Proof. Consider the digraph $G = (V, A)$, where $V = \{1, \dots, n+1\}$ and $A = \{(i, j) | i < j; i, j \in V\}$. When the demand is known with precision ($d(\xi) = \bar{d}$), it is known (see Section 7.4 from [27]) that the optimization problem stated in Lemma 1 can be reformulated as a shortest path problem between vertices 1 and $n+1$ in G where the cost of taking arc (i, j) is equal to $g_i + \sum_{k=i}^j w_{ik} \bar{d}_k$, which means that all demands $i \leq k \leq j$ are attended by producing at time period $i \in H$. Letting $\mathcal{Z} \subset \{0, 1\}^{|A|}$ be the set of incidence vectors of all paths from 1 to $n+1$ in G , the deterministic problem can thus be reformulated as

$$\min_{z \in \mathcal{Z}} \sum_{(i,j) \in A} \left(g_i + \sum_{k=i}^j w_{ik} \bar{d}_k \right) z_{ij}. \quad (22)$$

Since any demand $k \in H$ is attended by producing at a unique period $i \in H$, we see that any $z \in \mathcal{Z}$ must satisfy the constraints

$$\sum_{\substack{(i,j) \in A \\ i \leq k \leq j}} z_{ij} = 1, \quad \forall k \in H.$$

Thanks to Lemma 1, one readily verifies that the robust counterpart of the problem can be similarly reformulated as the minmax robust counterpart of problem (22), namely

$$\min_{z \in \mathcal{Z}} \max_{\xi \in \Xi} \sum_{(i,j) \in A} \left(g_i + \sum_{k=i}^j w_{ik} d_k(\xi) \right) z_{ij} = \min_{z \in \mathcal{Z}} \max_{\xi \in \Xi} \sum_{(i,j) \in A} \left(g_i + \sum_{k=i}^j w_{ik} \bar{d}_k + \sum_{k=i}^j w_{ik} \hat{D}_{kk} \xi_k \right) z_{ij}, \quad (23)$$

where the equality follows from the assumption on \hat{D} . The result follows directly from applying Lemma 2 to (23). \square

4.3. Adjustable setup

In this section we present a linear program that computes the value of $\text{opt}(\mathcal{PI})$ for general polytopes Ξ . Similarly to Theorem 1, we show that the complexity of problem \mathcal{PI} is related to the complexity of optimizing an affine function over Ξ . This time however, the linear program to be solved contains $O(n^3)$ constraints in addition to those describing Ξ . Let us first recall a well-known dynamic programming algorithm to solve the deterministic problem \mathcal{LS} , taken from [27].

Lemma 3. *Problem \mathcal{LS} can be solved by the following recursive function*

$$\text{opt}(\mathcal{LS}) = G(n)$$

where

$$G(j) = \min_{\{i,k|i < k \leq j\}} \left(G(i) + g_k + \sum_{l=i+1}^j w_{kl}d_l \right), \quad (24)$$

where $G_0 = 0$.

To compute \mathcal{PI} , it will be useful to reformulate the above dynamic program as the linear program provided in the next lemma.

Lemma 4. *The optimal solution cost of \mathcal{LS} is equal to the optimal solution cost of the following linear program, with optimization variables u_i for each $i \in H$*

$$\begin{aligned} \max \quad & u_n \\ \text{s.t.} \quad & u_j \leq u_i + g_k + \sum_{l=i}^j w_{kl}d_l \quad \forall i < k \leq j, i \in H \cup \{0\}, j \in H. \\ & u_0 = 0 \end{aligned}$$

Proof. We introduce optimization variable u_i to represent the value of $G(i)$. The counterpart of equation (24) for u is

$$u_j = \min_{\{i,k|i \leq k \leq j\}} \left(u_i + g_k + \sum_{l=i}^j w_{kl}d_l \right). \quad (25)$$

Then, we relax the equality in (25) to

$$u_j \leq \min_{\{i,k|i \leq k \leq j\}} \left(u_i + g_k + \sum_{l=i}^j w_{kl}d_l \right), \quad (26)$$

and ensure that the objective function of the linear program maximizes the value of u_j . Constraint (26) can be easily linearized to

$$u_j \leq u_i + g_k + \sum_{l=i}^j w_{kl}d_l \quad \forall i \leq k \leq j, i \in H. \quad (27)$$

Combining linear constraints (27) with the objective function that maximizes u_n yields the result. \square

Using the above results, we propose a linear programming reformulation for \mathcal{PI} .

Theorem 4. *For any uncertainty set Ξ , problem \mathcal{PI} can be solved by the following linear program in optimization vectors u and ξ*

$$\begin{aligned} \mathcal{PI} = \max \quad & u_n \\ \text{s.t.} \quad & u_j \leq u_i + g_k + \sum_{l=i}^j w_{kl} d_l(\xi) \quad \forall i \leq k \leq j, i \in H, j \in H \\ & \xi \in \Xi \end{aligned}$$

Proof. We define $\mathcal{PI}(\xi)$, as the problem \mathcal{PI} restricted to a fixed element $\xi \in \Xi$. One readily sees that

$$\text{opt}(\mathcal{PI}) = \max_{\xi \in \Xi} \text{opt}(\mathcal{PI}(\xi))$$

and moreover, $\mathcal{PI}(\xi)$ is a deterministic lot-sizing problem. Hence, we can apply Lemma 3 and use the following dynamic program for solving problem $\mathcal{PI}(\xi)$

$$\text{opt}(\mathcal{PI}(\xi)) = G_\xi(n)$$

where

$$G_\xi(j) = \min_{\{i, k | i \leq k \leq j\}} \left(G_\xi(i) + g_k + \sum_{l=i}^j w_{kl} d_l(\xi) \right).$$

Using Lemma 4, $\mathcal{PI}(\xi)$ has the same optimal solution as the linear program

$$\begin{aligned} \text{opt}(\mathcal{PI}(\xi)) = \max \quad & u_n \\ \text{s.t.} \quad & u_j \leq u_i + g_k + \sum_{l=i}^j w_{kl} d_l(\xi) \quad \forall i \leq k \leq j, i \in H, j \in H. \end{aligned} \quad (28)$$

Now, as $\text{opt}(\mathcal{PI}) = \max_{\xi \in \Xi} \text{opt}(\mathcal{PI}(\xi))$ we have that

$$\begin{aligned} \text{opt}(\mathcal{PI}) = \max \quad & u_n \\ \text{s.t.} \quad & u_j \leq u_i + g_k + \sum_{l=i}^j w_{kl} d_l(\xi) \quad \forall i \leq k \leq j, i \in H, j \in H \\ & \xi \in \Xi \end{aligned}$$

Recalling that $d(\xi)$ is an affine function of ξ ends the proof. \square

5. Numerical experiments

In this section, we present the experimental results obtained with the relaxation presented in the last sections for problems \mathcal{P}_1 and \mathcal{P}_0 . We use the budgeted polytope Ξ_Γ as uncertainty set due to its importance in the literature and the identity as the deviation matrix, so $d(\xi)_i = \bar{d}_i + \hat{d}_i \xi_i$.

We compare the quality of our lower bound with the one provided by dual affine decision rules from [25], recalled in Section 3.2. We also compare these lower bounds with the upper bound provided by the affine decisions rules (called primal affine decision rules [25]). We do not carry out experiments for problem \mathcal{P} because the primal and dual affine decision rules methods cannot be applied to this problem so that we have no comparison possible for our method.

One point that must be clarified before we proceed concerns the probabilistic distribution used in dual affine decision rules. We use a uniform distribution over the extremes points of

the uncertainty polytope. The reasons for that are two-fold: first we need a distribution that allows us to calculate matrix \mathcal{M} (the exception matrix) in a simple way, otherwise we would spend a lot of time and computational effort to obtain such a matrix; second, as we do not have a dominating element for our instances it is not clear whether using a distribution that is concentrated in a single point of the uncertainty polytope as the authors did in [25].

The tests were carried out on an Intel(R) Core(TM) i7 CPU M60, 2.6Hz 4GB Ram machine and all formulations and algorithms were coded in C++, compiled with a GNU G++ 4.5 compiler and IBM CPLEX 12.3. In the following, we explain how the instances are built and which experiments were carried out. We also comment the results presented.

5.1. Instances

We start by the description of the instances. We consider two set of instances:DOWN and DYN. The instances in the set DYN represent lot-sizing problems in which the costs associated are seasonal. Such test of instances is inspired by [6] and they fulfill a criteria know as *Wagner-Within*, which can be seen in [?]. Roughly, this criteria implies that is always cheaper to produce the client demands of each period in the period itself. More precisely, we have for each period i that $c_i = 20 + 5 \sin(\frac{i\pi}{12})$, $h_i = 5 + 2 \sin(\frac{i\pi}{12})$ and $p_i = 7 + 2 \sin(\frac{i\pi}{12})$

To contrast with such set of instances, we use a second set of instances that do not fulfill the Wagner-Within criteria. In the set of instances named DOWN, for each each period i is cheaper to produce the client demand in the period $\lfloor \frac{i}{3} \rfloor$. More precisely, we have that $c_i = 10 + 5(i \bmod 2)$, $h_i = 3$ and $p_i = 4$.

We consider the deviations fixed as 20% of the nominal demand, and the nominal deviations are given for $100 + 50 \sin(\frac{i\pi}{12})$ for each period i .

We consider horizons of planning that have 20, 30, 40, 50, 60, 70, 80, 90 and 100 periods. We consider the amount of deviations fixed as 20% of the nominal demand, and the nominal deviations are given for $100 + 50 \sin(\frac{i\pi}{12})$ for each period i .

We tested two different values of the parameter Γ for each number of periods in the horizon of planning, each of them depending on a given probability level (which come from the probabilistic bounds computed in [13]). Table 1 presents the values used for the experiments.

H	$\Gamma^{0.01}$	$\Gamma^{0.1}$
10	5	8
20	7	11
30	8	14
40	9	16
50	10	17
60	11	19
70	12	20
80	12	22
90	13	23
100	14	24

Table 1: Values of parameter Γ .

5.2. Lot-Sizing problem without setup costs

We present not only a comparison between the value of the bound properly stated but we also report the difference in execution time.

Let *AFFINE* represent the dual affine rules presented in the Section 3.2, *PI* the perfect information proposed and *DUAL* the affine decision rules presented in Section 3.1. We compare an approximation of the *optimality gap* of *PI* and *DUAL* using the solution of the approach

AFFINE as an upper bound. We define the *approximative optimality gap* for *PI* concerning the problem *AFFINE* in the instance I with t periods as

$$\frac{\text{opt}(\text{AFFINE}(t, I)) - \text{opt}(\text{PI}(t, I))}{\text{opt}(\text{AFFINE}(t, I))}.$$

In the same way we define the approximative approximation gap for the problem *DUAL* concerning the problem *AFFINE* in the instances I with t periods as

$$\frac{\text{opt}(\text{AFFINE}(t, I)) - \text{opt}(\text{DUAL}(t, I))}{\text{opt}(\text{AFFINE}(t, I))}.$$

In the following, we report the approximative optimality gap and the solution times, computed for each number of periods and each set of instances.

5.2.1. Instances DOWN

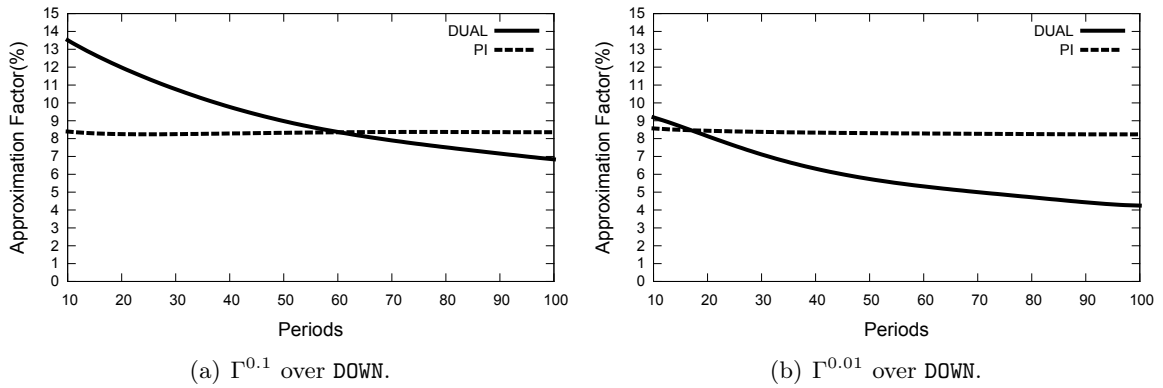


Figure 2: The optimality gaps for \mathcal{PI}_0 for the instances in DOWN.

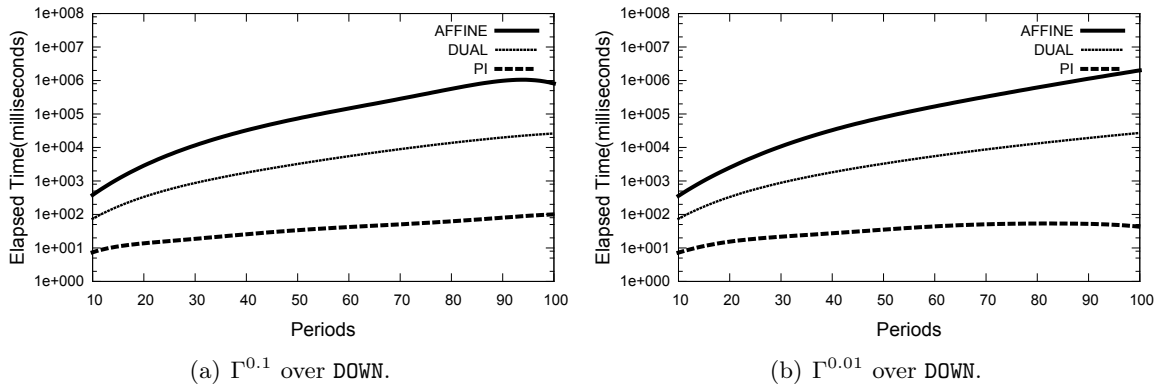


Figure 3: Logarithm of the elapsed time for \mathcal{PI}_0 to instances in DOWN.

Concerning the instances in the set DOWN, the approximation obtained by our approach is nearly constant at 8% while the dual affine rules are decreasing with the number of periods, Figure 2. Such behavior is understandable, because in this set of instances, the impact of the non-anticipativity constraints is bigger since the client demand is usually produced in early periods. Regardless that, our approach still provides a quite reliable approximation with a computation time much smaller as presented in Figure 3

5.2.2. Instances DYN

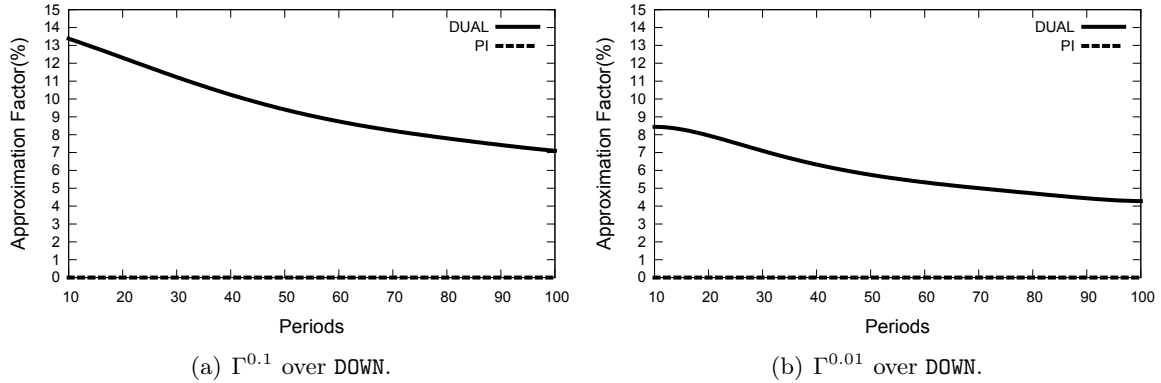


Figure 4: The optimality gaps for \mathcal{PI}_0 for the instances in DYN.

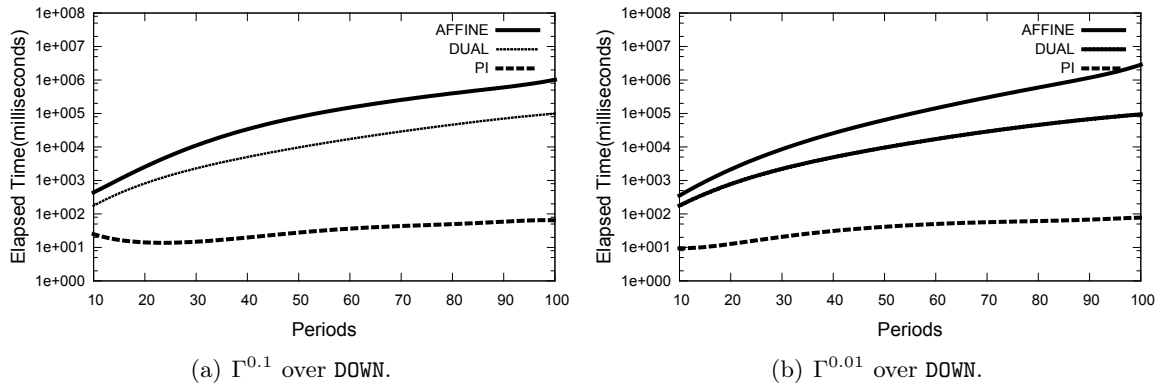


Figure 5: Logarithm of the elapsed time for \mathcal{PI}_0 to instances in DYN.

Concerning the instances in the set DYN, the approximation obtained by our approach can prove the optimality of the affine decisions approach while the dual affine rules, although decreasing the number of periods, still maintain a gap of 5% as we can see in Figure 4. Due to the Wagner-Within criteria, the non-anticipativity constraints have little impact in the solution, which explains the results obtained. As for the computation time, our approach is, being a simple combinatorial algorithm, much faster than the other two as reported in Figure 5, it is also interesting to notice that the dual affine decision rules approach is more time consuming in the instance set DYN.

6. Concluding remarks

We propose in this paper to adapt the perfect information relaxation to a robust lot-sizing problem to obtain a lower bound for the problem. The lower bound seems to perform very well numerically, especially for large problems for which primal and dual decision rules require large numbers of variables. One advantage of our approach is that it can handle binary adjustable variables, which is not the case of primal and dual affine decision rules. The tractability of our lower bound is highly problem-dependent. Hence, it would be interesting to investigate other class of multi-stage robust optimization problems and other uncertainty sets. Concerning the extensions of the problems under study, we think it may be interesting to consider problems with

capacities on the production and the storage. An alternative extension would consider problems with multiple items. Concerning the uncertainty set, it could be interesting to look for dedicated algorithms for ellipsoidal uncertainty sets, which have recently been the topic of several papers in the combinatorial robust optimization literature (e.g. [5]). Another aspect of the relaxation with perfect information is that it turns the multi-stage problems into a two-stages problems, for which several papers have recently proposed exact solution algorithms based on variants of the Benders decomposition (e.g. [4, 16, 29]). Hence, one could use these algorithms to solve the relaxation for much more general problems than the variants of the lot-sizing considered herein, and assess the quality of the obtained lower bound.

- [1] A. Agra, M. Christiansen, R. M. V. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo. The robust vehicle routing problem with time windows. *Computers & OR*, 40(3):856–866, 2013.
- [2] A. Agra, M. C. Santos, D. Nace, and M. Poss. A dynamic programming approach for a class of robust optimization problems. *SIAM Journal on Optimization*, 26(3):1799–1823, 2016.
- [3] M. Avriel and A. Williams. The value of information and stochastic programming. *Operations Research*, 18(5):947–954, 1970.
- [4] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.
- [5] F. Baumann, C. Buchheim, and A. Ilyina. Lagrangean decomposition for mean-variance combinatorial optimization. In *Combinatorial Optimization - Third International Symposium, ISCO 2014, Lisbon, Portugal, March 5-7, 2014, Revised Selected Papers*, pages 62–74, 2014.
- [6] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Program.*, 99(2):351–376, 2004.
- [7] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424.
- [8] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.
- [9] D. Bertsimas and I. Dunning. Multistage robust mixed integer optimization with adaptive partitions, 2014.
- [10] D. Bertsimas and A. Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.
- [11] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.
- [12] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.
- [13] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- [14] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Oper. Res.*, 54(1):150–168, Jan. 2006.
- [15] D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5(2):389 – 414, 2008. In Memory of George B. Dantzig.
- [16] A. Billionnet, M. Costa, and P. Poirion. 2-stage robust MILP with continuous recourse variables. *Discrete Applied Mathematics*, 170:21–32, 2014.
- [17] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
- [18] M. Bougeret, A. Pessoa, and M. Poss. Robust scheduling with budgeted uncertainty. 2016. Available online at <https://hal.archives-ouvertes.fr/hal-01345283/document>.
- [19] W. T. Z. C. C. Huang, I. Vertinsky. Sharp bounds on the value of perfect information. *Operations Research*, 25(1):128–139, 1977.
- [20] X. Chen and Y. Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.
- [21] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.
- [22] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Oper. Res.*, 58(4-Part-1):902–917, July 2010.
- [23] B. L. Gorissen and D. den Hertog. Robust counterparts of inequalities containing sums of maxima of linear functions. *European Journal of Operational Research*, 227(1):30–43, 2013.
- [24] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.
- [25] D. Kuhn, W. Wiesemann, and A. Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130:177–209, 2011.
- [26] J. B. Oostenbrink, M. J. Al, M. Oppe, and M. P. Rutten-van Mólken. Expected value of perfect information: an empirical example of reducing decision uncertainty by conducting additional research. *Value in Health*, 11(7):1070–1080, 2008.

- [27] Y. Pochet and L. Wolsey. *Production Planning by Mixed Integer Programming*. Springer Science & Business Media, 2013.
- [28] K. Postek and D. den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.
- [29] B. Zeng and L. Zhao. Solving two-stage robust optimization problems by a constraint-and-column generation method. *Operations Research Letters*, 41(5):457–461, 2013.

Appendix A. Reformulation of affine decision rules

Plugging (12)–(14) into \mathcal{P} and enforcing that y does not depend on ξ , we obtain the formulation below.

$$\begin{aligned}
\min_{\kappa} \quad & \kappa \\
\text{s.t.} \quad & \kappa \geq \sum_{i \in H} \left(g_i y_i + c_i (x_i^0 + \sum_{j=1}^{i-f} x_i^j \xi_j) + h_i (s_i^0 + \sum_{j=1}^{i-f} s_i^j \xi_j) + p_i (r_i^0 + \sum_{j=1}^{i-f} r_i^j \xi_j) \right) & \forall \xi \in \Xi \\
& s_i^0 + \sum_{j=1}^{i-f} s_i^j \xi_j \geq \sum_{k=1}^i \left(x_k^0 - \bar{d}_k + \sum_{j=1}^{l_k^x} x_k^j \xi_j - \sum_{l=1}^n \hat{D}_{kl} \xi_l \right) & \forall i \in H, \forall \xi \in \Xi \\
& r_i^0 + \sum_{j=1}^{i-f} r_i^j \xi_j \geq \sum_{k=1}^i \left(\bar{d}_k - x_k^0 - \sum_{j=1}^{l_k^x} x_k^j \xi_j + \sum_{l=1}^n \hat{D}_{kl} \xi_l \right) & \forall i \in H, \forall \xi \in \Xi \\
& x_i^0 + \sum_{j=1}^{i-f} x_i^j \xi_j \leq y_i M & \forall i \in H, \forall \xi \in \Xi \\
& x_i^0 + \sum_{j=1}^{i-f} x_i^j \xi_j \geq 0 & \forall i \in H, \forall \xi \in \Xi \\
& s_i^0 + \sum_{j=1}^{i-f} s_i^j \xi_j \geq 0 & \forall i \in H, \forall \xi \in \Xi \\
& r_i^0 + \sum_{j=1}^{i-f} r_i^j \xi_j \geq 0 & \forall i \in H, \forall \xi \in \Xi \\
& y_i \in \{0, 1\} & \forall i \in H,
\end{aligned}$$

Although the number of variables in the problem described above is polynomial, we still have to deal with an infinite number of constraints. As said already, one easily sees that we can restrict ourselves to the extreme points of Ξ ; yet, this typically leads to an exponential number of constraints. An alternative and more compact approach apply classical tools from robust optimization to reformulate the robust constraints as a polynomial number of deterministic constraints plus a polynomial number of additional real variables, see [7]. We reproduce the steps of this method in the following.

$$\begin{aligned}
\min_{\kappa} \quad & \kappa \\
s.t. \quad & \kappa - \sum_{i \in H} (g_i y_i + c_i x_i^0 + h_i s_i^0 + p_i r_i^0) \geq \sum_{i=1}^m q_i \theta_i \\
& \theta^T W_i \geq \sum_{j=i+f}^n (c_j x_j + h_j s_j + p_j r_j) \quad \forall i, 0 \leq i \leq m \\
& s_i^0 - \sum_{k=1}^i (x_k^0 - \bar{d}_k) \geq \sum_{j=1}^m q_j \alpha_j^i \quad \forall i \in H \\
& \alpha^T W_j \geq s_i^j + \sum_{k=i+f}^i x_k \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& r_i^0 - \sum_{k=1}^i (\bar{d}_k - x_k^0) \geq \sum_{j=1}^m q_j \beta_j^i \quad \forall i \in H \\
& \beta^T W_j \geq r_i^j - \sum_{k=i+f}^i x_k \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& -x_i^0 + y_i M \geq \sum_{j=1}^m q_j \pi_j^i \quad \forall i \in H \\
& \pi^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& x_i^0 \geq \sum_{j=1}^m q_j \theta_j^{x,i} \quad \forall i \in H \\
& (\theta^x)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& s_i^0 \geq \sum_{j=1}^m q_j \theta_j^{s,i} \quad \forall i \in H \\
& (\theta^r)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& r_i^0 \geq \sum_{j=1}^m q_j \theta_j^{r,i} \quad \forall i \in H \\
& (\theta^s)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& y_i \in \{0, 1\} \quad \forall i \in H,
\end{aligned}$$

To obtain the formulation for the cases without setup cost it suffices to remove the variable y from the formulation and the cost associated with it.

$$\begin{aligned}
& \min_{\kappa} \quad \kappa \\
& s.t. \quad \kappa - \sum_{i \in H} (c_i x_i^0 + h_i s_i^0 + p_i r_i^0) \geq \sum_{i=1}^m q_i \theta_i \\
& \theta^T W_i \geq \sum_{j=i+f}^n (c_j x_j + h_j s_j + p_j r_j) \quad \forall i, 0 \leq i \leq m \\
& s_i^0 - \sum_{k=1}^i (x_k^0 - \bar{d}_k) \geq \sum_{j=1}^m q_j \alpha_j^i \quad \forall i \in H \\
& \alpha^T W_j \geq s_i^j + \sum_{k=i+f}^i x_k \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& r_i^0 - \sum_{k=1}^i (\bar{d}_k - x_k^0) \geq \sum_{j=1}^m q_j \beta_j^i \quad \forall i \in H \\
& \beta^T W_j \geq r_i^j - \sum_{k=i+f}^i x_k \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& x_i^0 \geq \sum_{j=1}^m q_j \theta_j^{x,i} \quad \forall i \in H \\
& (\theta^x)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& s_i^0 \geq \sum_{j=1}^m q_j \theta_j^{s,i} \quad \forall i \in H \\
& (\theta^r)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H \\
& r_i^0 \geq \sum_{j=1}^m q_j \theta_j^{r,i} \quad \forall i \in H \\
& (\theta^s)^T W_j \geq x_i^j \quad \forall i, j, 0 \leq j \leq i - f, i \in H
\end{aligned}$$

Appendix B. Reformulation of dual affine decision rules

In this appendix, we sketch how to obtain the reformulation for the robust lot-sizing problem using the method presented in [25]. First, we associate an artificial probability measure μ with our uncertainty set Ξ . To keep the presentation simple, we will exemplify the framework from [25] with discrete probability distributions $\mathbb{P}(\xi)$ defined over the set of extreme points of Ξ . Hence, \mathbb{P} is any vector in $\mathbb{R}_+^{|\text{ext}(\Xi)|}$ that satisfies

$$\sum_{\xi \in \text{ext}(\Xi)} \mathbb{P}(\xi) = 1.$$

The first step of the approach of [25] relaxes the robust constraints to expectation constraints as follows.

$$\begin{aligned} \min_{\kappa} \quad & \kappa \\ \text{s.t.} \quad & \mathbb{E} \left(\left[\kappa - \sum_{i \in H} (g_i y_i + c_i x_i(\xi) + h_i s_i(\xi) + p_i r_i(\xi) + \pi(\xi)) \right] \xi \right) = 0 \\ & \mathbb{E} \left(\left[s_i(\xi) - \sum_{k=1}^i (x_k(\xi) - d_i(\xi)) + \alpha_i(\xi) \right] \xi \right) = 0 \quad \forall i \in H, \\ & \mathbb{E} \left(\left[r_i(\xi) + \sum_{k=1}^i (x_k(\xi) - d_i(\xi)) + \beta_i(\xi) \right] \xi \right) = 0 \quad \forall i \in H, \\ & \mathbb{E} ([x_i(\xi) - y_i M + \psi_i(\xi)] \xi) = 0 \quad \forall i \in H, \\ & \mathbb{E} ([x_i(\xi) + \theta_i^x(\xi)] \xi) = 0 \quad \forall i \in H, \\ & \mathbb{E} ([s_i(\xi) + \theta_i^s(\xi)] \xi) = 0 \quad \forall i \in H, \\ & \mathbb{E} ([r_i(\xi) + \theta_i^r(\xi)] \xi) = 0 \quad \forall i \in H, \\ & y_i \in \{0, 1\} \quad \forall i \in H, \end{aligned}$$

where ψ and θ are additional slack variables. One readily verifies that, for any probability distribution $\mathbb{P}(\xi)$, the above problem provides a lower bound for the optimal solution of problem \mathcal{PT}_1 .

Then, using advanced dualization techniques inspired by the dualization used in classical robust optimization, the authors of [25] are able to reformulate the above lower bound as the following mixed-integer linear program, where matrix \mathcal{M} is defined as $\mathcal{M} = \mathbb{E}(\xi^T \xi)$, and \mathcal{W} is the matrix defined as $\mathcal{W} = (W - qe_1^T)\mathcal{M}$, where e_1 is the vector with all entries equals to one. Specifically, the following mixed-integer linear program is obtained by applying reformulation (4.6) from [25] to \mathcal{PT}_1 .

$$\begin{aligned}
\min_{\kappa} \quad & \kappa \\
\text{s.t.} \quad & \kappa - \sum_{i \in H} (g_i y_i + c_i x_i^0 + h_i s_i^0 + p_i r_i^0) + \pi_0 = 0 \\
& -\pi_j = \sum_{i \in H} (c_i x_i^j + h_i s_i^j + p_i r_i^j) \quad \forall j \in H \\
& s_i^0 = \sum_{k=1}^i (x_k^0 - \bar{d}_k) + \alpha_i^0 \quad \forall i \in H \\
& s_i^j = \sum_{k=1}^i x_k^j - \sum_{l=1}^n \hat{D}_{jl} + \alpha_i^j \quad \forall i, j \in H \\
& r_i^0 = \sum_{k=1}^i (\bar{d}_k - x_k^0) + \beta_i^0 \quad \forall i \in H \\
& r_i^j = \sum_{k=1}^i (-x_k^0) + \sum_{l=1}^n \hat{D}_{jl} + \beta_i^j \quad \forall i, j \in H \\
& My - x_i^0 + \pi_i^0 = 0 \quad \forall i \in H \\
& -x_i^j + \pi_i^j = 0 \quad \forall i, j \in H \\
& x_i^0 + \theta_i^{x,0} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& s_i^j + \theta_i^{s,j} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& r_i^j + \theta_i^{r,j} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& \mathcal{W}\alpha^T = 0 \\
& \mathcal{W}\beta^T = 0 \\
& \mathcal{W}\pi^T = 0 \\
& \mathcal{W}(\theta^x)^T = 0 \\
& \mathcal{W}(\theta^s)^T = 0 \\
& \mathcal{W}(\theta^r)^T = 0 \\
& y_i \in \{0, 1\} \quad \forall i \in H.
\end{aligned}$$

To obtain the formulation for the cases without setup cost it suffices to remove the variable y from the formulation and the cost associated with it.

$$\begin{aligned}
& \min_{\kappa} \quad \kappa \\
& \text{s.t.} \quad \kappa - \sum_{i \in H} (c_i x_i^0 + h_i s_i^0 + p_i r_i^0) + \pi_0 = 0 \\
& \quad - \pi_j = \sum_{i \in H} (c_i x_i^j + h_i s_i^j + p_i r_i^j) \quad \forall j \in H \\
& \quad s_i^0 = \sum_{k=1}^i (x_k^0 - \bar{d}_k) + \alpha_i^0 \quad \forall i \in H \\
& \quad s_i^j = \sum_{k=1}^i x_k^j - \sum_{l=1}^n \hat{D}_{jl} + \alpha_i^j \quad \forall i, j \in H \\
& \quad r_i^0 = \sum_{k=1}^i (\bar{d}_k - x_k^0) + \beta_i^0 \quad \forall i \in H \\
& \quad r_i^j = \sum_{k=1}^i (-x_k^0) + \sum_{l=1}^n \hat{D}_{jl} + \beta_i^j \quad \forall i, j \in H \\
& \quad x_i^0 + \theta_i^{x,0} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& \quad s_i^j + \theta_i^{s,j} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& \quad r_i^j + \theta_i^{r,j} = 0 \quad \forall i \in H, \forall j \in H \cup \{0\} \\
& \quad \mathcal{W}\alpha^T = 0 \\
& \quad \mathcal{W}\beta^T = 0 \\
& \quad \mathcal{W}\pi^T = 0 \\
& \quad \mathcal{W}(\theta^x)^T = 0 \\
& \quad \mathcal{W}(\theta^s)^T = 0 \\
& \quad \mathcal{W}(\theta^r)^T = 0
\end{aligned}$$

Appendix C. Proof of Lemma 2

Let us detail the inner maximization of (21) as

$$\begin{aligned}
\sum_{k \in K} \alpha_k x_k \quad + \quad & \max \quad \sum_{k \in K} \xi_k \sum_{l \in L(k)} \beta_{lk} x_l \\
\text{s.t.} \quad & \sum_{k \in K} \xi_k \leq \Gamma, \\
& \xi_k \leq 1, \quad k \in K, \\
& \xi_k \geq -1, \quad k \in K.
\end{aligned} \tag{C.1}$$

By definition, $\beta_{lk} \geq 0$ for each $k \in K, l \in L(k)$, so that we can relax constraints (C.1) to $\xi \geq 0$ without affecting the optimal solution. Thanks to the strong duality in linear programming, the optimal solution cost of the above problem is equal to the optimal solution cost of its dual, given by

$$\begin{aligned}
\min \quad & \Gamma \theta + \sum_{k \in K} \varphi_k \\
\text{s.t.} \quad & \theta + \varphi_k \geq \sum_{l \in L(k)} \beta_{lk} x_l, \quad k \in K \\
& \theta, \varphi_k \geq 0.
\end{aligned}$$

Substituting φ_k by $\max(0, \sum_{l \in L(k)} \beta_{lk} x_l - \theta)$ for each $k \in K$, we can further reformulate (21) as

$$\min_{x \in \mathcal{X}, \theta \geq 0} \Gamma \theta + \sum_{l \in L} \alpha_l x_l + \sum_{k \in K} \max \left(0, \sum_{l \in L(k)} \beta_{lk} x_l - \theta \right). \tag{C.2}$$

The crucial step of our proof (which differs from Theorem 3 from [12]) is that, because the constraint $\sum_{l \in L(k)} x_l = 1$ holds for each $k \in K$, we can further reformulate (C.2) as

$$\min_{x \in \mathcal{X}, \theta \geq 0} \Gamma \theta + \sum_{l \in L} \alpha_l x_l + \sum_{k \in K} \sum_{l \in L(k)} x_l \max(0, \beta_{lk} - \theta).$$

The rest of the proof is identical to the proof of Theorem 3 from [12].