CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

New Solution Approaches for the Capacitated Supplier Selection Problem with Total Quantity Discount and Activation Costs under Demand Uncertainty

Daniele Manerba
Guido Perboli

November 2017

CIRRELT-2017-68

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

New Solution Approaches for the Capacitated Supplier Selection Problem with Total Quantity Discount and Activation Costs under Demand Uncertainty

Daniele Manerba[1,*], Guido Perboli[1,2]

[1] Department of Control and Computer Engineering and ICT for City Logistics and Enterprise (ICE) lab, Politecnicò di Torino, Corso Duca degli Abruzzi, 24 - I-10129 Torino, Italy

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract. The Capacitated Supplier Selection problem with Total Quantity Discount policy and Activation Costs is a procurement problem where a company needs to purchase a certain quantity of different products from a set of potential suppliers offering discounts based on the total quantity purchased. The buyer company wants to minimize the total expenditure that satisfies its own products demand, also considering the fixed costs needed to activate the business activities with the selected suppliers. We study the problem on a long-term perspective and thus consider the product demand as non-deterministic. Recent works have shown the importance of explicitly incorporate demand uncertainty in this economic setting, along with the evidence about the computational burden of solving the relative Stochastic Programming models for a sufficiently large number of scenarios. In this work, we propose different solution strategies (several variants of a Progressive Hedging based heuristic approach as well as a Benders algorithm) to efficiently cope with these models by taking advantage of the particular structure of the stochastic problem. The results obtained on benchmark instances show how the proposed methods outperform the existing ones and the state-of-the-art solvers in terms of efficiency and solution quality.

Keywords. Supplier selection, total quantity discount, stochastic demand, progressive hedging.

Acknowledgements. Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery Grants Program. While working on this paper, prof. Guido Perboli was the head of the Urban Mobility and Logistics Systems (UMLS) initiative of the interdepartmental Center for Automotive Research and Sustainable mobility (CARS) at Politecnico di Torino, Italy.

---

* Corresponding author: daniele.manerba@polito.it

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1 Introduction

In the specialized literature, the Supplier Selection problem deals with the (more tactical) decisions on which suppliers have to be used for a procurement, and with the (more operational) decisions on which amount of each required product/raw material has to be purchased in each visited supplier (Weber and Current, 1993). The aim is, in general, to satisfy a product demand at the minimum cost and thus is of great importance for the supply chain logistics of any company. Since the procurement processes potentially include a great variety of qualitative and quantitative factors to consider (reliability, economy of scale, transportation costs), many different Supplier Selection variants exist. In this work, we focus on a specific variant named the Capacitated Supplier Selection problem with Total Quantity Discount policy and Activation Costs under Demand Uncertainty (CTQD-AC$_{ud}$).

The CTQD-AC$_{ud}$ models, mixing tactical as well as operative decisions to minimize the overall procurement costs, the medium-long term procurement setting of a company that needs to purchase, from different potential suppliers, many different types of products, raw materials, or components for its business. This problem has several peculiarities. First, products have restricted availabilities at the suppliers (*capacitated*), thus creating the need to split the purchase of a product over different suppliers to satisfy its demand. Second, in order to incentive buyers, the suppliers propose a so-called *total quantity discount* (TQD) policy in which the cumulative quantity purchased (i.e., the number of units bought regardless of the type of products involved) determines the discount rate applied by the supplier to the total purchase cost. Note, however, that a buyer can benefit from the discounts proposed by a supplier only by activating, at the beginning of the purchasing period, a business contract. We assume that no more contracts can be activated after the initial choice and that no activated contracts can be terminated during the procurement period. The cost of activating a contract with a supplier guarantees the application of the declared discount as well as the availability and the price of the declared products. The buyer, from its side, is committed to purchase from that supplier an amount of products belonging to a certain range (i.e., the range related to the chosen discount). Finally, the products demand, being subject to uncertainty, is not known deterministically at the time of the contract activation with the suppliers, but it reveals only at the time of the actual operative purchasing.

Introduced in Manerba et al. (2017a), the CTQD-AC$_{ud}$ has been shown to lead to massive savings in costs when approached through Stochastic Programming (SP) techniques, giving a strong competitive advantage to the buyer company. Unfortunately, it has been also shown to be very hard to solve even for small-medium size instances and number of scenarios considered and the only existing branch-and-cut based solution framework shows poor efficiency, in particular when the number of suppliers and scenarios considered increases. This work is meant to overcome the lack of efficient algorithms to solve the CTQD-AC$_{ud}$ for large realistic instances and a consistent number of scenarios. To this aim, we develop and test a Benders algorithm and some variants of a Progressive Hedging (PH) based heuristic.

Apart from the efficiency, our main objective is also to create a method that preserves as much as possible the competitive advantage ensured by the SP model, i.e. finding solutions very close to the optimal ones.

The contribution of the present work is manifold. First, we enlarge the very limited specialized literature on supplier selection problems under both quantity discount policies and uncertainty while, especially in the recent years, these two aspect seem of critical importance in the purchasing assessment of a company resulting competitive in the globalized market. Second, we propose the first effective and efficient solution methods to cope with the CTQD-AC$_{ud}$ problem. Eventually, computational experiments on benchmark instances will show that our algorithms outperform state-of-the-art solvers and the existing branch-and-cut method both in terms of efficiency and quality of the solutions. Third, our new algorithms will give us the possibility to achieve optimal or very near-optimal solutions for all the non-closed benchmark instances. Finally, we believe that the acceleration strategies used to enrich the basic PH framework might be embedded, given their generality, in other similar solution algorithms for completely different problems.

The rest of the paper is organized as follows. In Section 2, we review the relevant literature about supplier selection under quantity discount policies and demand uncertainty. Section 3 presents the problem notation and its SP formulation with recourse. Section 4 discusses the basic development of our heuristic solution approach based on Progressive Hedging whereas Section 5 details all the acceleration strategies implemented in order to improve efficiency and effectiveness (including a primal heuristic and a multi-thread version). Section 6 presents and discusses all the results coming from our computational experience, together with the set of benchmark instances used for the tests. Finally, conclusions are drawn in Section 7.

2 Literature review

Given their critical role in determining firm logistic costs, Supplier Selection problems have been studied in the specialized literature since long through the use of both qualitative and quantitative methodologies and, up today, are still a vivid stream of research. Supplier Selection variants are so many and the relative literature is so vast that goes out of the scope of this paper (interested reader is referred to Dickson, 1966, Aissaoui et al., 2007, and Wetzstein et al., 2016 for surveys on the subject written in very different periods and with different focuses). In the following, instead, we will review the Supplier Selection literature related to quantity discounts and data uncertainty (main features of our CTQD-AC$_{ud}$).

Quantity discounts are price discounts provided by suppliers on the basis of the amount of product quantity purchased. Benton (1991), Munson and Rosenblatt (1998), and Munson and Jackson (2015) have analyzed the most relevant quantity discount scenarios from both the buyer's and seller's perspectives. In the recent decades, this practice has been studied mostly from a quantitative perspective in many different application contexts (dairy, chemical industry, project's resource investment, telecommunication systems) and considering many complicating factors such as multiple periods, multiple sites, inventory costs, buyers coalition, budgetary limitations and so on (see, e.g., McConnel and Galligan, 2004, van de Klundert et al., 2005, Mirmohammadi et al., 2009, Munson and Hu, 2010, Krichen et al., 2011, Jolai et al., 2013). Among the different existing policies (*incremental discount*, *fixed fees*, *truckload discount*), the *total quantity discount* (TQD) represents the most popular form applied so far and studied in the literature since it correctly models many multi-product procurement settings where the purchase is completed at a single point in time, without any auctions or rebate mechanisms (Crama et al., 2004, Shahsavar et al., 2016). Goossens et al. (2007) first studied the TQD as a combinatorial optimization problem, showed its $\mathcal{NP}$-hardness, and proposed a branch-and-bound algorithm based on a min-cost flow reformulation. Four TQD variants (considering market share constraints, the possibility of buying more than products demand, a limited number of winning suppliers, and multi-period context) are studied in the same paper. Moreover, the TQD capacitated version, in which quantities of product available at suppliers are limited, has been studied in Manerba and Mansini (2012, 2014) where the authors propose efficient branch-and-cut and Variable Neighborhood Decomposition Search (VNDS) matheuristic solution approaches. A TQD extension including transportation costs based on truckload shipping rates in also presented Mansini et al. (2012). Interesting enough, the TQD policy is considered also in some routing problems for inbound and outbound logistics (Nguyen et al., 2014, Manerba et al., 2017b).

Data uncertainty is a critical factor for many modeling frameworks. Concerning supplier selection problems, an explicit consideration of stochasticity has become more and more important in the recent years, in which the companies are supposed to sign long-term purchasing contracts in order to sustain their offer in a highly competitive and globalized market. Demand fluctuation is definitely the most studied type of uncertainty in the specialized literature (Yang et al., 2007, Awasthi et al., 2009, Zhang and Zhang, 2011). It is widely accepted that a precise forecast of the future product demand is hard to obtain for a company since it depends on several unknown a-priori internal and external factors. When considering long-term purchasing contracts, however, other problem's parameters may be subjected to volatility. For example, product prices and availabilities at the suppliers and their reliability are also affected by uncertainty due to market and environmental conditions (Anupindi and Akella, 1993, Parlar and Wang, 1993, Dada et al., 2007, Beraldi et al., 2017).

Only a few contributions consider both stochasticity and quantity discounts. Sen et al. (2013) model a multi-item, multi-supplier, multi-period, and multi-supplier problem of a large manufacturing company through a multi-stage stochastic optimization formulation. Heuristics are used to cope with three random events (a drop in price, a price change in the spot market, and a new discount offer). Hammami et al. (2014) propose a 2-stage SP model for a problem in the context of automotive manufacturing, integrating the exchange rate uncertainty, quantity discounts, transportation and inventory costs.

Very recently, Manerba et al. (2017a) enriched this quite unexplored research stream by introducing the CTQD-AC problem under uncertainty (CTQD-AC$_u$). The authors evaluate different sources of uncertainty (products demand, availability, and price) for such a long-term procurement settings and propose general and specific two-stage Stochastic Programming approaches to cope with them. Eventually, they focus on the special cases in which only the prices (CTQD-AC$_{up}$) or only the demands (CTQD-AC$_{ud}$) are stochastic. For both the problems, different scenario tree generations are developed, evaluated in terms of stability, and used to approximate the stochastic programs. The deterministic equivalent problems obtained are solved through an ad-hoc branch-and-cut algorithm exploiting valid inequalities, preprocessing routines, and a heuristic upper-bound. The findings show very clearly that using total quantity discount contracts to select suppliers represents itself a good way in mitigating the effects of products price fluctuations. Moreover, in the case of demand uncertainty, results show how an SP approach might lead to highly conservative and competitive solutions in terms of quality and percentage of quantities purchased at external suppliers. Unfortunately, this case is also the hardest in terms of CPU time needed for its exact solution. In the present work, we propose new efficient and effective solution approaches for the CTQD-AC$_{ud}$ special case.

3 Problem definition and formulation

This section presents, along with the necessary mathematical notation, an SP formulation with recourse for the CTQD-AC$_{ud}$. Let M be a set of suppliers and K be a set of products to be purchased. Each product $k \in K$, can be purchased in a subset $M_k \subseteq M$ of suppliers at a positive basic price f_{ik}, potentially different for each supplier $i \in M_k$, and without exceeding an available quantity q_{ik}. Note that, because of these restricted availabilities, the purchase of each product may be split over different suppliers. Each product has to be purchased for an amount greater than or equal to a positive integer demand, which is considered stochastic. We model this uncertainty by defining, for each demand of a product $k \in K$, an estimated deterministic component d_k and a stochastic oscillation $\widehat{d}_k(\xi)$, where ξ is a stochastic variable.

The TQD policy proposed by the supplier is modeled as follows. Each supplier $i \in M$ defines a set $R_i = \{1, \ldots, r_i\}$ of r_i consecutive and non-overlapping quantity intervals $[l_{ir}, u_{ir}]$ associated with a discount rate $\delta_{ir} \in [0, 1)$ such that $\delta_{i,r+1} \geq \delta_{ir}$ $r = 1, ..., r_i - 1$ (i.e., the higher the interval, the greater the discount). Then, for each supplier $i \in M$, the discount rate δ_{ir} is applied to the total purchase cost if the total quantity purchased lies in the interval $r \in R_i$, i.e., is greater than or equal to l_{ir} and less than or equal to u_{ir}. In order to guarantee the correct application of the discount policy to the entire purchase, we will assume $l_{i1} = 0, \forall i \in M$ and $\sum_{k \in K} q_{ik} \leq u_{i,r_i}, \forall i \in M$. However, the buyer can benefit from the discounts of a supplier $i \in M$ only by paying a fixed fee a_i required to activate the underlying business activity.

A two-stage SP formulation for the for CTQD-AC$_{ud}$ exploiting the tactical and the operational decisions involved in its procurement process is proposed in Manerba et al. (2017a). The first-stage decision is about which suppliers are involved in the purchasing, how much we expect to purchase from each supplier, and, consequently, in which discount interval we expect the total quantity of products purchased lies. The second-stage recourse actions consist in modifying the purchased quantities within the preselected interval for each supplier (locked by the first stage decisions). Moreover, if necessary to satisfy its demand, a further recourse action is to purchase a certain quantity of product k outside from the selected suppliers (i.e., buy in the so-called *spot-market*) by paying a "penalty" price g_k. Note that this last action makes the recourse complete, preventing from infeasible purchasing plans. On the contrary, it is not allowed to activate new contracts or to exclude any contracts with respect to those already decided in the first stage.

In our work, we keep the same above two-stage decomposition even if, for the sake of brevity, we do not report the two-stage model. However, we present a Deterministic Equivalent Problem (DEP) formulation slightly different to the one presented in Manerba et al. (2017a), that is more suitable for the solution methods we will propose. In particular, we use free variables to represent the variation on the purchased quantities decided as a recourse and explicit *non-anticipativity* constraints. In the DEP we consider a set S of scenarios to approximate the probability distribution of the stochastic variables $\widehat{d}_k$. More precisely, each scenario $s \in S$ is associated with a realization of the demand oscillation $\widehat{d}_k^s$ that occurs with probability p^s. Let us also define, for each scenario $s \in S$, the following variables:

- x_i^s := binary variable taking value 1 if a purchasing contract is activated with supplier $i \in M$ (and the corresponding activation cost is paid), and 0 otherwise;
- z_{ikr}^s := units of product $k \in K$ that we expect to purchase from supplier $i \in M_k$ in interval $r \in R_i$;
- y_{ir}^s := binary variable taking value 1 if the total products quantity we expect to purchase from supplier $i \in M$ lies in the discount interval $[l_{ir}, u_{ir}]$ with $r \in R_i$, and 0 otherwise;
- Z_{ikr}^s := variation in purchased quantity, with respect to the expectation z_{ikr}^s, of product $k \in K$ from supplier $i \in M_k$ in interval $r \in R_i$;
- W_k^s := quantity of product $k \in K$ that has to be purchased in the spot-market.

Then, the CTQD-AC$_{ud}$ problem can be stated as follows:

$$\min \sum_{s \in S} p^s \left[\sum_{i \in M} a_i x_i^s + \sum_{k \in K} \sum_{i \in M_k} \sum_{r \in R_i} (1 - \delta_{ir}) f_{ik} (z_{ikr}^s + Z_{ikr}^s) + \sum_{k \in K} g_k W_k^s \right] \tag{1}$$

$$\text{subject to} \qquad \sum_{i\in M_k}\sum_{r\in R_i} z^s_{ikr} \geq d_k \qquad k\in K, s\in S \tag{2}$$

$$\sum_{r\in R_i} z^s_{ikr} \leq q_{ik} \qquad k\in K, i\in M_k, s\in S \tag{3}$$

$$l_{ir}y^s_{ir} \leq \sum_{k\in K} z^s_{ikr} \leq u_{ir}y^s_{ir} \qquad i\in M, r\in R_i, s\in S \tag{4}$$

$$\sum_{r\in R_i} y^s_{ir} \leq x^s_i \qquad i\in M, s\in S \tag{5}$$

$$\sum_{i\in M_k}\sum_{r\in R_i} (z^s_{ikr} + Z^s_{ikr}) + W^s_k \geq d_k + \widehat{d}^s_k \qquad k\in K, s\in S \tag{6}$$

$$\sum_{r\in R_i} (z^s_{ikr} + Z^s_{ikr}) \leq q_{ik} \qquad k\in K, i\in M_k, s\in S \tag{7}$$

$$l_{ir}y^s_{ir} \leq \sum_{k\in K} (z^s_{ikr} + Z^s_{ikr}) \leq u_{ir}y^s_{ir} \qquad i\in M, r\in R_i, s\in S \tag{8}$$

$$z^s_{ikr} + Z^s_{ikr} \geq 0 \qquad k\in K, i\in M_k, r\in R_i, s\in S \tag{9}$$

$$x^{s_1}_i = x^{s_2}_i \qquad i\in M,\ s_1, s_2\in S \tag{10}$$

$$y^{s_1}_{ir} = y^{s_1}_{ir} \qquad i\in M, r\in R_i,\ s_1, s_2\in S \tag{11}$$

$$z^{s_1}_{ikr} = z^{s_2}_{ikr} \qquad k\in K, i\in M_k, r\in R_i,\ s_1, s_2\in S \tag{12}$$

$$x^s_i \in \{0,1\} \qquad i\in M, s\in S \tag{13}$$

$$y^s_{ir} \in \{0,1\} \qquad i\in M, r\in R_i, s\in S \tag{14}$$

$$z^s_{ikr} \geq 0 \qquad k\in K, i\in M_k, r\in R_i, s\in S \tag{15}$$

$$W^s_k \geq 0 \qquad k\in K, s\in S. \tag{16}$$

The objective function (1) establishes the minimization of the sum of activation and purchasing costs, weighted over all the scenarios by the scenario probability. Note that, the purchasing costs consider the expected purchased quantities, the actual variations, and the spot-market supplies. Constraints (2) ensure that the demand d_k for each product $k \in K$ is satisfied, whereas constraints (3) state that it is not possible to purchase from supplier i an amount of product k larger than the quantity available. Constraints (4) define interval bounds for each supplier. If interval r for supplier i is selected in scenario s ($y^s_{ir} = 1$), then the total amount purchased has to lie between the lower bound l_{ir} and the upper bound u_{ir}. On the contrary, the purchased quantity must be zero. Constraints (5) guarantee that at most one interval for each selected supplier is active, and that no intervals are active if the supplier is not selected. Constraints (6), (7), and (8) have the same meaning of the constraints (2), (3), and (4), respectively, but also consider the recourse decisions on the quantities purchased (Z-variables). Moreover, constraints (6) allow satisfying part of the product demand by using the spot market (W-variables). Constraints (9) simply deny purchasing a negative quantity of product (being Z_{ikr} a free variable) and, consequently, to change the discount interval chosen at the first-stage. Equations (10)–(12) are the non-anticipatity constraints forcing every scenario-based solutions to share the same first-stage decisions (i.e., to be implementable). Finally, binary and non-negativity conditions on variables appear in (13)–(16). Note that, the integrality of z variables is not explicitly declared in the formulation since a solution with this property always exists if demands, products availabilities, and lower/upper bounds are integral values.

4 A heuristic Progressive Hedging

Progressive Hedging (PH) is a decomposition-based algorithm proposed by Rockafellar and Wets (1991) for SP models. As is known, once explicitly defined a set of potential scenarios, these models results in a block-diagonal structure where each block corresponds to a single scenario second-stage problem and the linking (complicating) constraints and variables are those related to the first stage. Briefly, the PH first decomposes the problem over the scenarios by relaxing the complicating constraints in a Lagrangean fashion, then, at each iteration, calculates the optimal solutions of all the mono-scenario problems and evaluates if they involve the same first-stage decisions. Moreover, a non-necessarily feasible *temporarily global solution* (TGS) for the complete problem is also calculated by using some aggregation operators. The algorithm stops when a complete *consensus* on the first-stage decisions over all the scenarios is met (i.e., when the TGS becomes implementable), otherwise it adjusts the Lagrangean multipliers of the mono-scenario problems and iterates again. Unfortunately, the PH has been proved to converge to the optimal solution only in the case of continuous linear programs, hence using the same conceptual framework to tackle MILP problems (as first proposed in Løkketangen and Woodruff, 1996) may result in a heuristic approach. Over the years, several authors have applied PH-based heuristic algorithms to a wide variety of problems (see, e.g., Crainic et al., 2011, Watson and Woodruff, 2011, Veliz et al., 2015, Crainic et al., 2016, Perboli et al., 2017), often experimenting very good results. In the same spirit, we propose a PH-based heuristic solution approach for the CTQD-AC$_{ud}$ enhancing the standard implementation with several acceleration strategies.

Algorithm 1 presents the overall structure of our approach. The method starts by solving the Expected Value (EV) problem, i.e. the CTQD-AC$_{ud}$ where each stochastic variable is substituted by its deterministic expected value (Step 1). Since the EV problem is deterministic, we can find easily a temporary global solution $(\bar{x}^{(0)}, \bar{y}^{(0)}, \bar{z}^{(0)})$ to use as a first reference solution for the Augmented Lagrangean relaxation applied to decompose the CTQD-AC$_{ud}$ (Step 2). The details of this decomposition are presented in Section 4.1. Then, Step 3 initializes all the multipliers and penalties.

Steps 4-16 are those related to the core PH procedure. Basically, at each iteration t, all the mono-scenario subproblems mCTQD-AC$_{ud}$ coming from the decomposition are solved independently (Steps 7-9), then the TGS is calculated in Step 10 (see Section 4.2). If, in the current TGS, the consensus is met for all the variables, then the convergence procedure stops, otherwise the Lagrangean multipliers and penalties are updated (Section 4.3). As

Algorithm 1 Progressive Hedging-based heuristic.

1: Solve the EV problem of CTQD-AC$_{ud}$ to find $(\bar{x}^{(0)}, \bar{y}^{(0)}, \bar{z}^{(0)})$
2: Decompose the CTQD-AC$_{ud}$ by scenario using Augmented Lagrangean relaxation
3: Initialize the Lagrangean multipliers and penalties

4: $t \leftarrow 0$
5: **while** any termination criterion is not satisfied **do**
6:     $t \leftarrow t + 1$
7:     **for** each scenario $s \in S$ **do**
8:         Solve the corresponding mCTQD-AC$_{ud}$ subproblem
9:     **end for**
10:     Calculate $(\bar{x}^{(t)}, \bar{y}^{(t)}, \bar{z}^{(t)})$ by using the aggregation operators
11:     **if** consensus is met **then**
12:         **break**
13:     **else**
14:         Update the Lagrangean multipliers and penalties
15:     **end if**
16: **end while**

17: Optimally solve the model (2)–(16) by fixing variables for which the consensus is met

stated in Step 5, the method iterates until one of the implemented termination criteria is satisfied (Section 4.4).

At the end of the algorithm (Step 17), a final optimal procedure is run on the original model (2)–(16) reduced in complexity through a variable fixing, i.e. variables for which the consensus is met are fixed according to the TGS found in the last iteration of the PH. This final model can be either a pure LP problem (when the complete consensus has met) or still a MILP problem (when any termination criterion prematurely stops the convergence). In the latter case, however, the purpose is to have a small percentage of non-fixed first-stage variables and, in turn, a program much easier to solve than the original one.

4.1 Scenario decomposition

First, we obtain a model equivalent to (1)–(16) by substituting constraints (10)–(12) with

$$x_i^s = \bar{x}_i \qquad i \in M, s \in S \tag{17}$$

$$y_{ir}^s = \bar{y}_{ir} \qquad i \in M, r \in R_i, s \in S \tag{18}$$

$$z_{ikr}^s = \bar{z}_{ikr} \qquad k \in K, i \in M_k, r \in R_i, s \in S \tag{19}$$

where $\bar{x}_i \in \{0,1\}$ are the first-stage decisions on contract activation for each supplier $i \in M$, $\bar{y}_{ir} \in \{0,1\}$ are the first-stage decisions on interval selection for each $i \in M, r \in R_i$, and $\bar{z}_{ikr} \geq 0$ are the first-stage decisions on purchased quantities for each product $k \in K, i \in M_k, r \in R_i$. Then, following Rockafellar and Wets (1991), we apply a classical Lagrangean relaxation for the non-anticipatity constraints (19) and an Augmented Lagrangean technique to relax (17) and (18). This yields the objective function

$$\min \sum_{s\in S} p^s \Bigg(\sum_{i\in M} a_i x_i^s + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i}(1-\delta_{ir}) f_{ik}(z_{ikr}^s + Z_{ikr}^s) + \sum_{k\in K} g_k W_k^s + \\ + \sum_{i\in M} \lambda_i^s (x_i^s - \bar{x}_i) + \sum_{i\in M}\sum_{r\in R_i} \mu_{ir}^s (y_{ir}^s - \bar{y}_{ir}) + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i} \pi_{ikr}^s (z_{ikr}^s - \bar{z}_{ikr}) + \\ + \frac{1}{2}\sum_{i\in M} \rho_1 (x_i^s - \bar{x}_i)^2 + \frac{1}{2}\sum_{i\in M}\sum_{r\in R_i} \rho_2 (y_{ir}^s - \bar{y}_{ir})^2 \Bigg), \quad (20)$$

with Lagrangean multipliers $\lambda_i^s, i \in M, s \in S$, $\mu_{ir}^s, i \in M, r \in R_i, s \in S$, and $\pi_{ikr}^s, k \in K, i \in M_k, r \in R_i, s \in S$ for the relaxed constraints (17), (18), and (19), respectively, and penalty factors ρ_1 and ρ_2 for (17) and (18), respectively. Note that a quadratic expression is not considered to penalize the deviation of z-variables from the temporary global solution in order to maintain the linearity of the objective function with respect to the original variables. In fact, given the binary condition on x and y variables, (20) can be rewritten as

$$\min \sum_{s\in S} p^s \Bigg\{ \sum_{i\in M} \left[x_i^s \left(a_i + \lambda_i^s + \frac{\rho_1}{2} - \rho_1 \bar{x}_i \right) + \bar{x}_i \left(\frac{\rho_1}{2} - \lambda_i^s \right) \right] + \\ + \sum_{i\in M}\sum_{r\in R_i} \left[y_{ir}^s \left(\mu_{ir}^s + \frac{\rho_2}{2} - \rho_2 \bar{y}_{ir} \right) + \bar{y}_{ir} \left(\frac{\rho_2}{2} - \mu_{ir}^s \right) \right] + \\ + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i} \left[z_{ikr}^s \Big((1-\delta_{ir}) f_{ik} + \pi_{ikr}^s \Big) - \bar{z}_{ikr}\pi_{ikr}^s \right] + \\ + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i} (1-\delta_{ir}) f_{ik} Z_{ikr}^s + \sum_{k\in K} g_k W_k^s \Bigg\}. \quad (21)$$

The above relaxation makes the model scenario-separable and, for any given scenario $s \in S$, we obtain the following CTQD-AC$_{ud}$ problem with modified costs (mCTQD-AC$_{ud}$):

$$\min \sum_{i\in M} \left[x_i^s \left(a_i + \lambda_i^s + \frac{\rho_1}{2} - \rho_1 \bar{x}_i \right) + \bar{x}_i \left(\frac{\rho_1}{2} - \lambda_i^s \right) \right] + \\ + \sum_{i\in M}\sum_{r\in R_i} \left[y_{ir}^s \left(\mu_{ir}^s + \frac{\rho_2}{2} - \rho_2 \bar{y}_{ir} \right) + \bar{y}_{ir} \left(\frac{\rho_2}{2} - \mu_{ir}^s \right) \right] + \\ + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i} \left[z_{ikr}^s \Big((1-\delta_{ir}) f_{ik} + \pi_{ikr}^s \Big) - \bar{z}_{ikr}\pi_{ikr}^s \right] + \\ + \sum_{k\in K}\sum_{i\in M_k}\sum_{r\in R_i} (1-\delta_{ir}) f_{ik} Z_{ikr}^s + \sum_{k\in K} g_k W_k^s \quad (22)$$

subject to

$$\sum_{i \in M_k} \sum_{r \in R_i} z^s_{ikr} \geq d_k \qquad k \in K \tag{23}$$

$$\sum_{r \in R_i} z^s_{ikr} \leq q_{ik} \qquad k \in K, i \in M_k \tag{24}$$

$$l_{ir} y^s_{ir} \leq \sum_{k \in K} z^s_{ikr} \leq u_{ir} y^s_{ir} \qquad i \in M, r \in R_i \tag{25}$$

$$\sum_{r \in R_i} y^s_{ir} \leq x^s_i \qquad i \in M \tag{26}$$

$$\sum_{i \in M_k} \sum_{r \in R_i} (z^s_{ikr} + Z^s_{ikr}) + W^s_k \geq d_k + d^{s\Delta}_k \qquad k \in K \tag{27}$$

$$\sum_{r \in R_i} (z^s_{ikr} + Z^s_{ikr}) \leq q_{ik} \qquad k \in K, i \in M_k \tag{28}$$

$$l_{ir} y^s_{ir} \leq \sum_{k \in K} (z^s_{ikr} + Z^s_{ikr}) \leq u_{ir} y^s_{ir} \qquad i \in M, r \in R_i \tag{29}$$

$$z^s_{ikr} + Z^s_{ikr} \geq 0 \qquad k \in K, i \in M_k, r \in R_i \tag{30}$$

$$x^s_i \in \{0, 1\} \qquad i \in M \tag{31}$$

$$y^s_{ir} \in \{0, 1\} \qquad i \in M, r \in R_i \tag{32}$$

$$z^s_{ikr} \geq 0 \qquad k \in K, i \in M_k, r \in R_i \tag{33}$$

$$W^s_k \geq 0 \qquad k \in K. \tag{34}$$

Basically, the mCTQD-AC$_{ud}$ is a single-scenario CTQD-AC$_{ud}$ problem with a more complex (but still linear) objective function. This makes the mCTQD-AC$_{ud}$ as easy-to-solve (even if still $\mathcal{NP}$-hard) as the deterministic version of the CTQD-AC$_{ud}$, i.e. where the stochastic demand of each product is substituted by a deterministic value. Moreover, any solution method valid for the deterministic version of the problem can be applied with very little changes to solve the mCTQD-AC$_{ud}$. Hence, we have decided to use as a black-box solver for the mCTQD-AC$_{ud}$ the exact framework proposed in Manerba and Mansini (2012) that exploits some preprocessing routines, valid inequalities, and heuristic components.

4.2 Computation of the temporary global solution

The solutions of all the subproblems are used to build up a *temporary global solution* (TGS), i.e. the value of $(\bar{x}, \bar{y}, \bar{z})$ in a given iteration t of the algorithm. In particular, the classical expectation function is used as aggregation operator as follows:

$$\bar{x}^{(t)}_i = \sum_{s \in S} p^s x^{s(t)}_i \qquad i \in M, \tag{35}$$

$$\bar{y}^{(t)}_{ir} = \sum_{s \in S} p^s y^{s(t)}_{ir} \qquad i \in M, r \in R_i, \tag{36}$$

$$\bar{z}^{(t)}_{ikr} = \sum_{s \in S} p^s z^{s(t)}_{ikr} \qquad k \in K, i \in M_k, r \in R_i. \tag{37}$$

4.3 Penalties adjustment

At a given iteration t of the algorithm, let $\lambda_i^{s(t)}$, $\mu_{ir}^{s(t)}$, $\pi_{ikr}^{s(t)}$ be the Lagrangean multipliers and let $\rho_1^{(t)}$ and $\rho_2^{(t)}$ be the penalties. At the beginning of the procedure, $\rho_1^{(0)}$, $\rho_2^{(0)}$ and $\rho_3^{(0)}$ are initialized to a small positive value, whereas all $\lambda_i^{s(0)}$, $\mu_{ir}^{s(0)}$, $\pi_{ikr}^{s(0)}$ are initialized to 0. Then, at each PH iteration $t > 1$, the values of the multipliers and penalties are updated as follows:

$$\begin{aligned}
\lambda_i^{s(t)} &\leftarrow \lambda_i^{s(t-1)} + \rho_1^{(t-1)}(x_i^{s(t)} - \bar{x}_i^{(t)}), & \rho_1^{(t)} &\leftarrow \alpha\rho_1^{(t-1)},\\
\mu_{ir}^{s(t)} &\leftarrow \mu_{ir}^{s(t-1)} + \rho_2^{(t-1)}(y_{ir}^{s(t)} - \bar{y}_{ir}^{(t)}), & \rho_2^{(t)} &\leftarrow \beta\rho_2^{(t-1)},\\
\pi_{ikr}^{s(t)} &\leftarrow \pi_{ikr}^{s(t-1)} + \rho_3^{(t-1)}(z_{ikr}^{s(t)} - \bar{z}_{ikr}^{(t)}), & \rho_3^{(t)} &\leftarrow \gamma\rho_3^{(t-1)},
\end{aligned}$$

where α, β, and γ are constant factors strictly greater than 1. These factors influencing the increment of the penalties can be used to tune the algorithm for a better convergence.

4.4 Termination criteria

The PH naturally stops when the complete consensus is met, i.e. when constraints (17), (18), and (19) are completely satisfied. However, other classical criteria can be used to stop the PH convergence and start the finalization phase instead. In particular, we have decided to terminate the algorithm also when the maximum computational time (*maxTime*), the maximum number of iterations (*maxIter*), or the maximum number of iterations without any improvements in the percentage of variables that have reached consensus (*maxIter-WithoutImpr*) are exceeded. In particular, the last criterion seems necessary to detect and react to the possible cycling behavior of the PH when dealing with a non-continuous linear program.

5 Acceleration strategies

The basic PH framework has several drawbacks. First, it is not able to produce any feasible solutions until the complete consensus is met, i.e. until the very end of the procedure. Second, the convergence of the method may be very slow, deteriorating its effectiveness while the number of iterations increases. To partially overcome these issues, we have developed several acceleration strategies based on the model properties, discussed in the following.

5.1 Binary consensus

Let us denote as CTQD-AC$_{ud}(\tilde{x}, \tilde{y}, z, Z, W)$ the model (2)–(16) in which all the binary variables (i.e., $x_i, \forall i \in M$ and $y_{ir}, \forall i \in M, \forall r \in R_i$) have been fixed to a vector of known binary values $(\tilde{x}, \tilde{y})$. Despite the fact that the z variables logically belong to the first-stage decisions (together with x and y), in order to speed up the convergence we have decided to just look for the consensus of a restricted set of variables (i.e., the binary ones) and to complete the solution by solving a CTQD-AC$_{ud}(\tilde{x}, \tilde{y}, z, Z, W)$ at the end. The enormous gain in efficiency of such method has three main reasons:

1. the CTQD-AC$_{ud}(\tilde{x}, \tilde{y}, z, Z, W)$ is a pure LP problem since all the non-fixed variables are either continuous or free and thus easy-to-solve;
2. z variables are a great number and, in general, the more the variables, the more the iterations required to reach consensus among them;
3. evaluating the achievement of consensus for continuous variables is much difficult than for binary ones since an integrality check is not sufficient.

5.2 Premature stop of the exact solution for each subproblem

The solution time of each mono-scenario problem represents a clear bottleneck of the entire procedure, in particular because we are using a branch-and-cut exact method. Even if some authors have developed PH algorithms where each subproblem is solved by using specialized and very fast heuristics (see, e.g., Crainic et al., 2011), in our case we prefer to maintain the potentials of an exact framework but stopping it when a particular optimality gap is achieved or a certain CPU time is exceeded and returning the best solution found so far. Some preliminary computational tests have shown that setting as stopping rule an optimality gap threshold of 1% allows saving a great amount of time while maintaining a very good quality of the solutions.

5.3 Primal LP-based heuristic

In order to generate feasible solutions during (and not only at the end of) the PH procedure, we implement a primal heuristic based on the optimal solution of a linear program (Algorithm 2). The basic idea of this method, invoked at each iteration t after the calculation of the TGS, is to create a feasible and easy-to-solve LP problem by fixing, in model (2)–(16), all the binary variables (x, y) to some values $(\tilde{x}, \tilde{y})$. These values are chosen according to the current TGS $(\bar{x}^{(t)}, \bar{y}^{(t)}, \bar{z}^{(t)})$ through a simple rounding. More precisely, for each supplier $i \in M$, we round the current value of $x_i^{(t)}$ to the nearest integer (either 0 or 1). Then, if $\tilde{x}_i = 0$ (i.e., the supplier is not selected), we also set to 0 all the variables corresponding to the selection of its discount intervals ($\tilde{y}_{ir} = 0, \forall r \in R_i$). Otherwise, for each selected supplier ($\tilde{x}_i = 1$), we set to 1 only the $\tilde{y}_{i,r'}$ variable corresponding to the interval r' for which the TGS value is the maximum among the intervals. Note that this rounding always guarantees a feasible problem. Hence, the optimal solution of the resulting LP model can be stored if better than the incumbent one, without affecting the PH convergence.

Algorithm 2 LP-based primal heuristic at iteration t.

```
for each supplier i ∈ M do
    x̃_i ← ⌊x̄_i^(t) + 0.5⌋
    if x̃_i = 0 then
        for each discount interval r ∈ R_i do
            ỹ_ir ← 0
        end for
    else if x̃_i = 1 then
        Find r' such that ȳ_{i,r'}^(t) ≥ ȳ_ir^(t), ∀r ∈ R_i
        ỹ_{i,r'} ← 1
        ỹ_ir ← 0, ∀r ∈ R_i \ {r'}
    end if
end for
Optimally solve the CTQD-AC_ud(x̃, ỹ, z, Z, W)
```

5.4 Parallel implementation

The computational complexity of a PH algorithm clearly depends on the number of scenarios considered, since at each iteration $|S|$ mono-scenario subproblems have to be solved (see lines 6-8 of Algorithm 1). However, all these problems are completely independent and thus their relative solution procedures can be parallelized without affecting the correctness of the algorithm. Hence, we have implemented a parallel version of our PH (`pPH`) that allocates each subproblem to each logical CPU available on the machine. Note that this parallel implementation is a trade-off choice with respect to the basic procedure and does not ensure overall best performance. In fact, in the basic sequential PH (`sPH`), each mono-scenario subproblem is solved with a *black-box* procedure allowed to exploit the multi-threading features of the machine. Hence, `pPH` solves more subproblems simultaneously but using a potentially less powerful method. Since a threshold time is set for each subproblem resolution (see Section 5.2), then the solutions provided by the two methods may be different. For this reason, both versions of the PH will be tested and compared in the following section.

6 Computational experiments

This section is devoted to present the benchmark instances used to assess the performance of our computational approaches, along with the results and the analysis of the performed experiments. All these tests have been done on an *Intel(R) Core(TM) i7-5930K* CPU@3.50 GHz machine with 64 GB RAM and running *Windows 7* 64-bit operating system. We precise that such machine allows to use up to 12 logical CPUs for multi-threading computation.

6.1 Benchmark instances

The benchmark instances presented in Manerba et al. (2017a) for the and CTQD-AC$_{ud}$ are used to empirically assess the efficiency and the efficacy of our solution approaches. In order to allow the reader to fully understand the computational results on the different types of instances, we briefly recall their characteristics and summarize them in Table 1.

Parameter and values	$DP1$	$DP2$	Meaning
$f_k \sim \mathrm{U}(10, 200)$	-	-	basic price of product $k \in K$
$f_{ik} \sim \mathrm{U}(0.9f_k, 1.1f_k)$	-	-	price of product $k \in K$, $i \in M_k$
$q_{ik} \sim \mathrm{U}(1, 15)$	-	-	availability of product $k \in K$, $i \in M_k$
$d_k = \left\lceil \overline{d}_k - \left((\overline{d}_k - 1)\frac{f_k}{\max_{k\in K}\{f_k\}}\right)\right\rceil$, $\overline{d}_k = \left\lceil \lambda \max_{i\in M_k}\{q_{ik}\} + (1-\lambda)\sum_{i\in M_k} q_{ik}\right\rceil$, $\lambda \in [0, 1]$	-	-	expected demand of product $k \in K$
$a_i = \gamma^{AC}\sum_{k\in K}\frac{q_{ik}}{10}$, with $\gamma^{AC1} = \overline{f}$ and $\gamma^{AC2} = \left(\overline{f} + \frac{\overline{f}}{\overline{f}_i}\right)$, $\overline{f} = \sum_{i\in M}\overline{f}_i/\lvert S\rvert$, $\overline{f}_i = \sum_{k\in K} f_{ik}/\lvert K\rvert$	-	-	activation cost for supplier $i \in M$
$g_k = 1.2\max_{i\in M_k} f_{ik}$	-	-	spot-market price of product $k \in K$
$R_i = \{1, \dots, r_i\}$	$r_i = \{3, 4, 5\}$	$r_i = 3$	set of intervals for supplier $i \in M$
$l_{ir} = \lfloor \alpha_{ir}\sum_{k\in K} q_{ik}\rfloor$	$\alpha_{ir} \sim \mathrm{U}(0.6, 1)$	$\alpha_{i1} = 0$, $\alpha_{i2} = 0.7$, $\alpha_{i3} = 0.9$	LB of interval $r \in R_i$, $i \in M$
$u_{ir} = l_{i,r+1} - 1$	-	-	UB of interval $r \in R_i \setminus \{r_i\}$, $i \in M$
$u_{i,r_i} = \sum_{k\in K} q_{ik}$	-	-	UB of interval r_i,$i \in M$
δ_{ir}	$\sim \mathrm{U}(0.01, 0.05)$	$\delta_{i1} = 0.01$, $\delta_{i2} = 0.02$, $\delta_{i3} = 0.03$	discount rate of interval $r \in R_i$, $i \in M$

Table 1: Instances' general parameters

Each product $k \in K$ is assigned to a random subset M_k of suppliers, and to a basic price f_k, randomly chosen in $[10, 200]$. Then, for each product k sold by supplier i, the available quantity q_{ik} is generated uniformly in $[1, 15]$ whereas the product price f_{ik} is randomly generated in $[0.9f_k, 1.1f_k]$. Moreover, in order to discourage external supplies, the penalty price g_k is generated for each product $k \in K$ such that $g_k := 1.2\max_{i\in M_k} f_{ik}$. The deterministic demand d_k, for each product k, is generated by taking into account the average expensiveness of a product and its average availability among the suppliers. In particular, the more expensive the product, the lower the corresponding demand. Moreover, the relation between the demand and the overall availability of a product, represented by a parameter λ varying in $[0, 1]$, allows to better control the number of suppliers needed for a feasible solution. More precisely, the lower the value of λ, the higher the number of suppliers required to satisfy the entire demand. Finally, two types of activation costs (AC) have been generated. In *AC1*, the activation cost of a supplier $i \in M$ is proportional to the total quantity of products available in i multiplied by the global average product price, whereas in *AC2* activation costs are inversely proportional to the supplier's average product price.

Concerning the total quantity discount policy, for each supplier $i \in M$ and each interval $r \in R_i$, lower bounds are generated as an a priori percentage $0 \leq \alpha_{ir} < 1$ of the total amount of products available from a specific supplier, i.e. $l_{ir} = \lfloor \alpha_{ir} \sum_{k \in K} q_{ik} \rfloor$. Upper bounds u_{ir} for interval $r \in R \setminus \{r_i\}$ of supplier $i \in M$ are set to the value corresponding to the lower bound of the following interval minus 1, whereas $u_{i,r_i} = \sum_{k \in K} q_{ik}, i \in M$. Two different discount policy structures (DP) have been generated. In *DP1*, the number of discount intervals for each supplier $i \in M$ ranges randomly between 3 and 5, with a random interval width but ensuring $\alpha_{i,2} \geq 0.6$. Discount rates applied in all intervals are randomly generated such that $\delta_{i,r_i} \leq 0.05$ and $\delta_{ir} \leq \delta_{i,r+1}$, $r \in R \setminus \{r_i\}$. In *DP2*, instead, the number of intervals is equal to 3 for all the suppliers, with $\alpha_{i,1} = 0$, $\alpha_{i,2} = 0.7$ and $\alpha_{i,3} = 0.9$, $i \in M$. Moreover, discount rates are fixed for all the suppliers and equal to 1%, 2%, and 3%, respectively.

Hence, the complete benchmark dataset is composed by 72 deterministic instances, one for each combination of $\{5, 10, 20\}$ suppliers, $\{10, 20, 30\}$ products, the two types of discount policy structure (*DP1* or *DP2*), the two types of activation cost (*AC1* or *AC2*), and two quite extreme values for parameter λ (i.e., 0.1 or 0.8).

Finally, for any given deterministic instance, the stochastic demand values are generated by using different probability distributions. More precisely, for each scenario $s \in S$, the stochastic demand d_k^s of each product $k \in K$ is drawn according to a *Uniform* or a *Gumbel* probability distribution in $[0.5d_k, 2d_k]$ (i.e., the demand d_k may be halved or doubled at most). Then, its oscillation $\widehat{d_k^s} := d_k^s - d_k$ is simply calculated. Through an in-sample stability analysis, it has been also shown how considering 100 scenarios is sufficient to maintain under the 1% threshold (which seems a reasonable precision) the percentage ratio between the standard deviation and the mean of the optimal objective values of any instance over ten random and independent stochastic data generations.

6.2 A Benders algorithm

Manerba et al. (2017a) have highlighted the unsuitability of using the state-of-the-art Cplex's branch-and-cut based MILP solver (called hereafter `Cplex-B&C`) to cope with the CTQD-AC$_{ud}$ when considering a sufficiently large number of scenarios. Therefore, they developed (by exploiting valid inequalities, preprocessing routines, and a heuristic upper-bound) an improved solution framework, called hereafter `MMP`, that outperformed `Cplex-B&C`.

However, Cplex has been recently improved[1] by the introduction of a procedure based on the classical *Benders decomposition*. The algorithm actually mixes the effectiveness of a standard branch-and-cut approach with the generation of both optimality and feasibility cuts deriving from the specific decomposition (Benders, 1962). Despite the aging of such a method, the cutting generation has been implemented by using hints and improvements proposed in many successive works on the subject (McDaniel and Devine, 1977, Fischetti et al., 2010, Fischetti et al., 2016). Since this algorithm (called hereafter `Cplex-Benders`) is particularly aimed at solving SP problems, we have decided to test this new available approach and compare its performances with the `Cplex-B&C`'s and `MMP`'s ones. Table 2 shows this comparison on a subset of small and medium size instances from the benchmark set when considering the *Uniform* distribution for the stochastic demand. For each considered instance (identified by a combination of $|M|$, $|K|$, and λ values), and for each exact method, we report the CPU time in seconds needed to prove the optimality (t), the *time-to-best* (ttb), i.e. the CPU time in seconds needed to find the optimal solution, and the dimension in nodes of the branch-and-bound tree (BBn).

			`Cplex-B&C`			`MMP`			`Cplex-Benders`		
$\lvert M\rvert$	$\lvert K\rvert$	λ	t	ttb	BBn	t	ttb	BBn	t	ttb	BBn
5	10	0.1	1193	1029	934	63	23	15	10	6	71
5	10	0.8	5092	4756	1210	81	81	25	15	12	42
5	20	0.1	72348	64168	7380	738	542	62	57	52	393
5	20	0.8	4546	4012	1171	202	182	27	24	21	65
5	30	0.1	32455	24530	3750	1049	945	33	72	71	192
5	30	0.8	19113	18081	3250	844	819	27	55	54	83
10	10	0.1	154458	54805	15076	4183	4140	678	3801	1286	48029
10	10	0.8	154295	149507	8595	4160	3761	800	448	351	19490
		avg:	55437.5	40111.0	5170.8	1415.1	1311.7	208.4	560.3	231.6	8545.6

Table 2: `Cplex-Bender` vs `Cplex-B&C` and `MMP`

Concerning both the CPU times, `Cplex-Benders` outperforms `MMP` for all the instances. More precisely, on average, the total CPU time is more than halved and the best solution is found almost six times faster. Strangely, `Cplex-Benders` also creates and explores much greater branch-and-bound trees, thus requiring more memory. Since similar trends emerged also on the same instances when considering the *Gumbel* distribution, we will use `Cplex-Benders` as a benchmark to evaluate the performance of our PH algorithms in the following. We remark that the solver, if not forced to use the Benders algorithm, seems not able to detect the particular structure of the given problem and to choose the best solving procedure. Additional knowledge on the specific two-stage decomposition must be also given in order to let the solver to create the Benders cuts.

[1] See release notes of version 12.7.0 (IBM Knowledge Center, 2017)

6.3 Main results and analysis

In the following, we compare (on the complete set of benchmark instances) the sequential and the parallel version of our PH-based heuristic (`sPH` and `pPH`, respectively) with the Benders algorithm described in Section 6.2 (`Cplex-Benders`) in terms of efficiency and quality of the solution obtained. In particular, Tables 3–5 present this comparison for the instances in which the stochastic demand follows a Uniform distribution, whereas Tables 6–8 show the same results considering the Gumbel distribution. Each table concerns instances with the same number of suppliers and shows, for each instance and for each solution method, the computational time (t) and the time-to-best (ttb), both in seconds. Moreover, for `Cplex-Benders`, the percentage gap between the value of the best solution and the best lower bound found in the branch-and-cut tree ($gap\%$) is reported, whereas for the two PH versions, the percentage error with respect to the best solution found by `Cplex-Benders` is calculated (a negative value means that the relative PH algorithm has found a better solution with respect to Cplex). We precise that all the methods have an overall time limit of 14400 seconds (i.e., 4 hours). Moreover, for the PH algorithms, we have set *maxTime* to 7200 seconds, *maxIter* to 15 and *maxIterWithoutImpr* to 3.

					Cplex-Benders			sPH			pPH		
$\lvert M\rvert$	$\lvert K\rvert$	λ	DP	AC	$gap\%$	t	ttb	$\Delta\%$	t	ttb	$\Delta\%$	t	ttb
5	10	0.1	1	1	0.00	10	6	0.00	54	12	0.00	7	2
5	10	0.1	1	2	0.00	9	9	0.00	51	11	0.00	8	8
5	10	0.1	2	1	0.00	5	3	0.00	30	30	0.00	5	5
5	10	0.1	2	2	0.00	8	6	0.00	64	64	0.00	9	9
5	10	0.8	1	1	0.00	15	12	0.00	76	76	0.00	11	3
5	10	0.8	1	2	0.00	7	5	0.00	46	10	0.00	6	6
5	10	0.8	2	1	0.00	5	3	0.00	7	7	0.00	3	1
5	10	0.8	2	2	0.00	7	6	0.00	39	11	0.00	5	5
5	20	0.1	1	1	0.00	57	52	0.00	103	18	0.21	36	36
5	20	0.1	1	2	0.00	42	40	0.00	122	122	0.00	44	44
5	20	0.1	2	1	0.00	25	24	0.00	84	16	0.00	16	4
5	20	0.1	2	2	0.00	29	28	0.00	71	15	0.00	16	4
5	20	0.8	1	1	0.00	24	21	0.00	86	16	0.00	15	4
5	20	0.8	1	2	0.00	38	38	0.00	81	17	0.16	34	7
5	20	0.8	2	1	0.00	22	22	0.00	36	13	0.00	8	2
5	20	0.8	2	2	0.00	14	9	0.00	44	10	0.00	6	1
5	30	0.1	1	1	0.00	72	71	0.00	125	20	0.14	33	7
5	30	0.1	1	2	0.00	69	69	0.00	113	113	0.00	41	41
5	30	0.1	2	1	0.00	41	39	0.00	118	17	0.00	33	33
5	30	0.1	2	2	0.00	50	48	0.00	103	17	0.00	32	14
5	30	0.8	1	1	0.00	55	54	0.00	115	20	0.33	40	9
5	30	0.8	1	2	0.00	75	74	0.00	43	22	0.28	63	12
5	30	0.8	2	1	0.00	32	32	0.00	66	66	0.00	18	18
5	30	0.8	2	2	0.00	28	13	0.00	23	12	0.00	7	2
				avg:	0.00	31	28	0.00	71	31	0.05	21	12

Table 3: `Cplex-Benders` vs `sPH` and `pPH` for CTQD-AC$_{ud}$ instances with $|M| = 5$ (*Uniform* distribution).

We first look at instances with uniformly distributed demands (Tables 3–5). Benders algorithm shows quite good results for small/medium size instances ($|M| = 5$ and $|M| = 10$),

					Cplex-Benders			sPH			pPH		
$\|M\|$	$\|K\|$	λ	DP	AC	$gap\%$	t	ttb	$\Delta\%$	t	ttb	$\Delta\%$	t	ttb
10	10	0.1	1	1	0.00	3801	1286	0.48	206	206	0.95	45	45
10	10	0.1	1	2	0.00	163	158	0.02	170	146	0.04	39	31
10	10	0.1	2	1	0.00	58	46	0.08	83	75	0.11	10	7
10	10	0.1	2	2	0.00	85	48	0.03	97	67	0.03	15	13
10	10	0.8	1	1	0.00	448	351	0.00	293	293	0.00	81	81
10	10	0.8	1	2	0.00	54	52	0.00	106	39	0.29	36	36
10	10	0.8	2	1	0.00	22	22	0.00	72	72	0.16	11	7
10	10	0.8	2	2	0.00	25	22	0.00	92	51	0.00	21	13
10	20	0.1	1	1	0.00	3576	1841	0.44	375	139	0.05	153	147
10	20	0.1	1	2	0.00	1434	460	0.02	473	473	0.15	142	29
10	20	0.1	2	1	0.00	863	581	0.00	200	186	0.00	50	43
10	20	0.1	2	2	0.00	293	97	0.00	163	163	0.00	45	45
10	20	0.8	1	1	0.00	427	242	0.18	325	117	0.00	144	144
10	20	0.8	1	2	0.00	91	88	0.00	186	34	0.00	69	63
10	20	0.8	2	1	0.00	101	101	0.00	132	132	0.00	34	34
10	20	0.8	2	2	0.00	63	61	0.00	119	26	0.00	24	6
10	30	0.1	1	1	0.00	5800	2789	0.01	438	438	0.06	196	70
10	30	0.1	1	2	0.00	5438	3408	0.00	454	98	0.00	175	36
10	30	0.1	2	1	0.00	639	385	0.00	283	283	0.00	94	94
10	30	0.1	2	2	0.00	850	570	0.00	224	43	0.00	84	84
10	30	0.8	1	1	0.00	569	286	0.00	646	113	0.00	314	66
10	30	0.8	1	2	0.00	412	326	0.00	434	89	0.00	132	34
10	30	0.8	2	1	0.00	191	173	0.00	194	36	0.00	56	14
10	30	0.8	2	2	0.00	219	150	0.01	266	266	0.01	99	99
				avg:	0.00	1067	564	0.05	251	149	0.08	86	52

Table 4: `Cplex-Benders` vs `sPH` and `pPH` for CTQD-AC$_{ud}$ instances with $|M| = 10$ (*Uniform* distribution).

finding the optimal solution in all the cases. Computational times are on average around 30 seconds for $|M| = 5$ instances, and around 18 minutes for the $|M| = 10$ ones. However, performances drastically break down on $|M| = 20$ instances. Only 6 out of 24 instances are solved to optimality while, in the remaining cases, the best solution found have more than the 3% of optimality gap on average (with some picks around 6%). The time limit is reached in the most cases and the best solutions are found on average after more than 2 hours.

Our PH algorithms are more than competitive for the smallest instances and totally outperforms `Cplex-Benders` for the largest ones. As expected, the parallel PH is faster on average (both in terms of total computational time and in time-to-best) with respect to the sequential version, whereas, on the contrary, the latter method finds on average slightly better solutions. Anyway, since percentage errors are basically negligible, the quality of both the PH algorithms solutions is excellent. On average, even only considering the largest instances ($|M| = 20$), the percentage error is 0.05% and 0.15% for `sPH` and `pPH`, respectively. The 1% error is exceeded only two times by `pPH`, and never by `sPH`. Moreover, `sPH` and `pPH` are able to find the optimal solution 42 and 34 times out of 72, respectively, and a better solution (with respect to Cplex) on 9 and 10 instances out of the 18 non-closed ones, respectively. Given that the solution quality of our heuristic PH algorithms is actually comparable to that of an exact method, it is interesting to note that they are

					Cplex-Benders			sPH			pPH		
$\|M\|$	$\|K\|$	λ	DP	AC	$gap\%$	t	ttb	$\Delta\%$	t	ttb	$\Delta\%$	t	ttb
20	10	0.1	1	1	3.83	14400	13733	0.04	680	400	0.04	168	103
20	10	0.1	1	2	3.70	14400	8917	0.18	641	615	0.10	160	152
20	10	0.1	2	1	0.00	140	130	0.00	217	160	0.04	42	25
20	10	0.1	2	2	2.39	14400	1313	-0.25	267	267	-0.19	51	51
20	10	0.8	1	1	0.00	785	770	0.91	613	595	0.91	148	145
20	10	0.8	1	2	0.00	11183	2847	0.85	1117	1105	1.27	372	370
20	10	0.8	2	1	0.00	1715	726	0.14	443	240	1.46	112	44
20	10	0.8	2	2	0.00	880	876	0.00	536	536	0.92	115	115
20	20	0.1	1	1	2.36	14400	2876	-0.01	444	410	0.01	139	114
20	20	0.1	1	2	5.00	14400	13421	0.01	1399	1399	-0.10	488	325
20	20	0.1	2	1	3.90	14400	13096	0.04	649	615	0.18	193	193
20	20	0.1	2	2	3.43	14400	8380	0.19	648	525	0.21	151	151
20	20	0.8	1	1	6.62	14400	13331	-0.53	6021	3617	-0.53	4899	4204
20	20	0.8	1	2	2.86	14400	2733	0.11	3449	3419	0.11	1977	1971
20	20	0.8	2	1	4.64	14400	13676	0.01	1642	1594	-0.19	762	753
20	20	0.8	2	2	0.00	1297	1064	0.00	571	237	0.00	112	24
20	30	0.1	1	1	4.18	14400	10781	-0.04	2938	2849	-0.08	1220	440
20	30	0.1	1	2	4.93	14400	13604	-0.05	3645	3432	-0.11	1666	1666
20	30	0.1	2	1	4.44	14400	13378	0.02	2085	2085	0.05	633	633
20	30	0.1	2	2	3.90	14400	4774	-0.11	1288	1215	-0.12	399	81
20	30	0.8	1	1	5.63	14400	14223	-0.12	13387	13387	-0.12	11794	3742
20	30	0.8	1	2	6.30	14400	12095	-0.06	11715	11715	-0.06	13115	13115
20	30	0.8	2	1	3.64	14400	13198	0.00	2261	2261	0.00	1035	1035
20	30	0.8	2	2	3.36	14400	13786	-0.08	2022	2001	-0.08	744	744
				avg:	3.13	11467	8072	0.05	2445	2278	0.15	1687	1258

Table 5: `Cplex-Benders` vs `sPH` and `pPH` for CTQD-AC$_{ud}$ instances with $|M| = 20$ (*Uniform* distribution).

several times faster than Cplex in terms of overall convergence CPU time and time-to-best. The only exception is represented by the $|M| = 5$ instances, where `sPH` is 30 seconds slower on average, whereas for the remaining instances `sPH` and `pPH` are about 4 times and 10 times faster than Cplex, respectively.

Concerning the Gumbel distribution for the demand (Tables 6–8), we find basically the same trends and proportions among the three methods' performances in terms of quality and CPU time. This just reinforces the strength of our algorithms in solving such types of problems. In this case, the solutions found by the PH algorithms are much closer on average to those found by Cplex within 4 hours (always under the 0.05%), even if this set of instances seems a little bit easier to solve. In fact, for $|M| = 20$ instances, Cplex is able to guarantee solutions within the 2% of optimality gap on average, with some picks around 4.5%.

For the sake of completeness, we summarize in Table 9 and 10 some interesting details of the two PH versions developed and tested. Results are averaged over all the instances with the same number of suppliers. The column headers have the following meaning: t_I, t_H, and t_F are the CPU times in seconds dedicated by the algorithms to find an initial feasible solution (the EEV solution), to apply the primal heuristic during the search (see Section 5.3), and to optimally solve the final model with all or part of the binary variables fixed (line 17 of Algorithm 1), respectively; $cons\%$ is the percentage number of binary variables,

					Cplex-Benders			sPH			pPH		
$\|M\|$	$\|K\|$	λ	DP	AC	*gap%*	*t*	*ttb*	Δ%	*t*	*ttb*	Δ%	*t*	*ttb*
5	10	0.1	1	1	0.00	10	6	0.00	44	11	0.00	7	2
5	10	0.1	1	2	0.00	10	9	0.00	45	10	0.00	6	2
5	10	0.1	2	1	0.00	4	3	0.00	23	8	0.00	4	4
5	10	0.1	2	2	0.00	6	6	0.00	62	62	0.00	9	9
5	10	0.8	1	1	0.00	15	13	0.00	35	12	0.00	9	3
5	10	0.8	1	2	0.00	6	4	0.00	38	38	0.00	6	6
5	10	0.8	2	1	0.00	4	3	0.00	7	7	0.00	1	1
5	10	0.8	2	2	0.00	8	4	0.00	38	38	0.17	6	2
5	20	0.1	1	1	0.00	47	44	0.00	99	17	0.18	33	12
5	20	0.1	1	2	0.00	40	38	0.00	114	114	0.00	40	7
5	20	0.1	2	1	0.00	23	21	0.00	85	85	0.00	16	5
5	20	0.1	2	2	0.00	23	23	0.00	59	16	0.00	20	20
5	20	0.8	1	1	0.00	21	13	0.00	75	16	0.00	13	3
5	20	0.8	1	2	0.00	29	25	0.00	79	16	0.11	30	7
5	20	0.8	2	1	0.00	18	12	0.00	12	12	0.00	3	2
5	20	0.8	2	2	0.00	11	6	0.00	8	8	0.00	2	1
5	30	0.1	1	1	0.00	71	69	0.00	36	19	0.16	27	6
5	30	0.1	1	2	0.00	61	56	0.00	87	51	0.00	40	7
5	30	0.1	2	1	0.00	41	35	0.00	70	17	0.00	24	24
5	30	0.1	2	2	0.00	53	52	0.00	79	17	0.13	28	28
5	30	0.8	1	1	0.00	58	58	0.00	124	57	0.19	39	10
5	30	0.8	1	2	0.00	61	61	0.00	58	21	0.00	58	58
5	30	0.8	2	1	0.00	26	26	0.00	77	17	0.00	18	18
5	30	0.8	2	2	0.00	21	13	0.00	10	0	0.00	3	0
					0.00	28	25	0.00	57	28	0.04	18	10

Table 6: Cplex-Benders vs sPH and pPH for CTQD-AC$_{ud}$ instances with $|M| = 5$ (*Gumbel* distribution).

out of the totality, that have reached the consensus when the PH procedure stops; *it* and *itb* are the total number of iterations done by the PH and *iteration-to-best* (i.e. the number of iterations needed to find the best solution), respectively.

6.4 Economic analysis of the stochastic solution

An extensive analysis of the convenience of explicitly considering a Stochastic Programming formulation for the CTQD-AC$_{ud}$ (with respect to using approximated values for its uncertain data) has been performed in Manerba et al. (2017a). In particular, for each deterministic instance and the two probability distributions, they compute two well-known stochastic programming measures (Birge and Louveaux, 1997), i.e. the *Value of the Stochastic Solution* (VSS) and the *Expected Value of Perfect Information* (EVPI). More precisely, VSS=EEV-RP and EVPI=RP-WS, where RP is the objective value of the DEP solution (*recourse problem solution*), EEV is the solution value of the stochastic model with the first-stage decision fixed by solving the deterministic problem using expected values for approximating the random parameters (*expected value solution*), and WS is the solution value of a problem in which it is assumed to know at the first-stage the realizations of all the stochastic variables (*wait-and-see solution*). Results have shown quite high VSS values on average and also some very high picks (18 times out of the total instances, the

					Cplex-Benders			sPH			pPH		
$\|M\|$	$\|K\|$	λ	DP	AC	*gap%*	*t*	*ttb*	$\Delta\%$	*t*	*ttb*	$\Delta\%$	*t*	*ttb*
10	10	0.1	1	1	0.00	998	973	0.22	191	37	0.28	42	17
10	10	0.1	1	2	0.00	93	91	0.00	158	150	0.36	36	8
10	10	0.1	2	1	0.00	28	26	0.00	69	15	0.00	10	2
10	10	0.1	2	2	0.00	40	39	0.00	83	17	0.09	13	3
10	10	0.8	1	1	0.00	122	112	0.00	228	196	0.00	83	75
10	10	0.8	1	2	0.00	32	32	0.00	95	19	0.00	36	34
10	10	0.8	2	1	0.00	28	23	0.00	55	13	0.00	7	2
10	10	0.8	2	2	0.00	19	17	0.08	89	50	0.00	18	14
10	20	0.1	1	1	0.00	2867	466	0.00	407	218	0.14	181	101
10	20	0.1	1	2	0.00	646	523	0.00	376	139	0.14	120	28
10	20	0.1	2	1	0.00	362	178	0.00	194	37	0.00	54	29
10	20	0.1	2	2	0.00	188	137	0.00	158	158	0.00	48	11
10	20	0.8	1	1	0.00	213	210	0.00	256	93	0.00	120	50
10	20	0.8	1	2	0.00	81	77	0.00	169	169	0.00	72	72
10	20	0.8	2	1	0.00	66	64	0.00	144	27	0.00	25	8
10	20	0.8	2	2	0.00	46	32	0.00	92	21	0.00	21	8
10	30	0.1	1	1	0.00	3840	1951	0.01	469	195	0.09	219	45
10	30	0.1	1	2	0.00	3275	840	0.01	439	111	0.01	168	39
10	30	0.1	2	1	0.00	432	240	0.00	256	256	0.00	87	87
10	30	0.1	2	2	0.00	629	431	0.00	232	45	0.00	88	88
10	30	0.8	1	1	0.00	418	386	0.00	133	72	0.00	196	41
10	30	0.8	1	2	0.00	241	234	0.00	70	70	0.00	28	28
10	30	0.8	2	1	0.00	170	168	0.00	142	34	0.00	46	12
10	30	0.8	2	2	0.00	177	103	0.00	230	1	0.00	74	1
					0.00	625	306	0.01	197	89	0.05	75	33

Table 7: `Cplex-Benders` vs `sPH` and `pPH` for CTQD-AC$_{ud}$ instances with $|M| = 10$ (*Gumbel* distribution).

VSS exceeds the 30% of the solution value), demonstrating the importance of SP models in place for the CTQD-AC$_{ud}$. As already explained in the Introduction, this is one of the motivations supporting the present work. However, due to the computational burden of solving the CTQD-AC$_{ud}$ when considering 100 scenarios, a consistent part of the biggest instances (i.e., the ones with $|M| = 20$ and $|K| = \{20, 30\}$) have not been solved to optimality, thus no VSS or EVPI values are available for them. Thanks to the algorithms developed in this work, we are now able to complete such analysis. Table 11 shows, for each deterministic instance (uniquely identified by $|M|$, $|K|$, λ, DP, and AC parameters) and for each considered probability distribution, the percentage values of VSS and EVPI with respect to the objective value of the recourse problem solution, i.e., VSS%=100*VSS/RP and EVPI%=100*EVPI/RP.

We can see that the VSS% values are quite consistent, independently from the number of products considered and the discounts or activation costs characteristics. Concerning the Uniform distribution, the VSS% is almost 12% on average with some picks around 25%. Remarkably, one-third of the instances have a VSS% exceeding the 15%. Concerning the Gumbel distribution, the average VSS% is 3 percentage points less while some picks achieve the 14-15%. The EVPI% values for both the distributions are similar and on average around the 2.5%. All these results are in line with those obtained for the other benchmark instances, confirming the economic importance of explicitly considering the stochasticity into a model

					Cplex-Benders			sPH			pPH		
$\|M\|$	$\|K\|$	λ	DP	AC	*gap%*	*t*	*ttb*	$\Delta\%$	*t*	*ttb*	$\Delta\%$	*t*	*ttb*
20	10	0.1	1	1	1.90	14400	9869	-0.05	725	725	-0.05	170	44
20	10	0.1	1	2	1.98	14400	1560	0.47	601	601	-0.14	146	140
20	10	0.1	2	1	0.00	55	48	0.10	148	30	0.10	38	9
20	10	0.1	2	2	0.58	14400	8697	0.09	275	248	0.41	50	44
20	10	0.8	1	1	0.00	250	182	0.04	562	121	0.04	121	28
20	10	0.8	1	2	0.00	1608	953	0.20	1296	1296	0.20	412	412
20	10	0.8	2	1	0.00	314	296	0.00	348	348	0.00	77	36
20	10	0.8	2	2	0.00	245	201	0.18	653	653	0.08	174	174
20	20	0.1	1	1	2.25	14400	8065	0.06	463	281	0.06	152	114
20	20	0.1	1	2	3.79	14400	13277	0.10	1432	1432	0.10	538	538
20	20	0.1	2	1	3.08	14400	13874	0.01	627	627	0.01	201	201
20	20	0.1	2	2	2.40	14400	13987	0.11	586	586	-0.04	151	151
20	20	0.8	1	1	4.33	14400	13238	0.00	7011	6714	0.00	5692	5689
20	20	0.8	1	2	0.00	7457	5258	0.00	4102	4084	0.00	2422	2422
20	20	0.8	2	1	0.83	14400	9498	0.00	2003	1969	0.00	993	661
20	20	0.8	2	2	0.00	615	474	0.00	524	236	0.00	119	119
20	30	0.1	1	1	3.92	14400	9104	-0.19	3792	3792	-0.35	1590	1554
20	30	0.1	1	2	3.94	14400	13150	-0.16	4171	4171	-0.24	1738	1738
20	30	0.1	2	1	3.93	14400	13378	0.03	2758	2758	0.05	1011	1011
20	30	0.1	2	2	3.11	14400	13277	0.25	1485	1485	-0.01	498	479
20	30	0.8	1	1	3.63	14400	13350	-0.10	6244	1456	-0.11	3606	3606
20	30	0.8	1	2	4.49	14400	13136	-0.10	6876	1566	-0.10	4875	1182
20	30	0.8	2	1	1.08	14400	1905	0.00	2471	2420	0.00	971	521
20	30	0.8	2	2	0.00	5775	4820	0.00	1693	772	0.00	451	233
					1.89	10280	7567	0.04	2119	1599	0.00	1091	879

Table 8: Cplex-Benders vs sPH and pPH for CTQD-AC$_{ud}$ instances with $|M| = 20$ (*Gumbel* distribution).

	sPH						pPH					
$\|M\|$	t_I	t_H	t_F	*cons%*	*it*	*itb*	t_I	t_H	t_F	*cons%*	*it*	*itb*
5	0.4	2.6	0.8	94	6	3	0.4	3.1	1.4	91	8	4
10	0.8	7.0	1.4	95	10	6	0.8	7.2	1.5	95	10	6
20	2.1	16.2	528.0	92	11	7	2.1	16.3	566.1	93	11	7
avg:	1.1	8.5	172.0	94	9	5	1.1	8.7	184.6	93	9	6

Table 9: sPH and pPH details (*Uniform* distribution).

	sPH						pPH					
$\|M\|$	t_I	t_H	t_F	*cons%*	*it*	*itb*	t_I	t_H	t_F	*cons%*	*it*	*itb*
5	0.4	1.8	0.8	94	5	3	0.4	2.8	0.9	91	7	3
10	0.8	5.4	1.7	95	8	3	0.8	6.6	1.3	95	9	3
20	2.2	13.9	2.7	97	11	8	2.2	14.4	3.3	96	11	7
avg:	1.1	6.9	1.7	95	8	5	1.1	7.8	1.8	94	9	5

Table 10: sPH and pPH details (*Gumbel* distribution).

for the CTQD-AC$_{ud}$ and, in turn, the importance of having in place algorithms to efficiently solve such model for realistic instances.

Instance					Uniform distribution		Gumbel distribution	
$\|M\|$	$\|K\|$	λ	DP	AC	VSS%	EVPI%	VSS%	EVPI%
20	10	0.1	1	1	17.5	2.1	14.5	1.5
20	10	0.1	1	2	10.3	2.3	6.8	1.7
20	10	0.1	2	1	8.0	4.5	11.0	4.9
20	10	0.1	2	2	25.2	2.2	15.2	2.3
20	10	0.8	1	1	10.2	3.1	6.2	2.3
20	10	0.8	1	2	9.1	3.7	10.4	3.0
20	10	0.8	2	1	16.2	4.1	12.5	3.9
20	10	0.8	2	2	21.2	5.5	7.4	7.0
20	20	0.1	1	1	12.0	2.2	8.8	2.9
20	20	0.1	1	2	15.2	1.4	10.5	1.5
20	20	0.1	2	1	19.1	1.6	13.7	1.5
20	20	0.1	2	2	3.9	1.3	1.2	1.2
20	20	0.8	1	1	10.3	1.5	0.9	1.0
20	20	0.8	1	2	13.5	3.3	6.4	3.9
20	20	0.8	2	1	15.9	2.0	14.3	2.4
20	20	0.8	2	2	5.8	0.8	9.4	1.1
20	30	0.1	1	1	7.0	1.1	6.5	1.4
20	30	0.1	1	2	9.7	1.0	7.6	0.6
20	30	0.1	2	1	7.0	0.9	7.2	1.2
20	30	0.1	2	2	7.4	1.4	6.3	2.1
20	30	0.8	1	1	10.9	2.2	1.1	2.7
20	30	0.8	1	2	10.4	2.0	4.2	1.8
20	30	0.8	2	1	12.6	3.5	13.8	4.4
20	30	0.8	2	2	5.8	1.4	6.9	1.6
				avg:	11.8	2.3	8.5	2.4
				max:	25.2	5.5	15.2	7.0

Table 11: VSS% and EVPI% for CTQD-AC$_{ud}$ instances with $|M| = 20$

7 Conclusions

In this paper, we have considered the Capacitated Supplier Selection problem with Total Quantity Discount policy and Activation Costs in which the products demand is explicitly considered as a stochastic variable (CTQD-AC$_{ud}$). The problem aims at selecting a subset of the suppliers and the relative purchasing plan such that the product demands are satisfied at minimum cost, also considering the discount policy offered by the suppliers and the cost to activate a business activity with them. The CTQD-AC$_{ud}$ has been shown by the recent literature to be very important to tackle long-term procurement settings mainly because of the possible savings it allows with respect to considering expected values for the demands. However, the problem has been shown to be also very difficult to solve for a sufficiently large number of scenarios and no efficient methods have been proposed so far. We have tried to bridge this gap by developing and testing new solution methods based on the structural properties of the problem. In particular, we have tested a Benders algorithm and developed different variants of a Progressive Hedging-based heuristic approach. Our Progressive Hedging has been improved with respect to its standard implementation through the introduction of several acceleration strategies, enhancing its efficiency and the effectiveness. The proposed algorithms have outperformed the already existing methods in efficiency on a large set of benchmark instances, thus allowing to have optimal or near-optimal solutions also for the biggest ones (that was not solved yet). This, in turn, has allowed us to show how the analysis of the stochastic solution for the biggest instances (in terms of VSS and EVPI) is in line with the trends shown by the smallest ones.

Acknowledgments

Partial funding for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) through its Discovery Grants Program. While working on this paper, prof. Guido Perboli was the head of the Urban Mobility and Logistics Systems (UMLS) initiative of the interdepartmental Center for Automotive Research and Sustainable mobility (CARS) at Politecnico di Torino, Italy.

References

Aissaoui, N., Haouari, M., Hassini, E., 2007. Supplier selection and order lot sizing modeling: A review. Computer & Operations Research 34 (12), 3516–3540.

Anupindi, R., Akella, R., 1993. Diversification under supply uncertainty. Management Science 39, 944–963.

Awasthi, A., Chauhan, S., Goyal, S., Proth, J.-M., 2009. Supplier selection problem for a single manufacturing unit under stochastic demand. International Journal of Production Economics 117, 229–233.

Benders, J. F., Dec. 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4 (1), 238–252.

Benton, W. C., 1991. Quantity discount decisions under conditions of multiple items. International Journal of Production Research 29, 1953–1961.

Beraldi, P., Bruni, M. E., Manerba, D., Mansini, R., 2017. A stochastic programming approach for the traveling purchaser problem. IMA Journal of Management Mathematics 28 (1), 41–63.

Birge, J. R., Louveaux, F. V., 1997. Introduction to Stochastic Programming. Springer Verlag, New York.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., Wallace, S. W., 2011. Progressive hedging-based metaheuristics for stochastic network design. Networks 58 (2), 114–124.

Crainic, T. G., Gobbato, L., Perboli, G., Rei, W., 2016. Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic. European Journal of Operational Research 253 (2), 404–417.

Crama, Y., Pascual, J. R., Torres, A., 2004. Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. European Journal of Operational Research 159 (2), 364–378.

Dada, M., Petruzzi, N., Schwarz, L. B., 2007. A newsvendors procurement problem when suppliers are unreliable. Manufactoring & Service Operations Management 9, 9–32.

Dickson, G. W., 1966. An analysis of vendor selection systems and decisions. Journal of purchasing 2, 5–17.

Fischetti, M., Ljubić, I., Sinnl, M., 2016. Benders decomposition without separability: A computational study for capacitated facility location problems. European Journal of Operational Research 253 (3), 557–569.

Fischetti, M., Salvagnin, D., Zanette, A., Jul 2010. A note on the selection of benders' cuts. Mathematical Programming 124 (1), 175–182.

Goossens, D., Maas, A., Spieksma, F., van de Klundert, J., 2007. Exact algorithms for a procurement problem with total quantity discounts. European Journal of Operational Research 178 (2), 603–626.

Hammami, R., Temponi, C., Frein, Y., 2014. A scenario-based stochastic model for supplier selection in global context with multiple buyers, currency fluctuation uncertainties, and price discounts. European Journal of Operational Research 233 (1), 159–170.

IBM Knowledge Center, 2017. Benders algorithm in CPLEX v12.7.0. Accessed: 2017-09-01. `https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.7.0/ilog.odms.cplex.help/CPLEX/ReleaseNotes/topics/releasenotes127/newBenders.html`.

Jolai, F., Neyestani, M. S., Golmakani, H., 2013. Multi-objective model for multi-period, multi-products, supplier order allocation under linear discount. International Journal of Management Science and Engineering Management 8 (1), 24–31.

Krichen, S., Laabidi, A., Abdelaziz, F. B., 2011. Single supplier multiple cooperative retailers inventory model with quantity discount and permissible delay in payments. Computers & Industrial Engineering 60, 164–172.

Løkketangen, A., Woodruff, D. L., 1996. Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. Journal of Heuristics 2 (2), 111–128.

Manerba, D., Mansini, R., 2012. An exact algorithm for the capacitated total quantity discount problem. European Journal of Operational Research 222 (2), 287–300.

Manerba, D., Mansini, R., 2014. An effective matheuristic for the capacitated total quantity discount problem. Computers & Operations Research 41 (1), 1–11.

Manerba, D., Mansini, R., Perboli, G., 2017a. The capacitated supplier selection problem with total quantity discount policy and activation costs under uncertainty. CIRRELT-2017-13. Submitted to journal.

Manerba, D., Mansini, R., Riera-Ledesma, J., 2017b. The traveling purchaser problem and its variants. European Journal of Operational Research 259 (1), 1–18.

Mansini, R., Savelsbergh, M. V. P., Tocchella, B., 2012. The supplier selection problem with quantity discounts and truckload shipping. Omega 40, 445–455.

McConnel, M. B., Galligan, D. T., 2004. The use of integer programming to select bulls across breeding companies with volume price discounts. Journal of Dairy Science 87, 3542–3549.

McDaniel, D., Devine, M., 1977. A modified benders' partitioning algorithm for mixed integer programming. Management Science 24 (3), 312–319.

Mirmohammadi, S. H., Shadrokh, S., Kianfar, F., 2009. An efficient optimal algorithm for the quantity discount problem in material requirement planning. Computers & Operations Research 36, 1780–1788.

Munson, C. L., Hu, J., 2010. Incorporating quantity discounts and their inventory impacts into the centralized purchasing decision. European Journal of Operational Research 201, 581–592.

Munson, C. L., Jackson, J., 2015. Quantity discounts: An overview and practical guide for buyers and sellers. Foundations and Trends in Technology, Information and Operations Management 8 (1–2), 1–130.

Munson, C. L., Rosenblatt, M. J., 1998. Theories and realities of quantity discounts: An exploration study. Production and Operations Management 7 (4), 352–369.

Nguyen, H., Rainwater, C., Mason, S., Pohl, E., 2014. Quantity discount with freight consolidation. Transportation Research Part E: Logistics and Transportation Review 66, 66–82.

Parlar, M., Wang, D., 1993. Diversification under yield randomness in inventory models. European Journal of Operational Research 66, 52–64.

Perboli, G., Gobbato, L., Maggioni, F., 2017. A progressive hedging method for the multi-path traveling salesman problem with stochastic travel times. IMA Journal of Management Mathematics 28 (1), 65–86.

Rockafellar, R. T., Wets, R. J.-B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 16 (1), 119–147.

Sen, A., Yaman, H., Guler, K., Korpeoglu, E., 2013. Multi-period supplier selection under price uncertainty. Journal of Operational Research Society 65 (11), 1636–1648.

Shahsavar, A., Zoraghi, N., Abbasi, B., 2016. Integration of resource investment problem with quantity discount problem in material ordering for minimizing resource costs of projects. Operational Research, 1–28.

van de Klundert, J., Kuipers, J., Spieksma, F. C. R., Winkels, M., 2005. Selecting telecommunication carriers to obtain volume discounts. Interfaces 35, 124–132.

Veliz, F. B., Watson, J.-P., Weintraub, A., Wets, R. J.-B., Woodruff, D. L., 2015. Stochastic optimization models in forest planning: a progressive hedging solution approach. Annals of Operations Research 232 (1), 259–274.

Watson, J.-P., Woodruff, D. L., 2011. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science 8 (4), 355–370.

Weber, C. A., Current, J. R., 1993. A multiple-objective approach to vendor selection. European Journal of Operational Research 68, 173–184.

Wetzstein, A., Hartmann, E., jr., W. B., Hohenstein, N.-O., 2016. A systematic assessment of supplier selection literature state-of-the-art and future scope. International Journal of Production Economics 182, 304–323.

Yang, S., Yang, J., Abdel-Malek, L., 2007. Sourcing with random yields and stochastic demand: a newsvendor approach. Computers & Operations Research 34, 3682–3690.

Zhang, J.-L., Zhang, M.-Y., 2011. Supplier selection and purchase problem with fixed cost and constrained order quantities under stochastic demand. International Journal of Production Economics 129 (1), 1–7.