An exact method for the Capacitated Total Quantity Discount Problem

Daniele Manerba, Renata Mansini
Department of Information Engineering, University of Brescia, Italy, {daniele.manerba@ing.unibs.it, renata.mansini@ing.unibs.it}

Technical Report 2011-2-66, April 2011.

Abstract

This paper analyzes the procurement problem of a company that needs to purchase a number of products from a set of suppliers to satisfy demand. The suppliers offer total quantity discounts and the company aims at selecting a set of suppliers so to satisfy product demand at minimum purchasing cost. The problem known as Total Quantity Discount Problem (TQDP) is strongly NP-hard. We study different families of valid inequalities and provide a branch-and-cut approach to solve the capacitated variant of the problem (Capacitated TQDP) where the quantity available for a product from a supplier is limited. A hybrid algorithm, called `HELP` (Heuristic Enhancement from LP), is used to provide an initial feasible solution to the exact approach. `HELP` exploits information provided by the continuous relaxation problem to construct neighbourhoods optimally searched through the solution of mixed integer subproblems. A streamlined version of the proposed exact method can optimally solve in a reasonable amount of time instances with up to 100 suppliers and 500 products and largely outperforms an existing approach available in the literature and CPLEX 12.2 that frequently runs out of memory before completing the search.

Key words: branch and cut, hybrid heuristics, supplier selection, total quantity discount.

---

1. Introduction

Let $S := \{1, \dots, n\}$ be a set of suppliers, indexed by i, and let $K := \{1, \dots, m\}$ be a set of products, indexed by k. Each product $k \in K$ can be purchased in a subset $S_k \subseteq S$ of suppliers at a non-negative basic price p_{ik}, potentially different for each supplier $i \in S_k$. For each product $k \in K$ a positive integer demand d_k is defined, and for each product $k \in K$ and each supplier $i \in S_k$ a quantity availability q_{ik} is specified, such that $\sum_{i \in S_k} q_{ik} \geq d_k$.

Each supplier $i \in S$ defines a set $R_i = \{1, \ldots, r_i\}$, indexed by r, of r_i consecutive and non-overlapping discount intervals $[l_i^r,\ u_i^r]$, where l_i^r and u_i^r are minimum and maximum number of product units to be purchased from supplier i to be in interval r. We assume that $\sum_{k \in K} q_{ik} \leq u_i^{r_i}, i \in S$. A discount rate δ_i^r is associated to each interval $r \in R_i$ such that $\delta_i^{r+1} \geq \delta_i^r$, $r = 1, \ldots, r_i - 1$. The interval in which the total quantity purchased lies determines the discount applied by the supplier to the total purchase cost (*total quantity discount policy*). For sake of simplicity we convert the discount rates into non-increasing unit prices $p_{ik}^r \geq 0$, i.e. $p_{ik}^{r+1} \leq p_{ik}^r, r = 1, \ldots, r_i - 1$. The Capacitated Total Quantity Discount problem looks for a subset of suppliers so that total demand is satisfied at a minimum purchasing cost.

The problem generalizes the TQD problem described in Goossens et al. (2007), where suppliers are assumed to have unlimited availability for offered products. In this paper the authors provide a first mathematical formulation for the TQD problem and show that the problem is NP-hard and that no polynomial-time approximation algorithm with constant ratio exists (unless $P = NP$). Some variants of the problem is studied taking into account constraints on market share, number of suppliers selected, quantity purchased (introduction of a disposal cost for extra units bought) and a multi-period variant. They propose a branch-and-bound algorithm based on a min-cost flow problem formulation, and compare it to plain CPLEX 8.1 and to the same solver but with cuts inhibited. Computational results are provided on instances with 10, 20 and 50 suppliers, a maximum of 3 or 5 volume intervals and a number of products equal to 40 and 100.

In the specialized literature the problem of selecting suppliers has received a great deal of attention. Interested readers are refereed to Weber et al. (1991), where a survey on criteria and analytical methods used in the vendor selection process is provided. As far as procurement optimization is concerned, two different lines of research can be found in the literature. The first line assumes demand as a deterministic quantity. The multi-period scenario problem in Rosenblat et al. (1998), the supplier selection and purchase problem with multiple item, multiple supplier and quantity limitation described in Benton (1991) and a variant with concave purchase cost and minimal and maximal ordering quantities in Chauhan and Proth (2003) belong to the first line. A second stream of research studies procurement problems under stochastic demand assumption. Anupindi and Akella (1993), Parlar and Wang (1993), Dada et al. (2007), Yang et al. (2007) analyze sourcing selection with uncertain suppliers. Awasthi et al. (2009) study the problem with limitation on minimum and maximum order sizes and propose a heuristic method. Zhang and Zhang (2011) extend the previous stochastic problem including holding and shortage costs and fixed costs for each selected supplier and present a branch-and-bound algorithm. Burke et al. (2007) analyze single period and single product sourcing decisions under demand uncertainty and proposed an optimal solution approach.

Vendor's quality and reliability are the most typical factors taken into account in procurement decision, even if pricing and discount policy are among the most critical ones (e.g. see Mohammad Ebrahim et al. (2009) where multiple price discount enviroment is considerated in a supplier selection and order lot sizing context). In particular, the total quantity discount policy occurs in several application contexts as in dairy (Mcconnel and Galligan 2004), in chemical industry (Crama et al. 2004) and in telecommunication (Van de Klundert et al. 2005). More recently, quantity discount structure appears in several papers on material requirement planning (e.g. see Hamid Mirmohammadi et al. 2009 where a multi-period horizon is considered), supply chain planning (Che et al. 2010) and complex purchasing decisions (e.g. Munson and Hu (2010) incorporate quantity discounts into a centralized purchasing decision and study their inventory impacts, while Krichen et al. (2011) study convenience of making retailers coalizations in the presence of a single supplier's discount policy and permissible delay in payments). Finally, following Goossens et al. (2007), in Mansini et al. (2011) the TQD procurement problem is extended to include truckload shipping costs. The authors develop integer programming based heuristics to solve the problem and demonstrate their efficacy on a large set of randomly generated instances.

In this paper we introduce different valid inequalities for the Capacitated TQD problem and develop a branch-and-cut approach (`B&C`) using exact separation algorithms. A heuristic method, called `HELP`, is also introduced to provide an initial feasible solution for the branch-and-cut. The method uses information provided by LP relaxation solution to formulate mixed integer subproblems embedded into a Variable Neighborhood Search framework. More precisely, we use a VNS with Decomposition exploring neighborhoods exactly through the optimal solution of such mixed integer subproblems. We also tested a streamlined version for both `B&C` and `HELP` approaches (`B&C-st`, `HELP-st`) based on a new formulation of the problem where the more effective valid inequalities are directly added without separation. We introduce different classes of hard to solve instances to evaluate performance of all proposed approaches. Extensive computational results have shown how `B&C` and its streamlined version

B&C-st result to be more efficient with respect to plain CPLEX 12.2 and to the branch-and-bound proposed in Goossens et al. (2007).

This paper provides some important contributions:

- to the best of our knowledge, this is the first exact approach for the capacitated version of the TQD problem. The method in the streamlined version has resulted to be extremely efficient and able to optimize space requirement with respect to CPLEX 12.2 that frequently runs out of memory;
- we introduce different classes of valid inequalities, that provide the first polyhedral result for the problem, and analyze their separation algorithms;
- the hybrid algorithm HELP is a valuable approach combining meta-heuristic with the optimal solution of integer subproblems.

The paper is organized as follows. In Section 2 we describe the mathematical formulation of the problem, the properties of its continuous relaxation and the classes of valid inequalities. Section 3 is devoted to solution algorithms. We introduce procedure HELP and the branch-and-cut approach implemented with the preprocessing used. A detailed description of the separation algorithms is also provided. Instances generation and computational results are described in Section 4, along with the streamlined version of B&C. Finally, some conclusions are drawn in Section 5.

2. Mathematical Model

In this section, we describe a natural integer linear-programming formulation for the Capacitated TQDP and analyze some classes of valid inequalities.

2.1 Integer Linear Programming formulation

We introduce the following sets of decision variables:

$z_{ik}^r :=$ number of units of product k purchased from supplier i in interval r, $k \in K,\ i \in S_k,\ r \in R_i$,

$$y_i^r := \begin{cases} 1 & \text{if } \sum_{k \in K} z_{ik}^r \in [l_i^r, u_i^r] \\ 0 & \text{otherwise} \end{cases} \qquad i \in S,\ r \in R_i.$$

The Capacitated Total Quantity Discount Problem can be formulated as follows:

$$(CTQDP) \quad \min \quad \sum_{i\in S}\sum_{r\in R_i}\sum_{k\in K} p_{ik}^r z_{ik}^r \tag{1}$$

subject to

$$\sum_{i\in S}\sum_{r\in R_i} z_{ik}^r = d_k \qquad k \in K \tag{2}$$

$$\sum_{r\in R_i} z_{ik}^r \le q_{ik} \qquad i \in S, k \in K \tag{3}$$

$$\sum_{r\in R_i} y_i^r \le 1 \qquad i \in S \tag{4}$$

$$\sum_{k\in K} z_{ik}^r - l_i^r y_i^r \ge 0 \qquad i \in S, r \in R_i \tag{5}$$

$$\sum_{k\in K} z_{ik}^r - u_i^r y_i^r \le 0 \qquad i \in S, r \in R_i \tag{6}$$

$$z_{ik}^r \ge 0 \text{ integer} \qquad i \in S, r \in R_i, k \in K \tag{7}$$

$$y_i^r \in \{0,1\} \qquad i \in S, r \in R_i \tag{8}$$

Objective function (1) establishes the minimization of the purchasing costs. Constraints (2) ensure that demand d_k is satisfied for each product k, whereas constraints (3) state that it is not possible to purchase from supplier i an amount of product k larger than quantity q_{ik} available. Constraints (4) guarantee that at most one interval for each supplier is selected. Constraints (5) and (6) define interval bounds for each supplier. If interval r for supplier i is selected ($y_i^r = 1$), then total amount purchased has to lie between the lower bound l_i^r and the upper bound u_i^r. On the contrary, if interval r is not selected ($y_i^r = 0$), then $\sum_{k\in K} z_{ik}^r = 0$. Finally, constraints (7)–(8) state non-negativity, integer and binary conditions.

In problem formulation integrality of z variables is not necessary. Condition is always satisfied if demands, products availability and lower and upper bounds are integral:

Proposition 1 *If all input data other than costs are integral and the continuous relaxation of (CTQDP) has an optimal solution, then there exists an optimal solution in which all z variables have integer values.*

In Goossens et al. (2007) the authors show that if there exists an optimal solution for the continuous relaxation of the Total Quantity Discount Problem *(TQDP)*, then there exists an optimal solution which selects the highest discount interval for each supplier. This property also holds for the continuous relaxation of *(CTQDP)*:

Proposition 2 *If the continuous relaxation of (CTQDP) has an optimal solution, then there exists an optimal solution in which all z and y variables are equal to 0, except those corresponding to the highest interval of each supplier.*

In other words there always exists an optimal solution for the continuous relaxation of the ($CTQDP$) in which products are bought at the lowest prices.

2.2 Valid inequalities

The linear relaxation of model (1)–(8) can be strengthened introducing the following classes of valid inequalities.

If interval r for supplier i is selected ($y_i^r = 1$), then the total quantity to buy from other suppliers has to be at least equal to total demand $\sum_{k \in K} d_k$ minus the maximum quantity obtainable from supplier i in interval r (i.e. $\min\left\{\sum_{k \in K} q_{ik}, u_i^r\right\}$), and the following valid inequalities can be formulated:

$$\sum_{\substack{h \in R_i \\ h \neq r}} \sum_{k \in K} z_{ik}^h + \sum_{\substack{j \in S \\ j \neq i}} \sum_{h \in R_j} \sum_{k \in K} z_{jk}^h \geq \sum_{k \in K} d_k - \min\left\{\sum_{k \in K} q_{ik}, u_i^r\right\} y_i^r \quad i \in S, r \in R_i. \tag{9}$$

Each inequality (9) can be separated into $O(|K|)$ different inequalities:

$$\sum_{\substack{h \in R_i \\ h \neq r}} z_{ik}^h + \sum_{\substack{j \in S_k \\ j \neq i}} \sum_{h \in R_j} z_{jk}^h \geq d_k - \min\{q_{ik}, u_i^r\} y_i^r \quad i \in S_k, r \in R_i, k \in K. \tag{10}$$

Furthermore, each inequality (10) can be rewritten in the following equivalent form:

$$z_{ik}^r \leq \min\{q_{ik}, u_i^r\} y_i^r, \quad i \in S_k, r \in R_i, k \in K. \tag{11}$$

Let us define as K_i the set of products offered by supplier i and as $\bar{K} \subset K_i$ the subset of products which result to be necessary to reach interval $r \in R_i$ (i.e. at least one product of $\bar{K}$ has to be purchased to get interval r). Then, given a supplier i and an interval r, the following inequalities hold:

$$\sum_{k \in \bar{K}} z_{ik}^r \geq (l_i^r - \sum_{k \in K \setminus \bar{K}} q_{ik}) y_i^r \quad \forall \bar{K} \subset K_i, \bar{K} \neq \emptyset. \tag{12}$$

Let $\bar{S} \subset S$ be a subset of suppliers such that there exists a product $k \in K$ with $\sum_{i \in S \setminus \bar{S}} q_{ik} < d_k$. Then, given a product k, the following family of valid inequalities holds:

$$\sum_{i \in \bar{S}} \sum_{r \in R_i} y_i^r \geq 1 \qquad \forall \bar{S} \subset S, \bar{S} \neq \emptyset. \tag{13}$$

These constraints state that, if suppliers in $S \setminus \bar{S}$ are not enough to provide the required demand d_k of product k, at least a supplier in $\bar{S}$ must be visited. Given $\bar{S}$, it is evident that $\sum_{i \in \bar{S}} \sum_{r \in R_i} z_{ik}^r \geq d_k - \sum_{i \in S_k \setminus \bar{S}} q_{ik}$ also holds.

Since in the optimal solution of the continuous relaxation of *(CTQDP)* the first interval of a supplier may be selected also without buying anything from it, inequalities (13) need for the following additional constraints:

$$\sum_{r \in R_i} \sum_{k \in K} z_{ik}^r \geq \sum_{r \in R_i} y_i^r \qquad i \in S. \tag{14}$$

Inequalities (14) imply that an interval of a supplier can be selected only if at least one unit of a product is bought in such interval.

3. Solution Algorithms

This section is mainly devoted to describe the implemented branch-and-cut approach. For a description and references on this method see Caprara and Fischetti (1997). The method exploits polyhedral results (valid inequalities) exposed in Section 2 and the effective hybrid approach `HELP` to get a good quality initial solution.

3.1 A Hybrid Heuristic

Heuristic Enhancement from LP (`HELP`) is a hybrid method which combines different features. More precisely, the algorithm is globally structured like a Variable Neighborhood Decomposition Search (VNDS) which explores defined neighborhoods exactly through the optimal solution of mixed integer subproblems.

Let us indicate as $(z^{(LP)}, y^{(LP)})$ the value assigned to variables in the optimal solution of the continuous relaxation problem of *(CTQDP)*. We define as $CTQDP(\bar{y})$ the subproblem obtained from *(CTQDP)* by fixing all y variables to some feasible value $\bar{y}$. The following proposition is trivially true:

Proposition 3 *Let $(\bar{z}, \bar{y})$ be the optimal solution of the subproblem CTQDP($\bar{y}$) when $\bar{y}_i^r = 1$ if $l_i^r \leq \sum_{k \in K} z_{ik}^{r(LP)} \leq u_i^r$ and zero otherwise. Then $(\bar{z}, \bar{y})$ is a feasible solution for (CTQDP).*

A solution generated as in Proposition 3 is feasible since intervals are selected according to quantities purchased satisfying products demand. Subproblem $CTQDP(\bar{y})$ only involves continuous variables but provides integer optimal solutions.

The hybrid algorithm HELP, pseudo-coded in Figure 1, receives as input a feasible solution $s^I = (z^I, y^I)$ computed as in Proposition 3 and provides as output a possibly improved integer feasible solution $s^F = (z^F, y^F)$.

INPUT:
integer feasible solution $s^I = (z^I, y^I)$ with value v^I.
OUTPUT:
integer feasible solution $s^F = (z^F, y^F)$ with value v^F.

```
s^F := s^I, v^F := v^I;
while (T_MAX is not reached) do
    Set h := 1, s := s^I;
    while (h ≤ h_MAX) do
        Randomly select a set S' of h suppliers;
        Define the neighborhood N(s, h);
        Construct the corresponding subproblem and optimally solve it;
        Let s' be the optimal solution, with value v';
        if (v' < v^F) then
            s^F := s', v^F := v';
            s := s';
            h := 1;
        else
            h := h + h_STEP
        end if
    end while
end while
```

Figure 1: HELP procedure.

Given a set S' of h suppliers, the neighborhood $N(s, h)$ consists of all solutions that differ from s for at most h selected intervals in $\bigcup R_i, i \in S'$. The procedure starts by randomly selecting S' and defining the corresponding neighborhood $N(s, h)$. Then a mixed integer subproblem is formulated assuming that all y variables but those corresponding to the h selected suppliers are fixed to the value they have in solution s. Since $l_i^0 = 0, i \in S$, if nothing has been bought from a supplier, we assume that the first interval is selected. The local search phase of the VNDS is implemented by optimally solving the subproblem with a MIP solver. If the optimal solution $s' = (z', y')$ has a better value than the best incumbent solution s^F, then s^F is updated and h is reset to the starting value 1. If no better solution is found h is increased by h_{STEP} and the procedure is repeated until h reaches h_{MAX} (internal while loop). A maximum computing time T_{max} is set as stopping rule: the procedure is repeated until the total time elapsed exceeds such a time threshold (external while loop).

At each restart the procedure uses the same starting solution s^I. In preliminary experiments we notice that restarting the procedure from the best solution previously found is usually less effective. Moreover, notice that the comparison for the replacement of the solution s considers the best incumbent integer solution value v^F and not the best solution value found in the current internal while loop.

HELP has some merits. First of all the method does not require additional procedures to verify and eventually restore feasibility (the optimal exploring of each neighborhood always guarantees the feasibility of the generated solution). Moreover, since the method sequentially solves subproblems, the resolution of a new subproblem can be sped up by using as initial solution the best integer feasible solution already found s^F.

3.2 The Exact Approach

We propose a branch-and-cut algorithm (B&C) implemented using CPLEX 12.2 through Concert Technology 2.3. The method first applies an ad-hoc preprocessing routine, that includes variables fixing and parameters bounding. Then the resulting strengthened model is used as input for HELP that, within the predefined computing time T_{MAX}, provides an initial feasible solution. During the branch-and-cut process we regularly separate all the proposed valid inequalities and add them to the model. No branching rules are introduced different from those directly applied by CPLEX 12.2. Preprocessing and separation algorithms used are described below.

Preprocessing

Let us define as $S^* := \left\{ i \in S : \text{ there exixst } k \in K \text{ such that } \sum_{j \in S_k \setminus \{i\}} q_{jk} < d_k \right\}$ the set of suppliers that has to be necessarily selected in any feasible solution of the problem, as $K^* := \left\{ k \in K : \sum_{i \in S_k} q_{ik} = d_k \right\}$ the set of products for which no suppliers selection has to be done, and as $R_i^* := \{ r \in R_i : \sum_{k \in K} q_{ik} < l_i^r \}$ the set of intervals for each supplier $i \in S$ that can never be reached also buying all quantities available. We can now strengthen our model as follows. If product $k \in K^*$ then inequalities (3) can be replaced with assignment constraints $\sum_{r \in R_i} z_{ik}^r = q_{ik}$. Similarly, for all $i \in S^*$ inequalities (4) become equality constraints. Moreover, all the following variables can be set to zero:

$$y_i^r = 0 \quad i \in S, r \in R_i^*,$$

$$z_{ik}^r = 0 \quad i \in S, k \in K, r \in R_i^*.$$

Finally, each coefficient q_{ik} can be replaced by $q'_{ik} = min\{q_{ik}, d_k\}$ in the model, in the valid inequalities and in the definition of the sets S^*, K^* and $R_i^*, i \in S$.

Separation algorithms

Given a fractional solution $(y^{(LP)}, z^{(LP)})$, a separation procedure consists of determining a member $\alpha y + \beta z \geq \gamma$ of a given family of valid *(CTQDP)* inequalities such that $\alpha y^{(LP)} + \beta z^{(LP)} < \gamma$ or to prove that no such inequality exists. We now describe the separation problems for valid inequalities (11), (12) and (13). All of them are solved exactly.

Separation of inequalities (11). The number of inequalities (11) is $O(|K| \sum_{i \in S} |R_i|)$. This allows us to separate these inequalities in polynomial time just verifying one by one if they are violated in the current relaxation. In our B&C method all violated valid inequalities (11) are added to the model. This simple separation procedure is pseudocoded in Fig. 2.

```
for all i ∈ S
    for all r ∈ R_i\R_i^*
        if y_i^{r(LP)} ≠ {0,1} then
            for all k ∈ K
                if z_ik^{r(LP)} > 0 then
                    if z_ik^{r(LP)} > min{q'_ik, u_i^r} y_i^{r(LP)} then
                        new cut found (z_ik^r ≤ q'_ik y_i^r)
```

Figure 2: Separation of (11) inequalities.

Separation of inequalities (12). The separation problem for inequalities (12) is the following 0-1 Knapsack problem:

$$\Delta = \max \quad l_i^r y_i^{r(LP)} - \sum_{k \in K} q'_{ik}(1 - w_k) y_i^{r(LP)} - \sum_{k \in K} z_{ik}^{r(LP)} w_k$$

$$\sum_{k \in K} (1 - w_k) q'_{ik} \leq l_i^r - 1$$

$$w_k \in \{0, 1\} \quad k \in K$$

where w_k, $k \in K$, is equal to 1 if product k belongs to the set $\bar{K}$ and 0 otherwise. The separation problem is optimally solved using CPLEX 12.2 thus providing the most violated inequality or proving that no such an inequality exists. If the optimal solution provides an objective function value $\Delta > 0$, then inequality is violated and can be added to the model. Note that if $z_{ik}^{r(LP)} = 0$ then w_k can be set to 1 corresponding to the best objective function value. Since a 0-1 Knapsack problem is solved for each $i \in S$ such that $y_i^{r(LP)} \neq 0$, the number of inequalities (12) added is $O(|S|)$ for each relaxation.

Separation of inequalities (13). A similar 0-1 Knapsack problem can be used to separate inequalities (13):

$$\Delta = \min \quad \sum_{i \in S} w_i \sum_{r \in R_i} y_i^{r(LP)}$$

$$\sum_{i \in S} w_i q'_{ik} \geq \sum_{i \in S} q'_{ik} - d_k + 1$$

$$w_i \in \{0, 1\} \quad \forall i \in S$$

where $w_i, i \in S$, is equal to 1 if supplier i belongs to the set $\bar{S}$ and 0 otherwise. If the optimal solution provides an objective function value $\Delta < 1$, then the inequality is violated and can be added to the model. Since a 0-1 Knapsack problem is solved for each $k \in K$, the number of inequalities (13) added is $O(|K|)$ for each relaxation.

4. Experimental Analysis

In this section we discuss instances generation and analyze algorithms performance. All computational tests have been done on a *Intel Core Duo* 2 GHz computer, with 2 GByte RAM and running *Windows Vista* 32-bit operating system. All instances and the relative solutions report are available at the web page *http://www.ing.unibs.it/∼orgroup/instances.html*.

4.1 Benchmark instances

In Goossens et al. (2007) the authors introduce some benchmark instances for the non-capacitated TQD problem consisting of 10, 20 and 50 suppliers, 40 or 100 products and with a maximum of 3 or 5 volume intervals. Furthermore they create two type of instances, one completely random and one with a special structure, inspired by the instances studied in Van de Klundert et al. (2005). All these instances seem to be very tractable in practice. Indeed in their tables the results, averaged on 10 instances, show that in only one case out of 24 the computational time required by plain CPLEX 8.1 is higher than 60 seconds. In all the remaining cases time is never larger than 15 seconds, and in 19 cases out of 24 less than 3 seconds. Furthermore, we have tested the same set of instances on our computer using CPLEX 12.2. The computational times we got largely outperform the previous ones making any comparison with our `B&C` useless.

Since *(CTQDP)* has never be studied before in literature and the available instances for the special case TQD do not provide a sound test bed for our problem, we generate a new collection of hard to solve benchmark problems. This data set consists of two classes inspired by practical cases and that differ only for the discount policy.

Class 1: for each supplier i a probability larger than one-fifth is randomly defined to decide if a product is offered. Then for each available product k the quantity q_{ik} is generated uniformly in [0, 15]. Base prices p_{ik} for all products and suppliers are generated as follow:

$$p_{ik} = p_k + p_k \left(1 - \frac{q_{ik}}{max_{j \in S_k}\{q_{jk}\}}\right) + p_k \left(1 - \frac{\sum_{w \in K} q_{iw}}{max_{j \in S_k}\{\sum_{w \in K} q_{jw}\}}\right)$$

where p_k is chosen randomly in [1, 200]. As in practical cases, rare products are usually more expensive and larger size suppliers can advance lower prices. About discount policies, we consider suppliers with a number of intervals ranging randomly between 3 and 5. Interval upper bounds are generated as an a priori percentage $0 < \alpha_i^r \leq 1$ of the total amount of products available from a specific supplier, i.e. $u_i^r = \lfloor \alpha_i^r \sum_{k \in K} q_{ik} \rfloor$ for all $i \in S, r \in R_i$. Lower bounds $l_i^r, r = 2, \ldots, r_i$ are set to the value corresponding to the upper bound of the previous interval plus 1, and $l_i^1 = 0, i \in S$. Preliminary experimental results show that instances with a structure where first intervals are larger and last intervals smaller are on average harder to solve. Since such a structure is also reasonable in practice we set $\alpha_i^1 \geq 0.6, i \in S$. Discount rates applied in all intervals are randomly generated under the following conditions: $\delta_i^1 = 0$ (first interval of each supplier have no discount rate), $\delta_i^{r+1} \geq \delta_i^r$ and $\delta_i^r \leq 0.5, i \in S$. We generate the demand d_k for each product as follows:

$$d_k := \left\lceil a_k - \left((a_k - 1) \quad \frac{\overline{p}_k}{\max_{k \in K}\{\overline{p}_k\}}\right)\right\rceil$$

where $\overline{p}_k = \frac{\sum_{i \in S_k} p_{ik}}{|S_k|}$ and a_k is the demand for product k as proposed in Riera-Ledesma et al. (2003), i.e. $a_k := \lceil \lambda \max_{i \in S_k}\{q_{ik}\} + (1 - \lambda) \sum_{i \in S_k} q_{ik} \rceil$ with $\lambda \in [0, 1]$. The more expensive the product, the lower the corresponding demand. Moreover parameter λ allows to control the number of suppliers included in a feasible solution.

Class 2: data are generated as in Class 1, except for the number of intervals which is equal to 3 for all suppliers, with $\alpha_i^1 = 0.7, \alpha_i^2 = 0.9$ and $\alpha_i^3 = 1, i \in S$. Moreover, discount rates are fixed for all suppliers and equal to {0%,10%,50%}.

4.2 Computational Results

For our computational experiments we consider instances with $n = \{50, 100\}$ and $m = \{100, 200, 300, 500\}$. We also consider two values for parameter λ, i.e. $\lambda = 0.1$ and $\lambda = 0.8$, representing the case in which the most part of suppliers are necessary to satisfy the demand and that in which only a restricted part of suppliers are necessary, respectively. This means $2 \cdot 4 \cdot 2 = 16$ instances for each class, and 32 all together.

We test the exact approach as described in Section 3, using CPLEX 12.2's callbacks to incorporate our separation procedures in the main branch-and-cut process. After some initial experiments we notice that inequalities (13) were the less effective ones, while their separation time was quite high, and we decide to abandon them. To prevent tailing-off problems, we decide to stop separation procedures when the estimated gap from the best bound is less than 1.5% (15% for $n = 100$ and $m = 500$ instances). `B&C` method starts with our preprocessing and the `HELP` heuristic providing the initial solution. Concerning `HELP`, we have set $h_{MAX} = \frac{n}{10}$, $h_{STEP} = \frac{h_{MAX}}{5}$ and a time limit $T_{max} = \phi_1 m + \phi_2 \lfloor n/100 \rfloor$, where $\phi_1 = 1/3$ and $\phi_2 = 30$ for Class 1 and λ=0.8 instances (which seem to be the most easy to solve) and where $\phi_1 = 2$ and $\phi_2 = 200$ for all other instances. In practice T_{max} increases linearly with number of products, and has an additional bonus time when n =100. We set a threshold on the computational time assigned to each branch-and-cut routine equal to 4 hours (14400 s.). For the `B&C` method, the time required by `HELP` is considered separately.

Table 1 shows the comparison between our `B&C` method and the plain resolution by CPLEX 12.2 for each instance (identified uniquely by number of supplier n, number of products m, class cl and parameter λ). For each method the computational time $t(s)$ (expressed in seconds), the best solution value found (v), and its percentage gap from the best lower bound found in the branch-and-cut tree (*gap%*) are specified. In particular, concerning B&C columns, *HELP t(s)* indicates the time of the heuristic, *t(s)* shows the time of the pure branch-and-cut process, whereas the sum of the two provides the total time required by `B&C`. When solution found is optimal the value is highlighted in bold font.

Analyzing plain CPLEX 12.2 results we can notice that Class 2 instances are more difficult to solve than Class 1, and $\lambda = 0.1$ instances are harder than $\lambda = 0.8$ ones. This is probably

Table 1: Comparison of B&C with respect to CPLEX 12.2.

				CPLEX 12.2			B&C			
n	m	cl	λ	t(s)	v	gap %	HELP t(s)	t(s)	v	gap %
50	100	1	0.1	569	**1065331.43**	0.00	205	286	**1065331.43**	0.00
50	100	1	0.8	127	**259244.06**	0.00	33	169	**259244.06**	0.00
50	100	2	0.1	5831	**888275.88**	0.00	200	6839	**888275.88**	0.00
50	100	2	0.8	818	**209340.00**	0.00	202	2013	**209340.00**	0.00
50	200	1	0.1	1343	**1952152.07**	0.00	408	622	**1952152.07**	0.00
50	200	1	0.8	330	**504369.86**	0.00	68	474	**504369.86**	0.00
50	200	2	0.1	11215	**1706913.61**	0.00	408	11763	**1706913.61**	0.00
50	200	2	0.8	4111	**425279.00**	0.00	437	4731	**425279.00**	0.00
50	300	1	0.1	2158	2724237.34	1.62	703	2182	**2710818.62**	0.00
50	300	1	0.8	1047	**810963.79**	0.00	100	2667	**810963.79**	0.00
50	300	2	0.1	2029	2485703.36	2.66	625	14400	2447712.05	0.22
50	300	2	0.8	14400	632539.00	2.43	610	14400	639075.00	2.58
50	500	1	0.1	2117	5121182.19	3.33	1017	9588	**5087519.13**	0.00
50	500	1	0.8	3042	1320857.89	0.13	193	7453	**1320857.89**	0.00
50	500	2	0.1	2122	4133709.70	12.96	1051	14400	4159286.90	2.44
50	500	2	0.8	1430	1172101.50	17.98	1054	14400	1205396.96	8.22
100	100	1	0.1	715	**2040173.12**	0.00	405	583	**2040173.12**	0.00
100	100	1	0.8	398	**459820.43**	0.00	67	586	**459820.43**	0.00
100	100	2	0.1	1450	**1560407.87**	0.00	415	1764	**1560407.87**	0.00
100	100	2	0.8	6838	**364230.50**	0.00	509	7001	**364230.50**	0.00
100	200	1	0.1	1396	4387071.63	1.49	625	6004	**4360648.52**	0.00
100	200	1	0.8	7038	**883411.52**	0.00	100	2602	**883411.52**	0.00
100	200	2	0.1	1087	3483981.93	10.64	638	14400	3386225.72	0.15
100	200	2	0.8	1396	713550.50	2.72	618	14400	715143.00	2.76
100	300	1	0.1	1717	5747187.67	3.04	838	14306	**5719285.13**	0.00
100	300	1	0.8	987	1444455.99	4.45	147	14400	1464378.63	2.69
100	300	2	0.1	1528	4980973.86	15.46	819	14400	5819996.00	16.62
100	300	2	0.8	1847	1145346.00	14.52	1999	14400	1213495.00	11.91
100	500	1	0.1	1954	9398635.80	8.46	1209	14400	9899214.23	7.43
100	500	1	0.8	2216	2269527.59	3.52	289	14400	2288223.34	3.12
100	500	2	0.1	1420	8330278.52	22.44	1450	14400	10701193.50	25.13
100	500	2	0.8	1304	1907453.00	23.44	1501	14400	2030372.00	19.72

due to the common discount policy used by all the suppliers in Class 2 instances that makes harder the evaluation on convenience to purchase a particular amount of products in a particular supplier. On the contrary, when number of suppliers required to satisfy the total demand decreases ($\lambda = 0.8$), the solver can easily identify those more promising. Aside from just described differences, a common trend can be noticed: growing the size of instances, computational time tends to increase up to when the solver is no more able to terminate the search going out of memory (the value of *gap%* is different from zero and the threshold time limit of 14400s has not been reached). In such cases, independently of the size of the instances, CPLEX 12.2 runs out of memory on average at the same time, thus dramatically decreasing its performance (value of the best integer solution found) when size of instances increases. In a more depth analysis, it can be noticed that out of memory status always happens at the root node, probably due to the enormous number of separated cuts leading to an unmanageable LP model. We clarify that CPLEX 12.2 runs in default configuration (including presolve, cuts generation, dynamic search and parallel optimization), can use all available resources and can save, if it's necessary, the branch-and-cut tree on hard disk.

If we consider our B&C performance, it is evident how the method always reaches the optimal solution or the time limit, never running out of memory (CPLEX 12.2 terminates in such a status 17 times out of 32). Indeed, the introduction of our ad-hoc cuts strengthen the LP model, preventing the explosion of its dimension. Our B&C finds the optimal solution in 19 instances, of which only 14 are solved to optimality by CPLEX 12.2. For 21 out of 32 instances we achieve the optimal solution or a better solution than CPLEX 12.2 does, but in only 8 cases our computational time is lower. In the remaining 11 instances CPLEX 12.2 outperforms our method. Furthermore, in Table 2 we provide for each instance additional information on the value of the best solution found (v), the number of suppliers selected (#), the number of cuts (11) and (12) separated and the percentage gap of the solution value found by HELP with respect to B&C solution value (HELP%). It's worth noticing that the number of violated inequalities (11) is strongly higher than the number of (12) ones. To conclude, the main drawback of B&C seems to be the large computational time spent in separation procedures and thus in solving a large number of LP subproblems.

Table 2: B&C results.

n	m	cl	λ	v	#	(11)	(12)	HELP%
50	100	1	0.1	1065331.43	47	4737	233	0.32
50	100	1	0.8	259244.06	27	2688	594	0.70
50	100	2	0.1	888275.88	39	5295	1163	4.19
50	100	2	0.8	209340.00	32	4125	2107	3.08
50	200	1	0.1	1952152.07	39	9233	432	1.41
50	200	1	0.8	504369.86	33	4872	357	0.67
50	200	2	0.1	1706913.61	43	9613	918	8.21
50	200	2	0.8	425279.00	28	8403	3106	2.11
50	300	1	0.1	2710818.62	43	15804	710	2.21
50	300	1	0.8	810963.79	34	8776	1070	2.02
50	300	2	0.1	2447712.05	49	14875	2653	4.16
50	300	2	0.8	639075.00	37	13290	2940	7.67
50	500	1	0.1	5087519.13	33	28845	925	5.08
50	500	1	0.8	1320857.89	31	14210	950	3.49
50	500	2	0.1	4159286.90	45	28136	2019	5.87
50	500	2	0.8	1205396.96	37	21479	2418	6.16
100	100	1	0.1	2040173.12	77	10345	774	1.93
100	100	1	0.8	459820.43	55	5066	1356	5.35
100	100	2	0.1	1560407.87	91	9261	1757	10.80
100	100	2	0.8	364230.50	54	7580	5035	3.16
100	200	1	0.1	4360648.52	69	22905	1818	4.33
100	200	1	0.8	883411.52	64	5637	310	5.02
100	200	2	0.1	3386225.72	89	19239	2013	15.16
100	200	2	0.8	715143.00	63	15508	6814	7.37
100	300	1	0.1	5719285.13	77	29708	1699	5.53
100	300	1	0.8	1464378.63	69	19090	2494	0.00
100	300	2	0.1	5819996.00	99	29293	2621	0.00
100	300	2	0.8	1213495.00	71	21101	3938	0.00
100	500	1	0.1	9899214.23	97	11819	97	0.00
100	500	1	0.8	2288223.34	75	5090	131	3.12
100	500	2	0.1	10701193.50	99	49730	778	0.00
100	500	2	0.8	2030372.00	60	25080	1224	0.00

The streamlined version B&C-st

In order to maintain all good features shown by B&C, and at the same time to improve its performances, we introduce a streamlined version (B&C-st). The main idea is to add inequalities (11) directly to the model at the preprocessing time. Motivation for this choice is threefold: first the number of inequalities (11) is polynomial, second this family of cuts seems to be the most effective one and finally a strengthened model highly enhances our LP-based heuristic HELP behaviour. In this way we have to manage quite larger but more strengthened LP problems, and we also avoid separation times for this class of cuts. This initial strengthening allows to work with better bounds, reducing the size of the branching tree, and to abandon inequalities (12) since after direct insertion of inequalities (11) they become almost ineffective (while their exact separation method is really time consuming). Moreover, avoiding callbacks to implement separation procedures allows to use parallel optimization in branch-and-cut routines.

We run B&C-st on the same 32 instances and we report the results in Table 3, so to compare it with the plain resolution by CPLEX 12.2 and with the basic B&C method. This table reports, for each instance, the best solution value found (*Best value*) and, for the three methods, the computational time $t(s)$, the percentage error with respect to *Best value* ($\Delta_b\%$) and the percentage gap to the best bound (*gap%*). From this table on, for B&C and B&C-st methods, the column $t(s)$ will provide the total time, including the time required by HELP. In columns $\Delta_b\%$ zero values are highlighted in bold font, indicating that the method has achieved the best solution. It is evident how our streamlined method largely outperforms the other ones in all but 2 instances, never running out of memory. Furthermore, if the same solution is found by different methods, B&C-st always get it in a lower computational time.

Table 4 investigates how the strengthen model affects the HELP performance: for each variant, column $t(s)$ shows the computational time in seconds, v the value of the provided solution and $\Delta_c\%$ its percentage error from the solution value found by plain CPLEX 12.2. The winning method is highlighted in bold: in all but 5 instances HELP-st finds a better solution than HELP, and for 6 times it even finds a better solution than CPLEX 12.2 (negative percentage deviation). Finally, if we look at the average gap on all the 32 instances, we can conclude that HELP heuristic can improve its performance by about 4% if run on the strengthened model.

Essentially B&C-st has proved to be the best exact method to solve *(CTQDP)* instances.

Table 3: Exact methods comparison.

					CPLEX 12.2			B&C			B&C-st		
n	m	cl	λ	Best value	t(s)	Δ_b%	gap %	t(s)	Δ_b%	gap%	t(s)	Δ_b%	gap%
50	100	1	0.1	1065331.43	569	**0.00**	0.00	491	**0.00**	0.00	270	**0.00**	0.00
50	100	1	0.8	259244.06	127	**0.00**	0.00	202	**0.00**	0.00	61	**0.00**	0.00
50	100	2	0.1	888275.88	5831	**0.00**	0.00	7039	**0.00**	0.00	1954	**0.00**	0.00
50	100	2	0.8	209340.00	818	**0.00**	0.00	2215	**0.00**	0.00	526	**0.00**	0.00
50	200	1	0.1	1952152.07	1343	**0.00**	0.00	1030	**0.00**	0.00	523	**0.00**	0.00
50	200	1	0.8	504369.86	330	**0.00**	0.00	542	**0.00**	0.00	201	**0.00**	0.00
50	200	2	0.1	1706913.61	11215	**0.00**	0.00	12171	**0.00**	0.00	4258	**0.00**	0.00
50	200	2	0.8	425279.00	4111	**0.00**	0.00	5168	**0.00**	0.00	1254	**0.00**	0.00
50	300	1	0.1	2710818.62	2158	0.50	1.62	2885	**0.00**	0.00	958	**0.00**	0.00
50	300	1	0.8	810963.79	1047	**0.00**	0.00	2767	**0.00**	0.00	338	**0.00**	0.00
50	300	2	0.1	2447712.05	2029	1.55	2.66	15025	**0.00**	0.22	7793	**0.00**	0.00
50	300	2	0.8	632539.00	14400	**0.00**	2.43	15010	1.03	2.58	2841	**0.00**	0.00
50	500	1	0.1	5087519.13	2117	0.66	3.33	10605	**0.00**	0.00	1571	**0.00**	0.00
50	500	1	0.8	1320857.89	3042	**0.00**	0.13	7646	**0.00**	0.00	738	**0.00**	0.00
50	500	2	0.1	4099394.12	2122	0.84	12.96	15451	1.46	2.44	15519	**0.00**	0.88
50	500	2	0.8	1172101.50	1430	**0.00**	17.98	15454	2.84	8.22	15410	0.28	3.98
100	100	1	0.1	2040173.12	715	**0.00**	0.00	988	**0.00**	0.00	456	**0.00**	0.00
100	100	1	0.8	459820.43	398	**0.00**	0.00	653	**0.00**	0.00	184	**0.00**	0.00
100	100	2	0.1	1560407.87	1450	**0.00**	0.00	2179	**0.00**	0.00	835	**0.00**	0.00
100	100	2	0.8	364230.50	6838	**0.00**	0.00	7510	**0.00**	0.00	1553	**0.00**	0.00
100	200	1	0.1	4360648.52	1396	0.61	1.49	6629	**0.00**	0.00	873	**0.00**	0.00
100	200	1	0.8	883411.52	7038	**0.00**	0.00	2702	**0.00**	0.00	570	**0.00**	0.00
100	200	2	0.1	3386225.72	1087	2.89	10.64	15038	**0.00**	0.15	15047	0.02	0.12
100	200	2	0.8	708833.50	1396	0.67	2.72	15018	0.89	2.76	11584	**0.00**	0.00
100	300	1	0.1	5719285.13	1717	0.49	3.04	15144	**0.00**	0.00	1504	**0.00**	0.00
100	300	1	0.8	1429177.24	987	1.07	4.45	14547	2.46	2.69	2941	**0.00**	0.00
100	300	2	0.1	4867747.01	1528	2.33	15.46	15219	19.56	16.62	15340	**0.00**	0.25
100	300	2	0.8	1096232.00	1847	4.48	14.52	16399	10.70	11.91	15271	**0.00**	1.65
100	500	1	0.1	9240585.56	1954	1.71	8.46	15609	7.13	7.43	2666	**0.00**	0.00
100	500	1	0.8	2264190.57	2216	0.24	3.52	14689	1.06	3.12	3591	**0.00**	0.00
100	500	2	0.1	8202625.72	1420	1.56	22.44	15850	30.46	25.13	15636	**0.00**	0.12
100	500	2	0.8	1821572.00	1304	4.71	23.44	15901	11.46	19.72	15721	**0.00**	1.28

To better evaluate its goodness, we extend the computational tests generating and solving 5 different instances for each combination of $\{n, m, cl, \lambda\}$. Tables 5-8 present computational results obtained by plain CPLEX 12.2 and by `B&C-st` on these new instances, grouped by class and value of parameter λ. In each table, each instance is now identified not only by number of suppliers n and number of products m but also by a serial number $it = 1, \ldots, 5$. The meaning of each column has already been defined. Column Δ_c% for `B&C-st` is emphasized in bold when the method outperforms or get exactly the same value than CPLEX 12.2.

New results confirm preliminary tests analysis. In only 69 out of 160 instances the best solution found by CPLEX 12.2 has been demonstrated to be optimal, in 81 instances the solver runs out of memory (always at the root node), and in the remaining 10 instances the time limit is reached. In the last 91 instances gaps from the best bound vary from 0.03% to 26.73%. The presence of such large gaps is more due to poor bounds than to low quality incumbent integer solution.

Table 4: HELP comparison.

				HELP			HELP-st		
n	m	cl	λ	t(s)	v	$\Delta_c\%$	t(s)	v	$\Delta_c\%$
50	100	1	0.1	205	1068735.88	0.32	201	**1065713.75**	0.04
50	100	1	0.8	33	261066.88	0.70	35	**259637.67**	0.15
50	100	2	0.1	200	925521.50	4.19	202	**922546.06**	3.86
50	100	2	0.8	202	215795.50	3.08	205	**215792.00**	3.08
50	200	1	0.1	408	1979610.88	1.41	401	**1954079.50**	0.10
50	200	1	0.8	68	**507735.19**	0.67	66	510971.44	1.31
50	200	2	0.1	408	1847132.88	8.21	401	**1748833.38**	2.46
50	200	2	0.8	437	**434254.00**	2.11	402	441427.00	3.80
50	300	1	0.1	703	2770732.75	1.68	614	**2723116.50**	-0.04
50	300	1	0.8	100	**827349.94**	2.02	125	830231.81	2.38
50	300	2	0.1	625	2549444.50	2.56	625	**2493530.25**	0.31
50	300	2	0.8	610	**688103.00**	8.78	610	689846.50	9.06
50	500	1	0.1	1017	5346015.50	4.39	1059	**5116289.50**	-0.10
50	500	1	0.8	193	1366908.38	3.49	199	**1326334.63**	0.41
50	500	2	0.1	1051	4403359.00	6.52	1119	**4148555.00**	0.36
50	500	2	0.8	1054	1279601.50	9.17	1010	**1177402.00**	0.45
100	100	1	0.1	405	2079544.50	1.93	402	**2040997.38**	0.04
100	100	1	0.8	67	484411.91	5.35	64	**462729.88**	0.63
100	100	2	0.1	415	1728984.25	10.80	402	**1578341.38**	1.15
100	100	2	0.8	509	375725.50	3.16	421	**371188.00**	1.91
100	200	1	0.1	625	4549419.00	3.70	629	**4374175.50**	-0.29
100	200	1	0.8	100	927756.81	5.02	113	**900455.88**	1.93
100	200	2	0.1	638	3899494.50	11.93	647	**3461527.50**	-0.64
100	200	2	0.8	618	767879.63	7.61	616	**754109.69**	5.68
100	300	1	0.1	838	6035541.50	4.78	890	**5723244.00**	-0.42
100	300	1	0.8	147	1464378.63	1.36	146	**1454311.75**	0.68
100	300	2	0.1	819	5819996.00	14.42	940	**5050969.50**	1.39
100	300	2	0.8	1999	**1213495.00**	5.62	871	1229681.75	6.86
100	500	1	0.1	1209	9899214.00	5.33	1416	**9321888.00**	-0.82
100	500	1	0.8	289	2359518.00	3.97	277	**2287237.50**	0.78
100	500	2	0.1	1450	10701194.00	28.46	1236	**8428225.00**	1.18
100	500	2	0.8	1501	2030372.00	6.44	1321	**2009049.50**	5.33
						5.60			1.66

Let's now analyze B&C-st results. HELP-st heuristic has a discontinuous performance, showing the best results for Class 1 and λ=0.1 instances, where the procedure even outperforms CPLEX 12.2 in 18 out of 40 instances (for the remaining cases the percentage deviation is always lower than 1%). The performance of our complete exact algorithm is instead impressive. First of all, the algorithm never runs out of memory. Moreover, it reaches the optimal solution in 123 instances out of 160 (about twice the number of instances solved by CPLEX 12.2), while in the remaining 37 instances gaps from the best bound vary from 0.03% to 6.26%. Notice that all Class 1 instances have been solved to optimality.

Looking at the highlighted values in column $\Delta_c\%$, it is evident how B&C-st comes out to be the winning method in 156 out of 160 instances, with improvements exceeding 4% in some cases. In the remaining 4 instances CPLEX 12.2 outperforms B&C-st but with an improvement never larger than 0.46%. Furthermore, when both methods get the same solution value, B&C-st always requires a lower computational time. Finally, we believe that the total computational time of our method can be further improved better tuning the stopping rule for HELP.

In Table 9 all the results are summarized showing average percentage deviation ($\overline{\Delta_c}\%$) on the 5 instances for each pair (n, m). Column *impr* provides the total number of times out of 5 where B&C-st reaches or outperforms CPLEX 12.2, whereas into brackets the number of strict improvements are shown. We can clearly see that average percentage deviation are always negative (B&C-st gets better result than CPLEX 12.2) or at most equal to zero, with a maximum improvement by 2.66%. Looking at *impr* column, given n, the number of strict improvements obtained by our algorithm grows with m. The rate of growth seems to depend on the Class and on the λ value. Finally, if we consider the number of times our method at least equals CPLEX 12.2, it is interesting to notice that it tends to remain unchanged at its maximum value, but for 2 cases.

5. Conclusions

In this paper we present the capacitated version of TQD problem *(CTQDP)* and provide different families of valid inequalities. We propose a branch-and-cut approach implemented through CPLEX 12.2, including preprocessing routines, ad-hoc separation procedures for the studied valid inequalities and an innovative heuristic method to find a good starting solution. An extensive computational analysis has been conducted on a large number of randomly generated instances. After some preliminary experiments a streamlined version of the initial exact algorithm has been developed. The new method largely outperforms CPLEX 12.2 plain execution. The following main conclusions can be drawn:

- B&C-st has come out to be the most performing exact algorithm for the *(CTQDP)* available in the literature;
- plain CPLEX 12.2 does not seem to always be a reliable solver, since it frequently runs out of memory;
- HELP-st has shown to be a valuable method to provide an initial solution, especially for Class 1 and $\lambda = 0.1$. We believe the method deserves to be further specialized to become a strong stand-alone algorithm for *(CTQDP)*.

Table 5: Results for Class 1 and λ=0.1.

			CPLEX 12.2			HELP-st		B&C-st			
n	m	it	t(s)	v	gap%	t(s)	Δ_c%	t(s)	v	Δ_c%	gap%
50	100	1	569	1065331.43	0.00	201	0.04	270	1065331.43	**0.00**	0.00
50	100	2	514	1053147.65	0.00	200	0.28	270	1053147.65	**0.00**	0.00
50	100	3	413	1063178.97	0.00	203	0.04	244	1063178.97	**0.00**	0.00
50	100	4	508	933882.86	0.00	201	0.22	234	933882.86	**0.00**	0.00
50	100	5	240	994708.65	0.00	200	0.11	234	994708.65	**0.00**	0.00
			448.8		0.00	201	0.14	250.4			0.00
50	200	1	1343	1952152.07	0.00	401	0.10	523	1952152.07	**0.00**	0.00
50	200	2	1706	2159349.57	1.53	401	-1.06	607	2132512.11	**-1.24**	0.00
50	200	3	2789	1998714.50	0.00	404	0.33	522	1998714.50	**0.00**	0.00
50	200	4	1328	2103277.09	0.00	405	0.14	508	2103277.09	**0.00**	0.00
50	200	5	2072	1935851.94	0.00	409	1.01	517	1935851.94	**0.00**	0.00
			1847.6		0.31	404	0.10	535.4			0.00
50	300	1	2158	2724237.34	1.62	614	-0.04	958	2710818.62	**-0.49**	0.00
50	300	2	2173	2769848.99	1.19	614	-0.10	772	2762370.39	**-0.27**	0.00
50	300	3	1900	3157975.32	0.90	611	0.12	975	3150592.95	**-0.23**	0.00
50	300	4	1522	3207430.36	2.58	602	-0.37	1093	3180894.49	**-0.83**	0.00
50	300	5	1875	3222222.31	1.26	622	-0.29	805	3206291.08	**-0.49**	0.00
			1925.6		1.51	612.6	-0.14	920.6			0.00
50	500	1	2117	5121182.19	3.33	1059	-0.10	1571	5087519.13	**-0.66**	0.00
50	500	2	2255	4848327.44	3.43	1028	-0.66	1534	4816132.22	**-0.66**	0.00
50	500	3	2411	5082202.06	1.70	1003	0.13	1711	5073596.64	**-0.17**	0.00
50	500	4	2005	4986680.68	3.26	1043	0.33	1517	4976074.66	**-0.21**	0.00
50	500	5	2164	5068637.43	2.78	1007	0.32	1678	5053157.68	**-0.31**	0.00
			2190.4		2.90	1028	0.00	1602.2			0.00
100	100	1	715	2040173.12	0.00	402	0.04	456	2040173.12	**0.00**	0.00
100	100	2	849	1912898.23	0.00	405	0.11	481	1912898.23	**0.00**	0.00
100	100	3	939	1873337.80	0.00	408	0.00	453	1873337.80	**0.00**	0.00
100	100	4	948	1922911.67	0.00	402	0.15	503	1922911.67	**0.00**	0.00
100	100	5	1438	2155736.71	0.00	405	0.03	536	2155736.71	**0.00**	0.00
			977.8		0.00	404.4	0.07	485.8			0.00
100	200	1	1396	4387071.63	1.49	629	-0.29	873	4360648.52	**-0.60**	0.00
100	200	2	1471	4177015.49	1.80	625	-0.11	933	4154529.60	**-0.54**	0.00
100	200	3	1703	3820078.38	0.80	635	0.32	1100	3797749.45	**-0.58**	0.00
100	200	4	2250	3960301.20	1.32	621	-0.58	879	3926225.47	**-0.86**	0.00
100	200	5	1286	3699886.64	3.33	626	-0.75	962	3668949.23	**-0.84**	0.00
			1621.2		1.75	627.2	-0.28	949.4			0.00
100	300	1	1717	5747187.67	3.04	890	-0.42	1504	5719285.13	**-0.49**	0.00
100	300	2	7100	5731704.42	1.09	907	-0.28	1429	5711788.37	**-0.35**	0.00
100	300	3	6789	5624702.21	1.77	863	-0.50	1763	5584860.56	**-0.71**	0.00
100	300	4	1569	6151799.68	2.93	848	0.25	1583	6143443.91	**-0.14**	0.00
100	300	5	1486	6205971.68	4.93	849	-0.49	1338	6135946.34	**-1.13**	0.00
			3732.2		2.75	871.4	-0.29	1523.4			0.00
100	500	1	1954	9398635.80	8.46	1416	-0.82	2666	9240585.56	**-1.68**	0.00
100	500	2	2687	10048109.80	3.98	1240	0.46	2621	9971412.11	**-0.76**	0.00
100	500	3	2219	9925337.82	5.48	1469	-0.43	3317	9861834.02	**-0.64**	0.00
100	500	4	2197	9731289.08	7.37	1452	-0.68	5319	9632126.38	**-1.02**	0.00
100	500	5	2741	9936694.61	4.95	1267	0.38	5143	9851746.34	**-0.85**	0.00
			2359.6		6.05	1368.8	-0.22	3813.2			0.00

Table 6: Results for Class 1 and λ=0.8.

			CPLEX 12.2			HELP-st		B&C-st			
n	m	it	t(s)	v	gap%	t(s)	Δ_c%	t(s)	v	Δ_c%	gap%
50	100	1	127	259244.06	0.00	35	0.15	61	259244.06	**0.00**	0.00
50	100	2	275	239212.53	0.00	34	0.90	83	239212.53	**0.00**	0.00
50	100	3	120	255411.50	0.00	34	0.80	72	255411.50	**0.00**	0.00
50	100	4	77	245170.07	0.00	34	0.00	60	245170.07	**0.00**	0.00
50	100	5	73	257293.66	0.00	34	0.52	68	257293.66	**0.00**	0.00
			134.4		0.00	34.2	0.47	68.8			0.00
50	200	1	330	504369.86	0.00	66	1.31	201	504369.86	**0.00**	0.00
50	200	2	401	501271.64	0.00	67	1.76	199	501271.64	**0.00**	0.00
50	200	3	901	546469.73	0.00	67	2.37	177	546469.73	**0.00**	0.00
50	200	4	292	520573.11	0.00	67	2.74	175	520573.11	**0.00**	0.00
50	200	5	600	475366.42	0.00	70	2.28	164	475366.42	**0.00**	0.00
			504.8		0.00	67.4	2.09	183.2			0.00
50	300	1	1047	810963.79	0.00	125	2.38	338	810963.79	**0.00**	0.00
50	300	2	462	801325.23	0.00	101	0.44	198	801325.23	**0.00**	0.00
50	300	3	379	754396.40	0.00	114	1.88	264	754396.40	**0.00**	0.00
50	300	4	702	785817.78	0.00	105	0.98	208	785817.78	**0.00**	0.00
50	300	5	367	743753.17	0.00	100	0.25	204	743753.17	**0.00**	0.00
			591.4		0.00	109	1.19	242.4			0.00
50	500	1	3042	1320857.89	0.13	199	0.41	738	1320857.89	**0.00**	0.00
50	500	2	1738	1271879.51	0.64	174	2.47	491	1267401.38	**-0.35**	0.00
50	500	3	1651	1347170.84	0.03	185	0.05	377	1347170.84	**0.00**	0.00
50	500	4	2086	1219496.74	0.00	184	3.73	436	1219496.74	**0.00**	0.00
50	500	5	2073	1228369.20	0.00	211	2.95	400	1228369.20	**0.00**	0.00
			2118		0.16	190.6	1.92	488.4			0.00
100	100	1	398	459820.43	0.00	64	0.63	184	459820.43	**0.00**	0.00
100	100	2	630	422275.23	0.00	69	2.20	204	422275.23	**0.00**	0.00
100	100	3	499	445686.18	0.00	69	1.88	149	445686.18	**0.00**	0.00
100	100	4	582	472285.13	0.00	65	1.29	156	472285.13	**0.00**	0.00
100	100	5	675	416908.89	0.00	66	1.06	207	416908.89	**0.00**	0.00
			556.8		0.00	66.6	1.41	180			0.00
100	200	1	7038	883411.52	0.00	113	1.93	570	883411.52	**0.00**	0.00
100	200	2	1013	875510.12	0.00	110	1.98	501	875510.12	**0.00**	0.00
100	200	3	1136	900431.23	0.64	105	1.24	391	895753.50	**-0.52**	0.00
100	200	4	2639	883419.34	0.00	106	1.53	358	883419.34	**0.00**	0.00
100	200	5	3492	878740.30	0.00	109	1.78	421	878740.30	**0.00**	0.00
			3063.6		0.13	108.6	1.69	448.2			0.00
100	300	1	987	1444455.99	4.45	146	0.68	2941	1429177.24	**-1.06**	0.00
100	300	2	12343	1433131.13	0.00	151	3.53	1241	1433131.13	**0.00**	0.00
100	300	3	1108	1371394.39	1.02	160	1.24	1098	1365427.33	**-0.44**	0.00
100	300	4	1432	1285574.28	0.59	163	2.35	1480	1282178.92	**-0.26**	0.00
100	300	5	11491	1255026.29	0.00	167	2.54	1221	1255026.29	**0.00**	0.00
			5472.2		1.21	157.4	2.07	1596.2			0.00
100	500	1	2216	2269527.59	3.52	277	0.78	3591	2264190.57	**-0.24**	0.00
100	500	2	1681	2302769.93	3.56	289	0.04	4035	2290165.46	**-0.55**	0.00
100	500	3	2135	2245209.10	2.48	329	4.19	4808	2229422.19	**-0.70**	0.00
100	500	4	1389	2025046.59	1.68	271	1.96	3644	2025046.59	**0.00**	0.00
100	500	5	2696	2224661.28	1.47	329	3.90	3976	2218089.75	**-0.30**	0.00
			2023.4		2.54	299	2.17	4010.8			0.00

Table 7: Results for Class 2 and λ=0.1.

			CPLEX 12.2			HELP-st		B&C-st			
n	m	it	t(s)	v	gap%	t(s)	Δ_c%	t(s)	v	Δ_c%	gap%
50	100	1	5831	888275.88	0.00	202	3.86	1954	888275.88	**0.00**	0.00
50	100	2	762	750361.55	0.00	201	1.58	573	750361.55	**0.00**	0.00
50	100	3	930	784005.14	0.00	201	1.82	442	784005.14	**0.00**	0.00
50	100	4	1163	828232.43	0.00	207	2.79	487	828232.43	**0.00**	0.00
50	100	5	1588	752930.08	0.00	207	2.50	483	752930.08	**0.00**	0.00
			2054.8		0.00	203.6	2.51	787.8			0.00
50	200	1	11215	1706913.61	0.00	401	2.46	4258	1706913.61	**0.00**	0.00
50	200	2	14401	1542909.08	0.30	405	6.29	4530	1542150.28	**-0.05**	0.00
50	200	3	11534	1554503.81	0.00	402	3.03	2414	1554503.81	**0.00**	0.00
50	200	4	14403	1711142.17	0.80	401	3.97	6951	1703115.96	**-0.47**	0.00
50	200	5	14400	1569847.52	0.06	410	4.14	6888	1569847.52	**0.00**	0.00
			13190.6		0.23	403.8	3.98	5008.2			0.00
50	300	1	2029	2485703.36	2.66	625	0.31	7793	2447712.05	**-1.53**	0.00
50	300	2	2541	2509478.09	2.43	615	2.33	7008	2461007.61	**-1.93**	0.00
50	300	3	1328	2353932.35	6.80	627	3.51	15027	2333156.03	**-0.88**	0.48
50	300	4	2157	2204206.51	2.21	606	-0.02	4371	2164848.92	**-1.79**	0.00
50	300	5	2777	2387421.31	1.94	602	3.83	15002	2358755.50	**-1.20**	0.31
			2166.4		3.21	615	1.99	9840.2			0.16
50	500	1	2122	4133709.70	12.96	1119	0.36	15519	4099394.12	**-0.83**	0.88
50	500	2	1360	3856333.91	15.66	1020	3.57	15420	3847179.41	**-0.24**	0.17
50	500	3	1502	4072415.76	15.99	1100	4.08	14777	4037572.32	**-0.86**	0.00
50	500	4	1965	3870875.59	13.28	1003	4.72	15403	3860036.57	**-0.28**	1.14
50	500	5	2011	3870599.61	14.02	1010	4.21	15410	3846026.49	**-0.63**	0.69
			1792		14.38	1050.4	3.39	15305.8			0.58
100	100	1	1450	1560407.87	0.00	402	1.15	835	1560407.87	**0.00**	0.00
100	100	2	4545	1430206.02	0.00	416	2.68	3505	1430206.02	**0.00**	0.00
100	100	3	5595	1655161.62	0.00	406	1.97	2280	1655161.62	**0.00**	0.00
100	100	4	4193	1459152.50	0.00	409	2.40	1596	1459152.50	**0.00**	0.00
100	100	5	3717	1570368.95	0.00	409	4.03	1313	1570368.95	**0.00**	0.00
			3900		0.00	408.4	2.45	1905.8			0.00
100	200	1	1087	3483981.93	10.64	647	-0.64	15047	3386881.90	**-2.79**	0.12
100	200	2	14406	3354900.43	0.18	604	1.82	15004	3353533.05	**-0.04**	0.11
100	200	3	1133	3473375.93	11.08	621	-1.23	15021	3359507.13	**-3.28**	0.10
100	200	4	1194	3404381.86	11.91	645	-0.42	10297	3285983.24	**-3.48**	0.00
100	200	5	1198	3352716.72	9.03	648	0.18	15048	3254487.50	**-2.93**	0.12
			3803.6		8.57	633	-0.06	14083.4			0.09
100	300	1	1528	4980973.86	15.46	940	1.39	15340	4867747.01	**-2.27**	0.25
100	300	2	1629	5052173.09	18.22	926	1.63	15326	4918446.29	**-2.65**	0.12
100	300	3	1172	4824938.50	17.63	909	2.38	15309	4680882.21	**-2.99**	0.03
100	300	4	1344	5032282.62	16.73	887	0.43	15287	4942760.09	**-1.78**	0.25
100	300	5	1191	4959282.08	19.88	869	-0.71	15269	4852938.98	**-2.14**	0.20
			1372.8		17.58	906.2	1.02	15306.2			0.17
100	500	1	1420	8330278.52	22.44	1236	1.18	15636	8202625.72	**-1.53**	0.12
100	500	2	1326	8460982.77	22.94	1304	5.63	15704	8252585.36	**-2.46**	0.16
100	500	3	1754	8636182.58	19.31	1209	2.88	15609	8522604.60	**-1.32**	0.88
100	500	4	1419	9007483.64	22.68	1211	6.09	15611	8805448.70	**-2.24**	0.55
100	500	5	1095	8410177.22	26.73	1506	0.99	15906	8277027.11	**-1.58**	0.92
			1402.8		22.82	1293.2	3.35	15693.2			0.53

Table 8: Results for Class 2 and λ=0.8.

			CPLEX 12.2			HELP-st		B&C-st			
n	m	it	t(s)	v	gap%	t(s)	Δ_c%	t(s)	v	Δ_c%	gap%
50	100	1	818	209340.00	0.00	205	3.08	526	209340.00	**0.00**	0.00
50	100	2	622	221929.00	0.00	202	0.20	383	221929.00	**0.00**	0.00
50	100	3	1085	204096.50	0.00	201	4.44	419	204096.50	**0.00**	0.00
50	100	4	1263	233650.00	0.00	201	2.59	450	233650.00	**0.00**	0.00
50	100	5	613	208702.61	0.00	201	3.66	328	208702.61	**0.00**	0.00
			880.2		0.00	202	2.80	421.2			0.00
50	200	1	4111	425279.00	0.00	402	3.80	1254	425279.00	**0.00**	0.00
50	200	2	6744	470957.00	0.00	405	8.11	1628	470957.00	**0.00**	0.00
50	200	3	4460	387964.50	0.00	413	3.77	1084	387964.50	**0.00**	0.00
50	200	4	4677	405405.50	0.00	404	3.64	1676	405405.50	**0.00**	0.00
50	200	5	1337	416658.50	0.00	401	0.00	663	416658.50	**0.00**	0.00
			4265.8		0.00	405	3.86	1261			0.00
50	300	1	14400	632539.00	2.43	610	9.06	2841	632539.00	**0.00**	0.00
50	300	2	14402	599533.50	1.81	612	2.49	6981	599533.50	**0.00**	0.00
50	300	3	1422	663491.25	4.74	638	-0.71	2762	657022.00	**-0.98**	0.00
50	300	4	14404	624807.00	1.07	624	2.61	3760	624807.00	**0.00**	0.00
50	300	5	14401	635619.00	4.03	603	0.48	4418	631879.00	**-0.59**	0.00
			11805.8		2.82	617.4	2.78	4152.4			0.00
50	500	1	1430	1172101.50	17.98	1010	0.45	15410	1175384.00	0.28	3.98
50	500	2	14400	978685.50	3.72	1143	9.52	11830	978685.50	**0.00**	0.00
50	500	3	904	1196591.00	20.93	1007	1.80	15407	1178217.00	**-1.54**	4.38
50	500	4	1420	1135915.50	15.75	1018	0.46	15418	1141164.00	0.46	4.66
50	500	5	1081	1070461.50	18.14	1040	1.58	15440	1069179.00	**-0.12**	5.49
			3847		15.30	1043.6	2.76	14701			3.70
100	100	1	6838	364230.50	0.00	421	1.91	1553	364230.50	**0.00**	0.00
100	100	2	4612	383295.50	0.00	413	7.81	3290	383295.50	**0.00**	0.00
100	100	3	2483	367575.50	0.00	411	2.14	1774	367575.50	**0.00**	0.00
100	100	4	2459	387625.50	0.00	402	7.43	1729	387625.50	**0.00**	0.00
100	100	5	4071	374945.00	0.00	407	5.87	1874	374945.00	**0.00**	0.00
			4092.6		0.00	410.8	5.03	2044			0.00
100	200	1	1396	713550.50	2.72	616	5.68	11584	708833.50	**-0.66**	0.00
100	200	2	1204	759045.50	5.46	661	4.31	15061	748779.00	**-1.35**	1.74
100	200	3	1032	797245.50	8.43	680	3.18	15080	775076.00	**-2.78**	0.39
100	200	4	14404	770827.00	2.03	649	5.93	5004	768933.00	**-0.25**	0.00
100	200	5	1013	783995.00	9.64	607	4.91	15007	765640.00	**-2.34**	1.14
			3809.8		5.66	642.6	4.80	12347.2			0.65
100	300	1	1847	1145346.00	14.52	871	6.86	15271	1096232.00	**-4.29**	1.65
100	300	2	4976	1181107.95	5.28	865	6.41	15265	1164793.00	**-1.38**	2.59
100	300	3	1096	1207258.00	22.17	812	6.80	15212	1158439.50	**-4.04**	0.91
100	300	4	927	1187808.50	18.34	893	5.46	15293	1187005.00	**-0.07**	3.70
100	300	5	1100	1258930.80	19.59	862	3.82	15262	1214561.00	**-3.52**	1.77
			1989.2		15.98	860.6	5.87	15260.6			2.12
100	500	1	1304	1907453.00	23.44	1321	5.33	15721	1821572.00	**-4.50**	1.28
100	500	2	1287	1858030.00	21.12	1289	5.07	15689	1827191.50	**-1.66**	1.51
100	500	3	993	1917578.50	25.57	1221	0.77	15621	1919003.50	0.07	6.26
100	500	4	2934	1950188.00	19.98	1575	1.97	15975	1888163.50	**-3.18**	2.58
100	500	5	1026	1851845.00	22.14	1292	11.83	15692	1855526.00	0.20	1.62
			1508.8		22.45	1339.6	4.99	15739.6			2.65

Table 9: B&C-st summarized results.

		Class 1, λ=0.1		**Class 1, λ=0.8**		**Class 2, λ=0.1**		**Class 2, λ=0.8**			
n	m	$\overline{\Delta_c}$%	impr	$\overline{\Delta_c}$%	impr	$\overline{\Delta_c}$%	impr	$\overline{\Delta_c}$%	impr	avg $\overline{\Delta_c}$%	tot impr
50	100	0.00	5(0)	0.00	5(0)	0.00	5(0)	0.00	5(0)	0.00	20(0)
50	200	-0.25	5(1)	0.00	5(0)	-0.10	5(2)	0.00	5(0)	-0.09	20(3)
50	300	-0.46	5(5)	0.00	5(0)	-1.47	5(5)	-0.31	5(2)	-0.56	20(12)
50	500	-0.40	5(5)	-0.07	5(1)	-0.57	5(5)	-0.18	3(2)	-0.31	18(13)
100	100	0.00	5(0)	0.00	5(0)	0.00	5(0)	0.00	5(0)	0.00	20(0)
100	200	-0.68	5(5)	-0.10	5(1)	-2.50	5(5)	-1.48	5(5)	-1.19	20(16)
100	300	-0.56	5(5)	-0.35	5(3)	-2.37	5(5)	-2.66	5(5)	-1.48	20(18)
100	500	-0.99	5(5)	-0.36	5(4)	-1.83	5(5)	-1.81	3(3)	-1.25	18(17)
		-0.42	40(26)	-0.11	40(9)	-1.10	40(27)	-0.81	36(17)	**-0.61**	**156(79)**

References

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

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

Benton W.C. 1991. Quantity discount decisions under conditions of multiple items, multiple suppliers and resourse limitations. *International Journal of Production Research* **29** 1953-1961.

Burke G.J., J.E. Carrillo, A.J. Vakharia. 2007. Single versus multiple supplier sourcing strategies. *European Journal of Operational Research* **182** 95-112.

Caprara A., M. Fischetti. 1997. Branch-and-Cut Algorithms. in M. Dell'Amico, F. Maffioli, S. Martello, eds., Annotated Bibliographies in Combinatorial Optimization, Wiley 45-63

Chauhan S.S., Jean-Marie Proth. 2003. The concave cost supply problem. *European Journal of Operational Research* **148** 374-383.

Crama Y., J. R. Pascual, A. Torres. 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., N.C. Petruzzi, L.B. Schwarz. 2007. A newsvendors procurement problem when suppliers are unreliable. *Manufacturing & Service Operations Management* **9** 9-32.

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

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

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

Laporte G., J. Riera-Ledesma, J. J. Salazar-Gonzlez. 2003. A branch-and-cut algorithm for the undirected traveling purchaser problem. *Operations Research* **51**(6) 940–951.

Mansini R., M. Savelsbergh, B. Tocchella. 2011. The Supplier Selection Problem with Quantity Discounts and Truckload Shipping (submitted).

McConnel M.B., D. T. Galligan. 2004. The use of Integer Programming to select bulls across breeding companies with volume discounts. *Journal of Diary Science* **87** 3542–3549.

Mohammad Ebrahim R., J. Razmi, H. Haleh. 2009. Scatter search algorithm for supplier selection and order lot sizing under multiple price discount environment. *Advances in Engineering Software* **40** 766–776.

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

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

Rosenblat M.J., Y.T. Herer, I. Hefter. 1998. An acquisition policy for a single item multi-supplier system. *Management Science* **44** 96-100.

van de Klundert J., J. Kuipers, F. C. R. Spieksma, M. Winkels. 2005. Selecting Telecommunication Carriers to Obtain Volume Discounts. *Interfaces* **35** 124–132.

Weber C.A., J.R. Current, W.C. Benton. 1991. Vendor selection criteria and methods. *European Journal of Operational Research* **50** 2–18.

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

Zhang J., M. Zhang. 2011. Supplier selection and purchase problem with fixed cost and constrained order quantities under stochastic demand. *Int. J. Production Economics* **129** 1–7.