An effective hybrid heuristic for the Capacitated Total Quantity Discount Problem

Daniele Manerba, Renata Mansini

Department of Information Engineering, University of Brescia, Italy

Technical Report 2011-03-67, August 2011.

Abstract

A company needs to purchase a number of products from a set of suppliers that offer products at given prices applying quantity discounts. The Total Quantity Discount Problem (TQDP) is a procurement problem that looks for a subset of suppliers so to minimize purchasing cost while satisfying products demand. We consider the capacitated variant of the problem (CTQDP) where quantity of a product available from a supplier is limited. The problem is strongly NP-hard motivating the study of effective and efficient heuristic methods. We introduce an hybrid algorithm, called `HELP`, exploiting the optimal solution of mixed integer subproblems. The method, while conceived for this special problem, results to be a general approach that could be effectively adapted also to other combinatorial optimization problems. Proposed method has been tested on existing benchmark instances and has been able to improve best known values.

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

1 Introduction

In the last few decades, the supplier (or vendor) selection problem has gained increasing attention in both practice and literature. The general problem is usually formalized as two joint decisions on which suppliers should be selected and on how much should be ordered from the selected suppliers.

The CTQD problem is a supplier selection problem where suppliers apply volume discounts depending on the total quantity ordered. The problem can be described as follows: Let $S := \{1, \ldots, n\}$ be a set of suppliers and let $K := \{1, \ldots, m\}$ be a set of products. 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 $R_i = \{1, \ldots, r_i\}$ 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 CTQDP problem generalizes the TQD problem described in Goossens et al. [2007], where suppliers are assumed to have unlimited availability for offered products. The authors provide a first mathematical formulation, show NP-hardness and some properties and study some variants of the problem. They also 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 used as a plain branch-and-bound 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 Manerba and Mansini [2011] the authors provide the first branch-and-cut approach to solve the CTQDP and define a set of benchmark instances including up to 100 suppliers and 500 products. The availability of exact solutions for such test instances makes it possible the evaluation of the performance of heuristic algorithms. To the best of our knowledge no stand-alone heuristic approaches have been proposed for the CTQDP. Heuristics can however be found for other procurement problems for both the case where product demands are assumed to be deterministic quantity as well as a stochastic one (see for example Rosenblat et al. [1998], Chauhan and Proth [2003] and Awasthi et al. [2009]).

The Travelling Purchaser Problem is a classical example of procurement problem involving both travelling and purchasing cost. In Riera-Ledesma and Salazar-Gonzàlez [2005] the authors propose a heuristic approach based on a local search scheme. In Angelelli et al. [2009] and Angelelli et al. [2011] the authors analyze a dynamic extension of the TPP where quantities offered by the suppliers are assumed to decrease over time. Two groups of heuristics are described and compared. The first group are simplified approaches based on greedy criteria, the second one includes heuristics based on a look-ahead approach taking future prediction into account. In Mansini and Tocchella [2009] a version of the TPP is analyzed where total purchasing costs should not exceed a predefined threshold. The authors provide an enhanced local search heuristic and a VNS tested in a multi-start variant. In Mansini R.

[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 study a heuristic method, enhancing the basic procedure called `HELP` initially proposed in Manerba and Mansini [2011]. The method uses information provided by LP relaxation optimal solution to formulate and solve exactly mixed integer subproblems. More precisely, the framework is a Variable Neighborhood Search with Decomposition (see Hansen et al. [2001] for general information about this approach). Iteratively, some features of the current solution are kept fixed and the problem is decomposed to a smaller subspace explored exactly through the optimal solution of the resulting mixed integer subproblem. The effectiveness of the heuristic is proved over the set of benchmark instances proposed in Manerba and Mansini [2011]. The optimal solution value for some of these instances is still unknown.

`HELP` is a hybrid method combining a meta-heuristic approach with the exact solution of MIP subproblems. Computing optimal solutions although computationally intractable can be a valid tool when the subproblem solved has a limited size. Hybrid meta-heuristics have received considerable interest these recent years. A wide variety of hybrid approaches have been proposed in the literature for different combinatorial optimization problems. Although exact and approximate approaches have usually been considered mutually exclusive, many hybrid methods combine features of exact and approximate methods (see for instance Glover et al. [1996] and Nagar et al. [1996]). More recently, French et al. [2001] propose a hybrid algorithm, combining genetic algorithms and integer programming branch-and-bound approaches, to solve MAX-SAT problems. Haouari and Ladhari [2003] analyze a branch-and-bound local search method for the flow shop problem and Woodcock and Wilson [2010] investigate a hybrid method for solving the Generalized Assignment Problem that uses a commercial software to solve sub-problems generated by a Tabu Search guiding strategy. In Framinan and Pastor [2008] the authors propose a procedure called Bound Driven Search that combines features of exact enumeration methods (i.e. branch-and-bound) with approximate procedures (i.e. local search). Finally, a taxonomy of hybrid meta-heuristics is discussed in Talbi [2002] in an attempt to provide a common terminology and classification mechanisms.

This work provides some interesting contributions. First of all, to the best of our knowledge, no heuristic methods exist for the problem. Then, proposed hybrid algorithm provides some basic ideas which can be easily generalized also to other metaheuristic approaches. The paper is organized as follows. In Section 2 we describe the mathematical formulation of the problem where a polynomial number of inequalities are added to strengthen the LP formulation. In Section 3 the new version of procedure `HELP` is analyzed. Computational results are described in Section 4. Finally, some conclusions are drawn in Section 5.

2 Mathematical Model

Let us define as $S^* := \left\{ i \in S : \text{ there exist } k \in K \text{ such that } \sum_{j \in S_k \setminus \{i\}} q_{jk} < d_k \right\}$ 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 use 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:

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

subject to

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

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

$$\sum_{r \in R_i} z_{ik}^r = q_{ik} \qquad i \in S, k \in K^* \tag{4}$$

$$\sum_{r \in R_i} y_i^r \leq 1 \qquad i \in S \setminus S^* \tag{5}$$

$$\sum_{r \in R_i} y_i^r = 1 \qquad i \in S^* \tag{6}$$

$$\sum_{k \in K} z_{ik}^r - l_i^r y_i^r \geq 0 \qquad i \in S, r \in R_i \setminus R_i^* \tag{7}$$

$$\sum_{k \in K} z_{ik}^r - u_i^r y_i^r \leq 0 \qquad i \in S, r \in R_i \setminus R_i^* \tag{8}$$

$$z_{ik}^r \leq \min\{q_{ik}, u_i^r\} y_i^r, \quad i \in S, k \in K, r \in R_i \setminus R_i^*. \tag{9}$$

$$z_{ik}^r \geq 0 \qquad i \in S, k \in K, r \in R_i \setminus R_i^* \tag{10}$$

$$z_{ik}^r = 0 \qquad i \in S, k \in K, r \in R_i^* \tag{11}$$

$$y_i^r \in \{0,1\} \qquad i \in S, r \in R_i \setminus R_i^* \tag{12}$$

$$y_i^r = 0 \qquad i \in S, r \in R_i^* \tag{13}$$

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) state that for each product $k \in K^*$ all quantities available have to be purchased to satisfy demand. Constraints (5) guarantee that at most one interval for each supplier is selected, and constraints (6) that a supplier $i \in S^*$ has to be selected. Constraints (7) and (8) 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 $\sum_{k \in K} z_{ik}^r = 0$. Valid inequalities (9) are not strictly required by problem formulation but are added to strengthen linear relaxation and because they are polynomial in number. Finally, constraints (10)–(13) state non-negativity and binary conditions. Note that, in problem formulation integrality of z variables is not necessary; this condition is always satisfied if all input data are integral. Moreover, since demand requirements can not be exceeded, each coefficient q_{ik} can be replaced everywhere by $min\{q_{ik}, d_k\}$. Valid inequalities (9) and just described properties of the problem can be found in Manerba and Mansini [2011].

3 Solution Algorithm

This section is devoted to deeply explain our solution algorithm. Heuristic Enhancement from LP (`HELP`) is a hybrid method, 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 $\bar{y}$. We can enunciate the following proposition:

Proposition 1 *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).*

Trivially, a solution generated as above is feasible since intervals are selected according to quantities purchased satisfying products demand. Notice that, since $l_i^0 = 0, i \in S$, if nothing has been bought from a supplier, then the first interval is selected. Moreover, subproblem $CTQDP(\bar{y})$ only involves continuous variables but provides integer optimal solutions.

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

Algorithm 1 HELP

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

$s^F \leftarrow s^I, v^F \leftarrow v^I$
while T_{MAX} is not reached **do**
  $h \leftarrow h_{START}$, $s^C \leftarrow s^I$, $v^C \leftarrow v^I$
  while $h > 0$ **do**
    Randomly select two sets of suppliers H and H' with cardinality h and $h' = \alpha h$ respectively.
    Define the neighborhood $N(s^C, h, h')$
    Construct the corresponding subproblem and optimally solve it
    Let s' be the optimal solution with value v'
    if $v' < v^C$ **then**
      $s^C \leftarrow s'$, $v^C \leftarrow v'$
      $h \leftarrow h_{START}$
      if $v' < v^F$ **then**
        $s^F \leftarrow s^C$, $v^F \leftarrow v^C$
      end if
    else
      $h \leftarrow h - h_{STEP}$
    end if
  end while
end while

The procedure iteratively solve MIP subproblems where only few y variables are not fixed, i.e. those belonging to sets of suppliers randomly chosen. More precisely, given a current feasible solution s^C and two random subsets of suppliers H and H', we create a new subproblem where the values of all y variables are fixed as in s^C but for those referring to all intervals of suppliers in H and for those corresponding to the interval selected in s^C and its subsequent (if any exists) for each supplier in H'. Formally, let h and h' be the cardinality of

H and H', respectively. We define the neighborhood $N(s^C, h, h')$ as all solutions that differ from s^C for at most h selected intervals in $\bigcup R_i, i \in H$ and for at most h' selected intervals in $\bigcup\{w_i, w_i+1\} \in R_i \setminus \{r_i\}, i \in H'$, where w_i is the interval selected in the current solution s^C at suppleir i.
The procedure starts by randomly selecting H and H' and defining the corresponding neighborhood $N(s^C, h, h')$, then the mixed integer subproblem containing only $O(\sum_{i\in H}|R_i| + 2|H'|)$ variables y and all z variables is formulated. The local search phase of the VNDS is implemented by optimally solving this subproblem with a MIP solver.

If the optimal solution $s' = (z', y')$ has a better value than the best incumbent solution for that descend s^C, then s^C is updated and h is reset to its starting value h_{START}. If s' has a better value than the best solution ever found s^F, then s^F is updated too. If no better solution is found h is decreased by h_{STEP} and the procedure is repeated until h reaches value 0 (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). Notice that at each restart the procedure uses the same starting solution s^I. In some preliminary experiments we have noticed that using the best solution found instead of the initial one is usually less effective since it could easily lead to get stuck in local minima. We must specify that each subproblem created and optimally solved does not include inequalities (9) since these are not strictly necessary and really overload the model affecting computational time. These inequalities are indeed very useful in the first LP relaxation to provide an initial feasible solution as close as possible, concerning the selection of discount intervals, to the optimal solution.

`HELP` has some intrinsic 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 previous integer feasible solution. Finally, the general approach used by our heuristic can be easily generalized and customized to other MIP problem in which solutions strongly depend on a set of binary decision variables.

4 Computational Results

In this section we analyze our algorithm's performance. We have tested `HELP` on the complete set of benchmark instances proposed in Manerba and Mansini [2011] and available at the web page *http://www.ing.unibs.it/∼orgroup/instances.html*. This set consists of 5 random instances for each combination of number of suppliers $n = \{50, 100\}$, number of products $m = \{100, 200, 300, 500\}$, $\lambda = \{0.1, 0.8\}$ and $Class = \{1, 2\}$. This means 160 instances all together. λ is a parameter strictly related to the number of suppliers in feasible solutions: $\lambda = 0.1$ and $\lambda = 0.8$ represent the cases 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. The two classes differ only for the discount policy: in Class 1 suppliers can have 3 to 5 discount intervals with random rates, whereas in Class 2 suppliers have exactly 3 intervals with discount rates fixed and equal to {0%,10%,50%}.

Table 1: Results for Class 1 instances

			$\lambda = 0.1$							$\lambda = 0.8$						
			Best		HELP					Best		HELP				
$\|S\|$	$\|K\|$	i	t_{BK}	v_{BK}	t_H	#	w%	m%	b%	t_{BK}	v_{BK}	t_H	#	w%	m%	b%
50	100	1	270	1065331.43	201	1	0.09	0.04	**0.00**	61	259244.06	50	3	0.06	0.02	**0.00**
50	100	2	270	1053147.65	202	1	0.11	0.02	**0.00**	83	239212.53	51	1	0.13	0.11	**0.00**
50	100	3	244	1063178.97	202	5	**0.00**	**0.00**	**0.00**	72	255411.50	51	5	**0.00**	**0.00**	**0.00**
50	100	4	234	933882.86	201	2	0.01	0.00	**0.00**	60	245170.07	50	5	**0.00**	**0.00**	**0.00**
50	100	5	234	994708.65	201	2	0.04	0.01	**0.00**	68	257293.66	50	5	**0.00**	**0.00**	**0.00**
50	200	1	523	1952152.07	404	3	0.01	0.00	**0.00**	201	504369.86	102	2	0.62	0.22	**0.00**
50	200	2	607	2132512.11	409	1	0.19	0.08	**0.00**	199	501271.64	104	2	0.14	0.04	**0.00**
50	200	3	522	1998714.50	406	5	**0.00**	**0.00**	**0.00**	177	546469.73	102	3	0.17	0.04	**0.00**
50	200	4	508	2103277.09	405	2	0.09	0.05	**0.00**	175	520573.11	101	3	0.23	0.09	**0.00**
50	200	5	517	1935851.94	409	2	0.18	0.08	**0.00**	164	475366.42	101	5	**0.00**	**0.00**	**0.00**
50	300	1	958	2710818.62	607	3	0.10	0.02	**0.00**	338	810963.79	157	3	0.13	0.05	**0.00**
50	300	2	772	2762370.39	609	4	0.04	0.01	**0.00**	198	801325.23	152	5	**0.00**	**0.00**	**0.00**
50	300	3	975	3150592.95	610	0	0.03	0.02	0.02	264	754396.40	153	4	0.27	0.05	**0.00**
50	300	4	1093	3180894.49	604	1	0.53	0.14	**0.00**	208	785817.78	155	5	**0.00**	**0.00**	**0.00**
50	300	5	805	3206291.08	612	3	0.08	0.02	**0.00**	204	743753.17	151	3	0.25	0.10	**0.00**
50	500	1	1571	5087519.13	1076	1	0.25	0.11	**0.00**	738	1320857.89	274	4	0.12	0.02	**0.00**
50	500	2	1534	4816132.22	1057	3	0.03	0.01	**0.00**	491	1267401.38	255	2	0.99	0.34	**0.00**
50	500	3	1711	5073596.64	1023	0	0.23	0.08	0.02	377	1347170.84	257	5	**0.00**	**0.00**	**0.00**
50	500	4	1517	4976074.66	1083	1	0.74	0.32	**0.00**	436	1219496.74	269	4	0.52	0.10	**0.00**
50	500	5	1678	5053157.68	1023	0	0.16	0.07	0.02	400	1228369.20	264	4	0.41	0.08	**0.00**
100	100	1	456	2040173.12	401	4	0.00	0.00	**0.00**	184	459820.43	101	2	0.63	0.24	**0.00**
100	100	2	481	1912898.23	401	3	0.02	0.01	**0.00**	204	422275.23	101	1	0.17	0.09	**0.00**
100	100	3	453	1873337.80	402	5	**0.00**	**0.00**	**0.00**	149	445686.18	101	1	0.12	0.04	**0.00**
100	100	4	503	1922911.67	402	2	0.06	0.02	**0.00**	156	472285.13	101	0	0.95	0.40	0.03
100	100	5	536	2155736.71	402	2	0.01	0.01	**0.00**	207	416908.89	101	2	0.36	0.15	**0.00**
100	200	1	873	4360648.52	607	0	0.15	0.05	0.01	570	883411.52	204	3	0.03	0.01	**0.00**
100	200	2	933	4154529.60	605	0	0.11	0.05	0.00	501	875510.12	202	2	0.31	0.19	**0.00**
100	200	3	1100	3797749.45	616	1	0.06	0.03	**0.00**	391	895753.50	202	1	0.64	0.20	**0.00**
100	200	4	879	3926225.47	610	1	0.02	0.01	**0.00**	358	883419.34	203	5	**0.00**	**0.00**	**0.00**
100	200	5	962	3668949.23	611	1	0.01	0.00	**0.00**	421	878740.30	204	1	0.63	0.36	**0.00**
100	300	1	1504	5719285.13	812	2	0.02	0.01	**0.00**	2941	1429177.24	319	0	0.54	0.29	0.03
100	300	2	1429	5711788.37	813	0	0.18	0.08	0.01	1241	1433131.13	317	0	1.71	1.00	0.58
100	300	3	1763	5584860.56	821	0	0.10	0.04	0.01	1098	1365427.33	312	3	0.52	0.15	**0.00**
100	300	4	1583	6143443.91	821	1	0.07	0.02	**0.00**	1480	1282178.92	301	0	0.55	0.30	0.16
100	300	5	1338	6135946.34	814	0	0.08	0.04	0.01	1221	1255026.29	308	0	1.08	0.80	0.41
100	500	1	2666	9240585.56	1238	0	0.50	0.24	0.01	3591	2264190.57	543	0	0.62	0.34	0.11
100	500	2	2621	9971412.11	1269	0	0.14	0.07	0.01	4035	2290165.46	542	0	0.40	0.21	0.02
100	500	3	3317	9861834.02	1232	0	0.21	0.09	0.01	4808	2229422.19	544	0	2.58	1.73	0.66
100	500	4	5319	9632126.38	1268	1	0.27	0.11	**0.00**	3644	2025046.59	514	0	1.32	0.57	0.04
100	500	5	5143	9851746.34	1282	0	0.42	0.27	0.14	3976	2218089.75	530	0	3.33	1.92	0.92
							0.13	0.06	0.01					0.51	0.26	0.07

All computational tests have been done on an *Intel Core Duo* 2 GHz computer, with 2 GByte RAM and running *Windows Vista* 32-bit operating system. This is the same machine used for testing the exact methods proposed in Manerba and Mansini [2011]. After some preliminary tests, we have set $h_{START} = 10$, $h_{STEP} = 2$, $\alpha = \lfloor \frac{n}{100} \rfloor$ and a time limit $T_{max} = \phi_1|K| + \phi_2\lfloor |S|/100 \rfloor$, where $\phi_1 = 1/2$ and $\phi_2 = |K|/2$ 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

Table 2: Results for Class 2 instances

			$\lambda = 0.1$							$\lambda = 0.8$						
			Best		HELP					Best		HELP				
$\|S\|$	$\|K\|$	i	t_{BK}	v_{BK}	t_H	#	w%	m%	b%	t_{BK}	v_{BK}	t_H	#	w%	m%	b%
50	100	1	1954	888275.88	202	0	0.62	0.45	0.09	526	209340.00	201	3	2.65	0.97	**0.00**
50	100	2	573	750361.55	202	0	0.86	0.48	0.10	383	221929.00	202	2	1.45	0.62	**0.00**
50	100	3	442	784005.14	202	0	1.12	0.75	0.50	419	204096.50	204	1	2.44	1.37	**0.00**
50	100	4	487	828232.43	203	0	0.64	0.46	0.25	450	233650.00	203	2	0.96	0.42	**0.00**
50	100	5	483	752930.08	202	1	1.01	0.65	**0.00**	328	208702.61	201	2	0.21	0.12	**0.00**
50	200	1	4258	1706913.61	412	0	1.16	0.77	0.42	1254	425279.00	401	4	1.34	0.27	**0.00**
50	200	2	4530	1542150.28	409	0	0.85	0.67	0.41	1628	470957.00	411	0	2.11	1.32	0.79
50	200	3	2414	1554503.81	403	0	1.88	1.11	0.56	1084	387964.50	409	4	0.90	0.18	**0.00**
50	200	4	6951	1703115.96	406	0	1.07	0.91	0.66	1676	405405.50	406	3	0.08	0.03	**0.00**
50	200	5	6888	1569847.52	406	0	1.15	0.91	0.61	663	416658.50	402	5	**0.00**	**0.00**	**0.00**
50	300	1	7793	2447712.05	615	0	1.70	1.18	0.79	2841	632539.00	614	3	2.42	0.67	**0.00**
50	300	2	7008	2461007.61	613	0	1.41	1.12	0.87	6981	599533.50	603	5	**0.00**	**0.00**	**0.00**
50	300	3	15027	2333156.03	630	0	2.54	1.99	1.44	2762	657022.00	610	4	0.27	0.05	**0.00**
50	300	4	4371	2164848.92	609	0	1.02	0.91	0.73	3760	624807.00	615	2	1.62	0.91	**0.00**
50	300	5	15002	2358755.50	614	0	1.45	0.79	0.22	4418	631879.00	610	3	0.81	0.28	**0.00**
50	500	1	15519	4099394.12	1049	0	0.72	0.59	0.43	1430	1172101.50	1079	2	0.69	0.25	**0.00**
50	500	2	15420	3847179.41	1054	0	1.89	1.34	0.94	11830	978685.50	1043	3	0.63	0.22	**0.00**
50	500	3	14777	4037572.32	1048	0	3.91	2.60	1.85	15407	1178217.00	1028	1	1.43	0.86	**0.00**
50	500	4	15403	3860036.57	1031	0	0.59	0.34	0.08	1420	1135915.50	1022	5	**0.00**	**-0.15**	**-0.25**
50	500	5	15410	3846026.49	1022	0	1.42	0.77	0.39	15440	1069179.00	1019	2	0.36	0.14	**-0.16**
100	100	1	835	1560407.87	406	0	1.46	1.27	1.02	1553	364230.50	404	2	0.41	0.23	**0.00**
100	100	2	3505	1430206.02	404	0	0.84	0.59	0.40	3290	383295.50	402	0	2.75	1.60	0.88
100	100	3	2280	1655161.62	404	0	1.04	0.87	0.76	1774	367575.50	402	0	1.94	1.09	0.60
100	100	4	1596	1459152.50	403	0	1.15	0.85	0.60	1729	387625.50	402	0	3.34	2.41	0.83
100	100	5	1313	1570368.95	403	0	2.13	1.30	0.92	1874	374945.00	403	0	2.25	1.72	1.05
100	200	1	15047	3386881.90	608	0	1.10	0.81	0.62	11584	708833.50	612	0	5.00	2.92	1.91
100	200	2	15004	3353533.05	609	0	0.18	0.14	0.09	15061	748779.00	614	0	1.70	1.10	0.13
100	200	3	15021	3359507.13	612	0	0.62	0.50	0.37	15080	775076.00	616	0	4.04	3.14	1.54
100	200	4	10297	3285983.24	621	0	2.20	1.44	0.90	5004	768933.00	615	0	4.36	3.93	3.65
100	200	5	15048	3254487.50	611	0	1.85	1.46	0.89	15007	765640.00	607	0	2.28	2.08	1.91
100	300	1	15340	4867747.01	831	0	1.60	1.18	0.86	15271	1096232.00	813	0	5.18	4.01	2.02
100	300	2	15326	4918446.29	815	0	1.84	1.54	0.77	15265	1164793.00	842	0	2.99	2.63	2.35
100	300	3	15309	4680882.21	842	0	2.20	1.44	0.80	15212	1158439.50	837	0	4.91	4.45	3.79
100	300	4	15287	4942760.09	818	0	2.37	1.14	0.59	15293	1187005.00	831	0	0.52	0.27	0.08
100	300	5	15269	4852938.98	857	0	2.46	1.58	0.75	15262	1214561.00	837	0	2.30	1.42	0.93
100	500	1	15636	8202625.72	1268	0	2.64	1.96	1.62	15721	1821572.00	1382	0	5.24	4.18	2.00
100	500	2	15704	8252585.36	1278	0	4.01	2.09	0.74	15689	1827191.50	1283	0	2.24	1.60	0.87
100	500	3	15609	8522604.60	1279	0	1.29	0.87	0.14	993	1917578.50	1288	4	0.56	**-0.04**	**-0.32**
100	500	4	15611	8805448.70	1262	0	1.83	0.95	0.29	15975	1888163.50	1331	0	4.60	3.48	0.82
100	500	5	15906	8277027.11	1299	1	1.76	0.47	**-0.06**	1026	1851845.00	1239	0	5.20	3.75	1.87
							1,54	1,03	0,61					2,05	1,36	0,68

additional bonus time when n =100. Exact local search is implemented by CPLEX 12.2 and C++ Concert Technology.

Because of random structure of `HELP` algorithm, we decide to compute 5 trials for each instance. Averaged results are presented in Tables 1 and 2 (one table for each Class 1 and 2). Each table is divided into two parts, one for each λ value. For each instance (uniquely identified by the number of suppliers $|S|$, the number of products $|K|$ and the number of instance i), each table shows the following columns:

- t_{BK}: computational time in seconds of the best solution known;
- v_{BK}: value of the best solution known;
- t_H: computational time in seconds of HELP, averaged on 5 trials;
- #: number of times that HELP reaches or improves the best solution known;

- $w\%$: worst percentage gap from the value of the best solution known;
- $m\%$: percentage gap, averaged on 5 trials, from the value of the best solution known;
- $b\%$: best percentage gap from the value of the best solution known.

In these last three columns, if the solution value found by `HELP` is equal or better than the current best known value (i.e. the percentage gap is less or equal to 0) we have highlighted the figure in bold font.

A first sight analysis confirms (see Manerba and Mansini [2011] that Class 2 instances are more difficult to solve than Class 1, and $\lambda = 0.1$ instances are harder than $\lambda = 0.8$ ones. Considering Class 1, `HELP` results a heuristic with impressive performance: for 58 instances out of 80, `HELP` finds the optimal solution in at least one trial and for 11 instances among these the heuristic gets the optimal solution in all the five trials. In all these cases computational times for `HELP` are smaller than the times t_{BK} required by the exact approach. When the procedure can not reach the optimal solution, gaps from the best knows values are modest anyway, sometimes irrelevant. Indeed, the worst percentage gaps never overtake 0.74% for $\lambda = 0.1$ and 3.33% for $\lambda = 0.8$, and the worst averaged gaps on 40 instances are 0.13% for $\lambda = 0.1$ and 0.51% for $\lambda = 0.8$. Results for Class 2 are less strong, but however still very good. `HELP` is able to reach or improve best known values in at least one trial for 23 instances out of 80, and for 3 instances among these the heuristic reaches this goal for all the five trials. In all these cases computational times are largely lower than t_{BK} values. In this set of instances, the worst percentage gaps reach 4.01% for $\lambda = 0.1$ and 5.24% for $\lambda = 0.8$, and the worst averaged gaps on 40 instances are 1.54% for $\lambda = 0.1$ and 2.05% for $\lambda = 0.8$. Medium averaged gaps for the two values of λ settle around to 1%, anyway. Hence, the heuristic has some discontinuous performance for Class 2 instances, mostly for the hardest ones (100 suppliers), but if we compare computational times the gaps are enormous. In fact most part of solutions for Class 2 instances need about 15000 seconds (more than 4 hours) to be found by an exact method, while `HELP` never exceed 1382 seconds of running time. Finally, a very positive note of these results is that for 4 instances `HELP` has found a better solution than the best known (i.e. negative percentage gaps).

To conclude `HELP` is an efficient and effective heuristic method that results especially robust when complexity of instances increases (larger number of products and suppliers), thanks also to a good parameters setting.

5 Conclusions

In this paper an effective heuristic method is proposed to solve the CTQDP problem. The algorithm, called `HELP`, combines a VNS-like meta-heuristic structure with the exact solution of MIP subproblems. The method, tested on a large set of benchmark instances, is shown to be extremely effective (it has reached the best known value in 77 instances and improved it in 4 instances, out of 180) and quite efficient.

References

E. Angelelli, R. Mansini, and M. Vindigni. Exploring greedy criteria for the dynamic traveling purchaser problem. *Central European Journal of Operations Research*, 17:141–158, 2009.

E. Angelelli, R. Mansini, and M. Vindigni. Look-ahead heuristics for the dynamic traveling purchaser problem. *Computers & Operations Research*, 38:1867–1876, 2011.

A. Awasthi, S.S. Chauhan, S.K. Goyal, and Jean-Marie Proth. Supplier selection problem for a single manufacturing unit under stochastic demand. *International Journal of Production Economics*, 117:229233, 2009.

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

J. M. Framinan and R. Pastor. A proposal for a hybrid meta-strategy for combinatorial optimization problems. *Journal of Heuristics*, 14:375–390, 2008.

A.P. French, A.C. Robinson, and J.M. Wilson. Using a hybrid genetic-algorithm/branch and bound approach to solve feasibility and optimization integer programming problems. *Journal of Heuristics*, 7:551–564, 2001.

F. Glover, M. Parker, and J. Ryan. Coloring by tabu branch & bound. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26:285–307, 1996.

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

P. Hansen, N. Mladenovi, and D. Perez. Variable neighborhood decomposition search. *Journal of Heuristics*, 7:335–350, 2001.

M. Haouari and T. Ladhari. A branch-and-bound based local search method for the flow shop problem. *Journal of the Operational Research Society*, 54:1076–1084, 2003.

D. Manerba and R. Mansini. An exact algorithm for the capacitated total quantity discount problem. *Technical report - Department of Information Engineering - University of Brescia*, 2011.

R. Mansini and B. Tocchella. The traveling purchaser problem with budget constraint. *Computers & Operations Research*, 36:2263–2274, 2009.

B. Tocchella Mansini R., M. Savelsbergh. The supplier selection problem with quantity discounts and truckload shipping. *Submitted*, 2011.

A. Nagar, S.S. Heragu, and J. Haddock. A combined branch and bound and genetic algorithm based for a flowshop scheduling algorithm. *Annals of Operations Research*, 63:397–414, 1996.

J. Riera-Ledesma and J.J. Salazar-Gonzàlez. A heuristic approach for the travelling purchaser problem. *European Journal of Operational Research*, 162(1):142–152, 2005.

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

E.-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8:541–564, 2002.

A.J. Woodcock and J.M. Wilson. A hybrid tabu search / branch & bound approach to solving the generalized assignment problem. *European Journal of Operational Research*, 207(2):566–578, 2010.