Journal of the Operational Research Society (2011) 62, 690–699 0160-5682/11

www.palgrave-journals.com/jors/

Optimal supplier choice with discounting

B Goldengorin[1], J Keane[2], V Kuzmenko[3] and MK-S Tso[2]*

[1]University of Groningen, The Netherlands; [2]University of Manchester, Manchester, UK; and [3]Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine

This paper investigates a model for pricing the demand for a set of goods when suppliers operate discount schedules based on total business value. We formulate the buyers's decision problem as a mixed binary integer program, which is a generalization of the capacitated facility location problem (CFLP). A branch and bound (BnB) procedure using Lagrangean relaxation and subgradient optimization is developed for solving large-scale problems that can arise when suppliers' discount schedules contain multiple price breaks. Results of computer trials on specially adapted large benchmark instances of the CFLP confirm that a sub-gradient optimization procedure based on Shor and Zhurbenko's *r*-algorithm, which employs a space dilation in the direction of the difference between two successive subgradients, can be used efficiently for solving the dual problem at any node of the BnB tree.

Journal of the Operational Research Society (2011) **62,** 690–699. doi:10.1057/jors.2009.164
Published online 24 February 2010

Keywords: allocation; decision support systems; integer programming; location

1. Introduction

The rapid development of business-to-business electronic markets has triggered the need for efficient algorithms to allow an internet agent to source, that is to price in real time, an order for goods from multiple competing suppliers. The task is often complicated by two factors: (1) *fixed costs*, which serve to reduce the number of suppliers actually used to fulfill an order, and (2) *discounts schedules* offered by suppliers to encourage purchase of greater quantities. Such discount schedules may involve either the cancellation of a fixed charge, for example to cover carriage costs, or increasing percentage reductions off list price over a sequence of price breaks.

Our study is motivated by the requirements of an online supplier of pharmaceuticals to high-street chemists (retail pharmacists). The company acts as an internet broker in the sense that it carries no stock but, on receiving an enquiry (tentative order) for quantities of pharmaceutical products, it polls a set of wholesalers to determine which suppliers to use taking into account all applicable discounts. It then provides in real time a price quotation in answer to the enquiry based on cost, but including a profit mark-up. The enquiry is converted to a firm order if the total price of the basket of goods is judged to be acceptable by the customer. Thus the broker's task is to source the order at least cost. Note that the terms 'enquiry' and 'order' will be regarded as synonymous below.

**Correspondence: MK-S Tso, School of Mathematics, Alan Turing Building, University of Manchester, Oxford Road, Manchester, M13 9PL UK.*
E-mail: mtso@manchester.ac.uk

We assume that the company has online access to a negotiated 'static' price list for each supplier, and there is no aggregation of customers' enquiries or economies achievable through 'bulk buying', although in certain business contexts this would be an interesting possibility to consider.

We formulate below a general model for the '*buyer's decision problem*' (BDP) incorporating two common types of discount offered by suppliers: *Type A*: a cancellation or reduction in the fixed charge, and *Type B*: a percentage off the list price of each item. Discounts based on the value of an order have been termed a '*business volume*' discount (*BVD*) in contrast to a '*total quantity*' discount when price breaks are defined by *number of units* purchased (Goossens *et al*, 2004). Discounting may also take the form either of an '*all units*' policy modelled here, or an '*incremental*' policy. For a perspective on discounting theory and practice, see (Munson and Rosenblatt, 1998) where a field study on 39 firms is reported. An optimization model for vendor selection in the presence of price breaks was reported in (Chaudhry *et al*, 1993), though at that time there was no requirement for an online tool. The precise form of discounting employed will depend on the application, however the methods we develop for *BDP* can be easily generalized to other cases.

We show that the optimal allocation of a basket of goods to a set of suppliers in the market is an integer programming problem that resembles the capacitated facility location problem (CFLP). A supplier's price schedule represents a set of *per unit costs* and *fixed setup costs*. An interesting feature of our model is that the cost functions implied by the discount structures described here are distinctive, being discontinuous and in general neither concave nor convex. We note the resemblance to 'staircase cost functions' proposed for modelling production and distribution costs when a plant can be constructed in a range of sizes on a single site incurring scale-dependent setup (and running) costs (Holmberg, 1994), (Holmberg and Ling, 1997). Unlike our situation however, their fixed costs *increase* monotonically with plant capacity.

The fast response time required in an online context motivates the need for an efficient computational procedure for solving BDP. Such a procedure should be capable of rapidly solving large instances with possibly scores of suppliers offering hundreds of products and operating discount schedules containing multiple breakpoints. Although the efficient solution of large-scale instances of CFLP has been the subject of much research over many years (Cornuejols *et al*, 1991), (Agar and Salhi, 1998), (Bornstein and Azlan, 1998), the development of more efficient solution heuristics for large-scale problems remains an area of active research, see for example (Barahona and Chudak, 2005) and (Klose and Görtz, 2007). Such research has focussed almost exclusively on Lagrangean relaxation techniques, see (Krarup and Bilde, 1977), (Beasley, 1988) and (Körkel, 1989). In this paper, we report the results of investigating several new heuristics for finding tight lower bounds for the Lagrangean dual problem (LDP) for BDP involving subgradient optimization. A novel feature of our study is the use of the '*r-algorithm*' proposed by Shor and Zhurbenko (Shor, 1998), (Shor and Zhurbenko, 1971), which employs space dilation techniques to implement the subgradient optimization. We note the recent development of memoryless space dilation techniques in (Sherali *et al*, 2001) and related work by (Wu *et al*, 2006) in the context of CFLP with general setup costs.

An outline of our paper follows. We first give in Section 2 some motivating examples to further illustrate the practical context of our study. In Section 3, we formulate the BDP as a mixed integer linear program assuming an *all units BVD* discount policy. We show that a transformation making use of '*pseudosuppliers*' results in a variant of the CFLP that may be solved for large-scale instances (involving many suppliers and multiple price breakpoints) by an efficient subgradient procedure, the *r*-algorithm. Such a procedure, which employs the geometric concept of space dilation in the direction of two successive subgradients, represents an alternative to the 'classical' subgradient approach (Krarup and Bilde, 1977), as implemented for example by (Beasley, 1988). In Section 4, we formulate the LDP and provide details of an efficient solution method. Details of a BnB implementation including two new branching rules and fathoming heuristics are provided. We discuss the specialization to the BDP formulation with pseudosuppliers of the 'open' and 'close' penalties developed in (Khumawala, 1972), (Akinc and Khumawala, 1977) and further elaborated in (Beasley, 1988). Section 5 presents the results of computer experiments on a number of generated instances (a) to compare solution efficiencies of the *r*-algorithm with a classical subgradient method, and (b) to examine the time requirements of solution procedures under different levels and types of discounting. We conclude in Section 6 with a discussion and suggestions for future research.

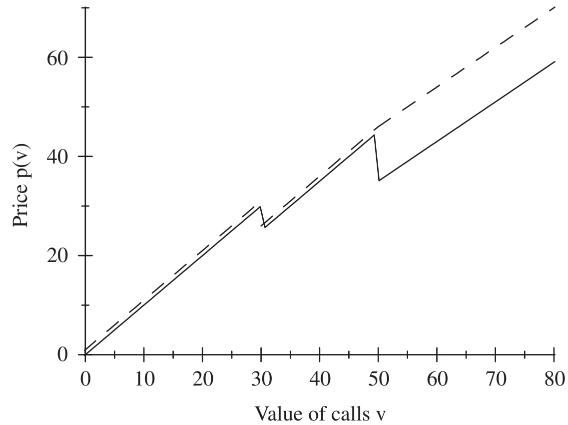

Figure 1 BVD 'sawtooth' discount function.

2. Motivating examples

2.1. Example 1

A mobile phone company announces that customers making over £30 of calls in a month will receive a refund of £5. In addition, customers making over £50 of calls in a month will qualify for a 20% discount from published tariffs. The discount schedule contains three piecewise linear segments created from two price breakpoints at £30 and £50, representing '*all units*' discounts of *Type A* and *Type B*, respectively. The price $p(v)$ of calls made to a total value v is given by

$$p(v) = \begin{cases} v & \text{if } v < 30 \\ v - 5 & \text{if } 30 \leqslant v < 50 \\ 0.8v - 5 & \text{if } 50 \leqslant v \end{cases}$$

and the graph given by the bold line in Figure 1 has the typical sawtooth form (Sadrian and Yoon, 1994). From this graph we also observe the '*more for less*' phenomenon (Goossens *et al*, 2004) that it can be cheaper to make more calls if the value of telephone business is just less than either breakpoint, in order to benefit from the next discount regime. By contrast the dotted graph in Figure 1 represents the price-value relationship under an '*incremental*' discount scheme in which only calls made above the £50 breakpoint qualify for the reduced tariff.

2.2. Example 2

Prices for packs of tulips and roses from two florists are given in Table 1. Delivery charges are £10 for florist 1 and £5 for florist 2. Charges are waived on orders over £50 in value. Consider two separate orders: (a) for seven packs of tulips and three packs of roses, and (b) for seven packs of tulips only.

Table 2 contains the optimal 'transportation' matrices indicating how each order should be fulfilled at minimum cost. The bracketed entries in the final column represent the 'total business value' of the suborder from each supplier, used to

Table 1 Florists' price lists (£)

	Delivery charge	*Price per pack*		*Discount breakpoint*
		Tulips	*Roses*	
Florist 1	10	7	6	50
Florist 2	5	9	3	50

Table 2 Minimum cost orders

	Delivery charge	*No. of packs and total cost*		*Sub-order value*
		Tulips	*Roses*	
(a) Order for seven packs of tulips, three packs of roses				
Florist 1	0	7	1	(55)
Florist 2	5	0	2	(6)
Cost (£)	5	49	12	66
(b) Order for seven packs of tulips				
Florist 1	0	7	1	(55)
Florist 2	0	0	0	(0)
Cost (£)	0	49	6	55

determine whether a discount applies. In the optimal solution to (a) costing £66, packs of roses are purchased from both florists—illustrating that use of a single source is suboptimal. In the optimal solution to (b) costing £55, the demand is exceeded by ordering an additional single pack of roses from the first florist—illustrating the '*more for less*' phenomenon.

3. The buyer's decision problem (BDP)

Consider a set of n products (items) indexed by $j \in J$ and suppose that each item j can be supplied by a common set of suppliers $s \in S$ at unit cost c_j^s from supplier s. We refer to $c^s = (c_1^s, \ldots, c_n^s)$ as the *list price* for items supplied by supplier s. Assume that each supplier offers discounts based on the total value of the order placed with that supplier computed according to the supplier's list price. The amount and nature of the discount is governed by a sequence of q price bands. If we suppose, for ease of notation, that q is the same for each supplier, then given a set of breakpoints $V_0^s < V_1^s < \cdots < V_q^s$ the price bands for supplier s are consecutive intervals $I_1^s = [V_0^s, V_1^s), \ldots, I_q^s = [V_{q-1}^s, V_q^s)$. We will assume without loss of generality that $V_0^s = 0$ and $V_q^s = \infty$. An order placed with supplier s for x_j^s units of item $j\,(j \in J)$ forms a vector $x^s = (x_1^s, \ldots, x_n^s)$ which has *total business value* (TBV) given by $v^s(x^s) = \sum_{j\in J} c_j^s x_j^s$.

The discounted price $\pi^s(x^s)$ that supplier s charges for the order is assumed to take the form

$$\pi^s(x^s) = f_k^s + \rho_k^s \sum_{j\in J} c_j^s x_j^s, \qquad \text{if } v^s(x^s) \in I_k^s$$

$$= f_k^s + \sum_{j\in J} c_{kj}^s x_j^s, \quad \text{say}$$

for given constants $\{\rho_k^s\}, \{f_k^s\}, k \in K = \{1, \ldots, q\}$.

We usually expect that higher order values will attract greater levels of discount so that $1 = \rho_1^s \geqslant \rho_2^s \geqslant \ldots$ and $f_1^s \geqslant f_2^s \geqslant \ldots$ in which case the price of suborder x^s is decreasing for successive price bands. We observe that, as a consequence of this monotonicity, only one price band per supplier will be utilized in an optimal solution. Note that $1 = \rho_1^s$ means that $c_j^s = c_{1j}^s$, that is the first price band $k = 1$ corresponds to list price.

The broker receives an order comprising a set of demands for D_j units of item $j\,(j \in J)$ and seeks to reallocate $d = (D_1, \ldots, D_n)$ among the suppliers at minimum cost. The set of suborders $\{x^s\}$ should meet the total demand

$$\sum_{s\in S} x^s \geqslant d \tag{1}$$

and minimize the total cost of supply

$$C(x) = \sum_{s\in S} \pi^s(x^s) \tag{2}$$

by taking advantage of all suppliers' discounts. Minimizing (2) will of course tend to restrict the number of suppliers. We refer to this cost optimization as the buyer's decision problem (BDP) and note that x^s may be real or integer, depending on the context of the application.

Let x_{kj}^s denote the quantity of item j ordered from supplier s in price band k. Now $v^s(x^s) = \sum_{j\in J} c_j^s x_j^s \in I_k^s$ for some $k = k_s$, say, and clearly $x_{kj}^s = 0$ whenever $k \neq k_s$, since it will be optimal only for one price band ever to be used for any supplier. Define

$$y_k^s = \begin{cases} 1 & \text{if } V^s(x^s) \in I_k^s \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and re-index

$$y_1^1, \ldots, y_q^1; \quad y_1^2, \ldots, y_q^2; \ldots; \quad y_1^s, \ldots, y_q^s$$

as a single sequence $\{y_i\}_{i\in I}$. We henceforth refer to $I = \{1, \ldots, m\}$ as a set of *pseudosuppliers*. A corresponding re-indexing of $f_k^s = f_i$, $c_{kj}^s = c_{ij}$, $x_{kj}^s = x_{ij}$ and $I_k^s = [L_i, U_i)$ leads to the following generalized CFLP in which the capacity constraints represent lower and upper bounds on TBV.

$$\text{BDP:} \quad \min_{y_i, x_{ij}} \sum_{i\in I} \left\{ f_i y_i + \sum_{j\in J} c_{ij} x_{ij} \right\} \tag{4}$$

$$\text{s.t.} \quad L_i y_i \leqslant \sum_{j\in J} c_{ij}^0 x_{ij} \leqslant U_i y_i \qquad \forall i \in I, \tag{5}$$

$$\sum_{i\in P^s} y_i \leqslant 1 \qquad \forall s \in S, \tag{6}$$

$$\sum_{i\in I} x_{ij} \geqslant D_j \qquad \forall j \in J, \tag{7}$$

$$y_i \in \{0, 1\} \qquad \forall i \in I, \tag{8}$$

$$x_{ij} \in \mathbb{Z}^+ \qquad \forall i \in I, \forall j \in J, \tag{9}$$

where P^s is the set of pseudosuppliers i corresponding to real supplier s, and $c_{ij}^0 = c_{1j}^s$ is the unit list price of item j for pseudosupplier i for $i \in P^s$.

The objective function (4) is precisely that of a standard (either simple or capacitated) plant location model. Constraint (5) states that the TBV of the suborder supplied from pseudosupplier i should fall within the appropriate price band. As with the CFLP model proposed in (Beasley, 1988), the constraint (5) places both lower and upper bounds on the supply from pseudosupplier i. However in BDP the constraint (5) represents limits on *total value* rather than *total demand*. Constraint (6) ensures that at most one price list per supplier can appear in an optimal solution. The demand constraint (7) may be expressed either as an equality or as a lower bound. Our use of an inequality corresponds to the 'more-for-less' formulation mentioned by (Goossens *et al*, 2004), Section 4.2., which allows for the possibility that it may be cheaper to over-fulfill demand in order to benefit from a higher level of discount. Due to the assumed monotonicity in pricing, a given bundle of goods from any supplier will be optimally supplied using the single (highest) discount band k.

We may compare the BDP model with the usual formulation of CFLP, see for example (Beasley, 1988). In CFLP the variable x_{ij} $(0 \leqslant x_{ij} \leqslant 1)$ is defined as the *fraction* of total demand d_j (for a single commodity) from customer j supplied by a warehouse i. The contribution to the total cost is $c_{ij}x_{ij}$ where c_{ij} is cost to supply 100% of d_j to customer j from warehouse i. There is also a fixed cost f_i to open warehouse i. By contrast in our model x_{sj} represents the *actual* amount of product j from supplier s and c_{sj} the corresponding unit cost of product j. Both the fixed cost f_s and the variable cost c_{sj} are potentially discounted depending on the *total value* $v^s(x^s)$ of the suborder supplied by supplier s calculated from that supplier's list price. Our cost function is in general discontinuous and can be compared to the staircase cost functions considered by (Holmberg, 1994). The minimization of a discontinuous function that is either piecewise linear or concave is generally not well behaved and rather than deal with such functions explicitly, we introduce the notion of a pseudosupplier (s, k) to denote supplier s restricted to operating within price band k. The re-indexing $(s, k) \to i$ which defines the ith pseudosupplier allows clear comparisons with CFLP.

We note that in practice there may be restricted availability of some product j from supplier s. We could therefore include in our formulation of *BDP* additional *stock* constraints of the form

$$x_{ij} \leqslant S_{ij}, \qquad i \in I, \quad j \in J \tag{10}$$

for some constants $\{S_{ij}\}$. However, for ease of technical exposition and in computational experiments described in Section 6, we have excluded such upper bound constraints from the standard formulation of our model.

Finally, we emphasize that while x_{ij} are integer variables in this formulation of BDP, in other applications x_{ij} may be continuous, so that (9) will be replaced by $x_{ij} \in \mathbb{R}^+$.

4. Lagrangean relaxation of BDP

To construct the Lagrangean relaxation of BDP we write the constraints (7) in the form $D_j - \sum_{i\in I} x_{ij} \leqslant 0$ and introduce as a vector of corresponding Lagrange multipliers, the dual variables $\lambda = (\lambda_1, \ldots, \lambda_n)$ where $\lambda_j \geqslant 0$, $j \in J$. The LDP corresponding to BDP can be stated as

$$\text{LDP:} \quad \max_{\lambda \geqslant 0} F_\lambda \tag{11}$$

where

$$F_\lambda = \min_{y_i, x_{ij}} \left\{ \sum_{i=1}^m f_i y_i + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - \lambda_j) x_{ij} + \sum_{j=1}^n D_j \lambda_j \right\} \tag{12}$$

subject to (5), (6), (8) and (9).

For each $\lambda \geqslant 0$ this Lagrangean dual subproblem is a relaxation of BDP, and decomposes into separable subproblems for pseudosuppliers that are easily solved. As F_λ is a concave piecewise-linear function of λ, we may employ subgradient techniques to seek a constrained maximum of (11), which provides the best lower bound for the BDP as a whole.

To define optimal values of the binary variables y_i we will use the BnB method and Lagrangean heuristics. As stated in (Beasley, 1993), the purpose of a Lagrangean heuristic is to generate a sequence of Lagrange multipliers (defining lower bounds) and a sequence of feasible solutions (defining upper bounds) for the LDP. Solving the corresponding relaxation LDP at any node of the BnB tree provides a lower bound to the true integer optimum at that node. At any node some variables

y_i are fixed at 0, some at 1 and others are undetermined. Accordingly at a general node we suppose, in the notation of (Akinc and Khumawala, 1977) and (Beasley, 1988), that I has been partitioned into the index sets K_0, K_1, K_2 such that

$$\begin{aligned} &y_i = 0, \quad i \in K_0, \quad y_i = 1, \quad i \in K_1, \\ &y_i \in \{0, 1\}, \quad i \in K_2. \end{aligned} \tag{13}$$

Thus K_0, K_1 are the pseudosuppliers fixed closed, open, respectively; K_2 are the undetermined pseudosuppliers. Then let $P_L = |K_1|$, $P_U = |K_1 \cup K_2|$ and add the explicit bounds

$$P_L \leqslant \sum_{i=1}^{m} y_i \leqslant P_U \tag{14}$$

on the total number of actual suppliers used. The solution to the Lagrangean relaxation (12) for prescribed λ together with additional constraints (13) and (14) can be reduced to solving two knapsack problems. The first knapsack problem performs the minimization over $\{x_{ij}\}$ for each non-closed pseudosupplier $i \in K_1 \cup K_2$. The contribution to the dual function (12) from pseudosupplier i if open is:

$$\alpha_i = f_i + \min_{\{x_{ij}\}} \sum_{j=1}^{n} (c_{ij} - \lambda_j) x_{ij} \tag{15}$$

subject to (5) where $x_{ij} \in \mathbb{Z}^+$. We note that solving the 1-D integer knapsack is NP-hard (Martello and Toth, 1990). However, since our aim in solving the relaxation is to find a lower bound to BDP it suffices to solve instead the *continuous* knapsack relaxation. The minimum of (15) subject to (5) over $x_{ij} \in \mathbb{R}^+$ is achieved by a greedy heuristic, in which we order the x_{ij} by non-decreasing value of the ratio $(c_{ij} - \lambda_j)/c_{ij}^0$ and set the components of x_{ij} in turn to their maximum value subject to (5). In case of infeasibility, we set $y_i = 0$ and $x_{ij} = 0$ for each j. The speed and simplicity of the solution $\{x_{ij}^*\}$ thus found is of course at the expense of an increased duality gap.

The second knapsack problem is a minimization problem on the set of Boolean variables $\{y_i\}$

$$F_\lambda = \min_{\{y_i\}} \sum_{i \in K_2} \alpha_i y_i + \sum_{i \in K_1} \alpha_i + \sum_{y=1}^{n} D_j \lambda_j \tag{16}$$

subject to (6) and (8). For each supplier s, let $\beta_s = \min_{i \in P^s \cap K_2} \{\alpha_i\}$ and form the corresponding list of pseudosuppliers $i_1, i_2, \ldots$ in non-decreasing order of β_s. Define the sequence of partial sums $\{\phi_t\}$ by

$$\begin{aligned} \phi_0 &= \sum_{i \in K_1} \alpha_i + \sum_{j=1}^{n} D_j \lambda_j \\ \phi_1 &= S_0 + \alpha_{i_1} \\ &\vdots \\ \phi_t &= S_{t-1} + \alpha_{i_t} \end{aligned} \tag{17}$$

The smallest value of ϕ_{t^*} such that $P_L \leqslant t^* \leqslant P_U$ gives an optimal solution y^* to (16) and provides a lower bound $Z_D = F_\lambda$ to the value of BDP at this node. Let Z_U be the value of the *incumbent*, that is of the best feasible solution found so far. We decide that the branch is fathomed if $Z_D \geqslant Z_U$, otherwise we continue to develop this node. Some further details of the algorithm are outlined below.

4.1. Reduction tests

This straightforward procedure for solving the dual subproblem by solving two knapsack problems provides the opportunity to check whether each y_i variable can be fixed at 0 or 1 in subsequent branchings. We describe in this section a modification of the reduction tests stated in (Christofides and Beasley, 1983) in order to prevent multiple discount levels for the same supplier appearing in an optimal dual solution.

Let M^* denote the set of pseudosuppliers that are open in the optimal solution to the LDP (12) subject to (14). Denote by $\sigma : I \rightarrow S$ the mapping of pseudosuppliers onto real suppliers. Let $S^* = \sigma(M^*)$ and let $N^* = \{j \in K_2 \backslash M^* : \sigma(j) \in S^*\}$ be the set of closed pseudosuppliers in K_2 that cannot be simultaneously open with the existing set of open pseudosuppliers M^*.

4.1.1. Open penalties For any closed pseudosupplier $i \in K_2 \backslash M^*$ we calculate the change in $Z_D = F_\lambda$ if pseudosupplier i is forced open.

Case 1: If $i \in N^*$ then $\sigma(i) = \sigma(k)$ for some already open pseudosupplier $k \in K_2 \cap M^*$ and we must close pseudosupplier k.
Otherwise ($i \notin N^*$) we consider three further cases.

Case 2: $|M^*| = P_U$. Set $k = \arg\max_{j \in K_2 \cap M^*} \{\alpha_j\}$ and close pseudosupplier k.

Case 3: $P_L < |M^*| < P_U$. Dual solution is otherwise unchanged.

Case 4: $|M^*| = P_L (P_L \neq P_U)$. Set k as in Case 2 and close pseudosupplier k if the objective function decreases as a result.

The new lower bound that results if pseudosupplier i is opened is then $Z_D + \omega_i$ where:

$$\omega_i = \begin{cases} \alpha_i - \alpha_k & \text{for Cases 1 and 2} \\ \alpha_i & \text{for Case 3} \\ \alpha_i - \max\{0, \alpha_k\} & \text{for Case 4} \end{cases} \tag{18}$$

4.1.2. Close penalties For $i \in K_2 \cap M^*$ we evaluate the change in Z_D as a result of closing pseudo-supplier i. Let $l = \arg\min_{j \in (K_2 \backslash M^*) \backslash N^*} \{\alpha_j\}$. If closing pseudosupplier i takes the cardinality of M below the lower limit then pseudosupplier l is forced to enter M^*. Otherwise pseudosupplier l enters only if an improvement in the dual objective results. The new lower bound that results from closing pseudosupplier i is $Z_D + \gamma_i$

where

$$\gamma_i = \begin{cases} -\alpha_i + \alpha_l & \text{if } |M^*| = P_L \\ -\alpha_i + \min\{0, \alpha_l\} & \text{if } |M^*| > P_L \end{cases} \tag{19}$$

4.1.3. Narrowing the bounds P_L, P_U Each ϕ_t in the sequence (17) represents the value of an optimal dual solution F_λ subject to the cardinality constraint $|M^*| = t$. Therefore F_λ is a lower bound for the original BDP with the constraint (14) replaced by the tight cardinality constraint $\sum_{i=1}^m y_i = t$. When solving the dual subproblem, it requires only a little additional computation to determine the maximal interval $[t_1, t_2]$ for which $F_\lambda < Z_U, \forall t \in [t_1, t_2]$. We can thereby reduce the interval $[P_L, P_U]$ to $[t_1, t_2]$. Note that $[t_1, t_2]$ will be non-empty since $\phi_{t^*} < Z_U$, otherwise this branch would have been fathomed on solving the dual subproblem.

4.2. Subgradient procedure

The family of '*r*-algorithms' (Shor and Stetsyuk, 2002) has been developed for the unconstrained maximization of concave objective functions over a continuous domain. The *r*-algorithm is a refinement of the classical subgradient method with space dilation developed by Shor and coworkers in the early 1970s (Shor, 1970), which Todd has identified as an example of a rank-one quasi-Newton method (Todd, 1986). The refinement employs space dilation in the direction of the difference between two successive subgradients. At the optimal solution $\{x_{ij}^*\}$ to the Lagrangean dual subproblem (12), the value of the supergradient is

$$\nabla_j F = D_j - \sum_{i=1}^m x_{ij}^*, \quad j = 1, \ldots, n$$

The non-negativity constraints $\lambda_j \geqslant 0$ can be incorporated by the use of a symmetrizing transformation $\lambda_j = |u_i|$ where $u_i \in \mathbb{R}$.

The computational efficiency of the *r*-algorithm depends on a choice of the space dilation coefficient α and adaptive tuning the values of the step multiplier (see Shor, 1998, p 104). We have applied the *r*-algorithm with α chosen in the interval [2,4]. This choice seeks to achieve as large as possible an improvement of the objective function along the current direction of search. In computer experiments reported below, timings are compared for the classical subgradient method and the *r*-algorithm on simulated instances. In our experiments, no speed improvements were introduced by the recently developed memoryless space dilation and reduction strategy of (Sherali *et al*, 2001).

4.3. Generating feasible solutions

The BnB algorithm requires a heuristic to generate good feasible solutions in order to prune the BnB tree efficiently. Given a set of open pseudosuppliers M^* from the solution to the LDP (11) we seek

$$Z(M^*) = \sum_{i \in M^*} f_i + \min_{x_{ij}} \sum_{i \in M^*} \sum_{j \in J} c_{ij} x_{ij} \tag{20}$$

subject to

$$L_i \leqslant \sum_{j \in J} c_{ij}^0 x_{ij} \leqslant U_i, \quad i \in M^* \tag{21}$$

$$\sum_{i \in M^*} x_{ij} \geqslant D_j, \quad j \in J \tag{22}$$

and constraint (9). This problem, whether x_{ij} is integer or real, is a generalized transportation problem with upper and lower bounds on the total value sourced from each pseudosupplier. In computer experiments, we have taken the above model with only the lower bound in (21) and used CPLEX to solve the continuous problem. The optimal solution is rounded up to the nearest integer solution. We then check whether the upper bound in (21) holds. If not, owing to the nature of our discount function we are able to find another pseudosupplier for which constraint (21) will be valid and for which the optimal value will be less.

4.4. Branching procedure

We have implemented two new heuristics for choosing the branching variable from among the undecided variables $\{y_i\}_{i \in K_2}$ that are non-integer valued in a solution to the relaxed LDP (11), (12). In our first approach we have used a heuristic procedure due to (Belyaeva *et al*, 1978) and not widely known, for estimating the optimal values of $\{y_i\}$. The calculation is based on an 'average'

$$y_i^* = \lim_{k \to \infty} \left\{ \frac{\sum_{t=1}^k h_t y_i^t}{\sum_{t=1}^k h_t} \right\} \tag{23}$$

of the values of y_i encountered during the subgradient iteration $\{\lambda^t\}$ that we presume converged to some optimal value λ^*. Here h_t and y_i^t are, respectively the step-length and value of y_i at step t of the iteration. If all y_i^* are integer, then no branching is needed. Otherwise we select the '*most non-integral*' y_i using the smallest value of $|y_i^* - \frac{1}{2}|$ as the criterion and branch first on the subproblem $y_i = 1$.

In our second approach that is also new, we determined the branching variable y_i through the open and close penalties (18), (19) obtained after solving each dual subproblem (12). During the sequence of iterations $\{\lambda^t\}$ let Z_D^t denote the dual bound F_λ (16) for $\lambda = \lambda^t$ and let ω_i^t and γ_i^t denote the corresponding values of (18), (19). The corresponding best lower bounds are

$$F_i^O = \max_t \{Z_D^t + \omega_i^t\} \quad \text{and} \quad F_i^C = \max_t \{Z_D^t + \gamma_i^t\} \tag{24}$$

We branch by the variable i for which $F_i^C - F_i^O$ is maximal. If $F_i^C \geqslant F_i^O$, then we set $y_i = 1$ (open). Otherwise we set $y_i = 0$ (closed). Our computational experiments on Set 3 instances

Table 3 Comparison of subgradient procedures with and without scaling

OR Lib instance	*No. suppliers*	*No. items*	*Scaling y/n*	*Classical subgradient*			*r-algorithm*		
				No. branchings	*No. iterations*	*Time*	*No. branchings*	*No. iterations*	*Time*
Cap 71	16	50	y	4	94	3	2	211	6
			n	21	1240	12	2	166	5
Cap101	25	50	y	6	43	4	2	201	7
			n	9	442	8	2	161	7
Cap131	50	50	y	9	115	5	4	250	15
			n	31	886	27	2	201	13

Table 4 Effect of discount type and level on execution times (s)

Cap 131 derived instance	*Timings (s)*			
	Classical		*r-algorithm*	
	Unscaled	*Scaled*	*Unscaled*	*Scaled*
A0 + B0	1.8	1.4	2	3
A25	191	62	133	147
A40	548	325	545	666
A50	1413	755	2492	2167
A60	1702	958	2231	2246
A75	1082	316	1037	645
A100	275	36	110	151
B10	3	3	4	3
B15	6	2	4	3
B20	13	5	4	3
A25 + B10	126	36	90	87
A50 + B10	106	44	43	46
A75 + B10	71	26	15	20
A100 + B10	50	6	7	7

below have shown that such an assignment of y_i leads very fast to a primal feasible solution that is close to an optimal solution.

5. Computational experiments

We have carried out computer experiments to compare the performance of the BnB procedure described above using two subgradient methods for finding lower bounds, the '*r*-algorithm' and the 'classical' subgradient approach of Christofides and Beasley (1983). In this comparison, three datasets were used. Problem sets 1 and 2 were adapted from three *uncapacitated facility location* problem instances derived from the OR benchmarks library (Beasley, 1990): *Cap 71*, *Cap 101* and *Cap 131*. Problem set 3 was adapted from the instance *Uniform-123* in the Library of Discrete Location Problems maintained by the Sobolev Institute of Mathematics in Novosibirsk (Kochetov and Ivanenko, 2003). It is known to be a hard instance. Programs were compiled using Compaq Visual Fortran 6 and run on a PC with Intel X86-735 Mhz processor under Microsoft Windows 2000.

5.1. OR Library—Set 1

This problem set is adapted from three instances in the OR benchmarks library: *Cap 71* with $m = 16$ plants, $n = 50$ customers; *Cap 101* with $m=25$, $n=50$; *Cap 131* with $m=n=50$. A single breakpoint at $V = 50\,000$ was introduced in all suppliers, when a fixed *Type A* discount of value 7500 became applicable. The range of variation of costs (unit prices) $\{c_{ij}\}$ is $[0, 1.4 \times 10^6]$ that of demands $\{D_j\}$ is $[31, 1.3 \times 10^4]$. We wished to compare the efficiency of the subgradient procedures and the sensitivity of each procedure to scaling of the the demands $\{D_j\}$.

Table 3 summarizes the comparison. Detailed are the number of branchings, the total number of iterations and computing times (s) for both methods. Each instance was solved with and without scaling the demands $\{D_j\}$ to unity.

We see that the number of branchings is uniformly less for the r-algorithm, indicating that the *r*-algorithm returns tighter lower bounds than the classical subgradient method. Solution times however are not markedly different due to greater overheads in computing subgradients for the *r*-algorithm. Scaling

Table 5 Exact and approximate solution times for 'Uniform-123' instances

Uniform-123 + discount type	*% Accuracy*	*Range factor D if scaled*	*Classic subgradient*		*r-algorithm*	
			Time (s)	*Value Z_U*	*Time (s)*	*Value Z_U*
A0 + B0	exact	unscaled	957	71 342*	1379	71 342*
A0 + B0	exact	2	2137	71 342*	1557	71 342*
A0 + B0	exact	1000	7065	71 342*	1612	71 342*
A0 + B0	1	unscaled	383	72 034	565	71 342*
A0 + B0	2	unscaled	147	72 606	314	72 080
A0 + B0	3	unscaled	61	73 377	78	73 422
		10	938	71 342	403	72 681
A30	3	unscaled	293	71 716	1596	71 344
A50	3	unscaled	903	68 815	2836	68 902
A50	5	unscaled	292	70 396	411	69 476
B20	3	unscaled	92	72 282	154	71 704
B50	3	unscaled	99	71 086	5442	72 375

before solution is effective for the simple subgradient method, but does not affect the performance of the r-algorithm.

5.2. *OR Library—Set 2*

Table 4 presents CPU timings (s) to solve instances from problem set 2. Each instance is derived from the OR Library dataset *CAP131* with $m = n = 50$ by adding different types and levels of discount to create two price bands. The level of discount offered in the second price band is indicated in the dataset name as a *percentage reduction*, respectively in fixed cost $\{f_i\}$ in the case of Type A and from the unit prices $\{c_{ij}\}$ in the case of Type B.

The results indicate that the r-algorithm generally takes more time to compute the bound than the classical subgradient method. The case A0 + B0 is the undiscounted case that corresponds to a simple plant location (SPLP) model without pseudosuppliers and is clearly very fast. Problems with Type A discounts require much more computing time to solve than instances with a purely Type B discount although the instance A100 that corresponds to the total removal of fixed costs for orders above the value threshold is also solved in less time. The most difficult cases are Type A instances with 50–60% discount. Timings are much reduced when both types of discount are applicable. This might be explained, intuitively, by the observation that when suppliers offers both types of discount simultaneously there is an incentive to use fewer suppliers, making the corresponding instance easier to solve.

5.3. *Sobolev Institute—Set 3*

For problem set 3, instances with $m = n = 100$ and different types/levels of discount were generated from a single dataset *Uniform-123* containing $m=100$ plants and $n=100$ customers with $D_j = 1$, each $j \in \{1, \ldots, n\}$. Each supplier was given a single price break with $f_1^s = 3000$ and $f_2^s = 0$, giving rise to 200 pseudosuppliers. D_j was either left unsealed or generated as uniformly distributed pseudorandom numbers in the range $[1, D]$ where the range factor D is tabulated (Table 5).

The results in Table 5 show CPU times (s) taken to find *exact* and *approximate* solutions to within a prescribed level of accuracy. The value of the best feasible solution found so far, Z_U, is also given with * indicating termination occurs at a true minimum.

We observe that the classical subgradient approach solves many of our instances more efficiently than the r-algorithm. However, solution times are generally of the same order of magnitude, although in one instance (B50;3%) the r-algorithm took more than 50 times longer. We note, however that the performance of the r-algorithm is also influenced by different choices of tuning parameters. When scaling of $\{D_j\}$ is performed so the demand for different products varies, we see an increase in the solution times using the simple sub-gradient algorithm. However the solution times by the r-algorithm remain much more stable. The largest solution times for approximate solutions are observed on instances with the pure Type A discounts of 50%.

6. Conclusions

We have addressed a general problem of importance in e-commerce, how to determine a minimum cost assignment of an order for a basket of goods to a set of suppliers taking into account fixed charges and some common types of discounting policies. The fast response time required in an online context motivates the need for an efficient computational procedure.

Our integer programming formulation may be regarded as a new type of capacitated facility location model for which solution procedures based on Lagrangean relaxation have been extensively studied. Two BnB algorithms have been implemented that differ in the method of computing the lower bounds. The first employs the dual-based Lagrangean heuristic based on the 'classical' subgradient method of

Krarup and Bilde (1977) and the second is the 'r-algorithm' based on space dilation in the direction of difference of two consecutive supergradients due to Shor and Zhurbenko (1971).

Computational experiments show that:

- Although the r-algorithm produces tighter bounds giving a reduced BnB tree, the classical subgradient algorithm achieved comparable solution times on test problems.
- The 'self-tuning' nature of the r-algorithm means however that solution times may be less sensitive to large differences in scale of the problem coefficients if this method is adopted.
- Discounts of Type A (change in fixed costs) may present more of a computational challenge than Type B (% change in variable cost).

We conclude that Lagrangean relaxation techniques can efficiently solve to optimality large-scale instances of the buyers decision problem involving many hundreds of suppliers and price lists containing multiple price breakpoints. On the basis of the instances solved, it is unclear whether the extra programming required to implement the r-algorithm will be justified by faster solution speeds.

The methods developed here are the basis for further studies into online sourcing of goods with different discounting policies. The issue of *bulking* or aggregating orders is one we have not addressed here. In practice it is common for suppliers to operate more complex discounting policies, for example based on the total value of orders aggregated over a time window. Such an environment requires an optimal strategy that evolves over time. A supplier faced with current competition in a specific market for products and services may make use of these optimal solutions as a tool for evaluating alternative price lists and discounting strategies. We foresee natural developments of our model motivated by recent research into the theory and practice of reverse auctions.

Finally, we note that optimization models involving linear and fixed transaction costs have recently been proposed for portfolio optimization in the financial context. Such models incorporate a stochastic dimension as the objective function coefficients are rates of return that are assumed to occupy a probability space (Lobo *et al*, 2007).

Acknowledgements— Thanks are due to the two anonymous referees whose perceptive comments led to a substantially improved manuscript. We also gratefully acknowledge the assistance provided by the Manchester Institute for the Mathematical Sciences (MIMS) in providing financial support to allow visits of the first and the third authors (BG, VK) to the University of Manchester in 2004 and 2006.

References

Agar MC and Salhi S (1998). Lagrangean heuristics applied to a variety of large capacitated plant location problems. *J Opl Res Soc* **49**: 1072–1084.

Akinc U and Khumawala BM (1977). An efficient branch and bound algorithm for the capacitated warehouse location problem. *Mngt Sci* **23**: 585–594.

Barahona F and Chudak FA (2005). Near-optimal solutions to large-scale facility location problems. *Discrete Opt* **2**: 35–50.

Beasley JE (1988). An algorithm for solving large capacitated warehouse location problems. *Eur J Opl Res* **33**: 314–325.

Beasley JE (1990). OR-Library: Distributing test problems by electronic mail. *J Opl Res Soc* **41**: 1069–1072.

Beasley JE (1993). Lagrangean heuristics for location problems. *Eur J Opl Res* **65**: 383–399.

Belyaeva LV, Zhurbenko NG and Shor NZ (1978). On a method of solving a class of dynamic distributive problems. *Econ Math Method (in Russian)* **14**: 137–146.

Bornstein CT and Azlan HB (1998). The use of reduction tests and simulated annealing for the capacitated plant location problem. *Location Sci* **6**: 67–81.

Chaudhry S, Forst F and Zydiak J (1993). Vendor selection with price breaks. *Eur J Opl Res* **70**: 52–66.

Christofides N and Beasley JE (1983). Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. *Eur J Opl Res* **12**: 19–28.

Cornuejols G, Sridharan R and Thizy JM (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *Eur J Opl Res* **50**: 280–297.

Goossens DR, Maas AJT, Spieksma FCR and van der Klundert JJ (2004). *Exact algorithms for procurement problems under a total quantity discount structure*. Technical Report OR 0452, Faculty of Economics and Applied Economics, K.U. Leuven.

Holmberg K (1994). Solving the staircase cost facility location problem with decomposition and piecewise linearization. *Eur J Opl Res* **75**: 41–61.

Holmberg K and Ling J (1997). A Lagrangean heuristic for the facility location problem with staircase costs. *Eur J Opl Res* **97**: 63–74.

Khumawala BM (1972). An efficient branch and bound algorithm for the warehouse location problem. *Mngt Sci* **B18**: 718–731.

Klose A and Görtz S (2007). A branch-and-price algorithm for the capacitated facility location problem. *Eur J Opl Res* **179**: 1109–1125.

Kochetov Yu and Ivanenko D (2003). Computationally difficult instances for the uncapacitated facility location problem. In: T Ibaraki *et al* (eds). *Proceedings of the Fifth Metaheuristics International Conference (MIC2003)*, Kyoto; Japan.

Körkel M (1989). On the exact solution of large-scale simple plant location problems. *Eur J Opl Res* **39**: 157–173.

Krarup J and Bilde O (1977). Plant location, set covering and economic lot size: An $O(mn)$—algorithm for structured problems. In: *Numerische Methoden bei Optimierungsaufgaben, Band 3 (Tagung, Oberwolfach, 1976)*. Birkhäuser: Basel, pp 155–180. *Internat Ser Numer Math*, Vol. 36.

Lobo MS, Fazel M and Boyd S (2007). Portfolio optimization with linear and fixed transaction costs. *Ann Opl Res* **152**: 376–394.

Martello S and Toth P (1990). *Knapsack Problems. Algorithms and Computer Implementations*. John Wiley & Sons Ltd: Chichester.

Munson CL and Rosenblatt MJ (1998). Theories and realities of quantity discounts: An exploratory study. *Prod Opl Mngt* **7**: 352–369.

Sadrian A and Yoon Y (1994). A procurement decision support system in business volume discount environments. *Opns Res* **42**: 14–23.

Sherali HD, Choi G and Ansari Z (2001). Limited memory space dilation and reduction algorithms. *Comput Opt Appl* **19**: 55–77.

Shor NZ (1970). Utilization of the operation of space dilation in the minimization of convex functions. *Cybern Syst Anal* **6**: 7–15.

Shor NZ (1998). Nondifferentiable optimization and polynomial problems. In: *Nonconvex Optimization and Its Applications, Vol. 24*. Kluwer Academic Publishers: Dordrecht.

Shor NZ and Zhurbenko NG (1971). A minimization method using the operation of extension of the space in the direction of the difference of two successive gradients. *Cybernetics* **7**: 450–459.

Shor NZ and Stetsyuk PI (2002). Lagrangian bounds in multiextremal polynomial and discrete optimization problems. *J Glob Optim* **23**: 1–41.

Todd MJ (1986). The symmetric rank-one quasi-Newton method is a space-dilation subgradient algorithm. *Opl Res Lett* **5**: 217–219.

Wu LY, Zhang XS and Zhang JL (2006). Capacitated facility location problem with general setup cost. *Comput Opl Res* **33**: 1226–1241.

Received August 2008;
accepted September 2009 after one revision