Available online at www.sciencedirect.com
SCIENCE DIRECT®

Discrete Applied Mathematics 133 (2004) 3–28

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Discrete facility location and routing of obnoxious activities

P. Cappanera[a,*], G. Gallo[a], F. Maffioli[b]

[a] Dipartimento di Informatica, Università di Pisa, Via F. Buonarroti 2, 56127 Pisa, Italy
[b] Dipartimento di Elettronica ed Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Received 11 October 2000; received in revised form 5 March 2002; accepted 15 December 2002

Abstract

The problem of simultaneously locating obnoxious facilities and routing obnoxious materials between a set of built-up areas and the facilities is addressed.

Obnoxious facilities are those facilities which cause exposure to people as well as to the environment i.e. dump sites, chemical industrial plants, electric power supplier networks, nuclear reactors and so on. A discrete combined location-routing model, which we refer to as *Obnoxious Facility Location and Routing* model (*OFLR*), is defined. *OFLR* is a *NP*-hard problem for which a Lagrangean heuristic approach is presented. The Lagrangean relaxation proposed allows to decompose *OFLR* into a *Location subproblem* and a *Routing subproblem*; such subproblems are then strengthened by adding suitable inequalities. Based on this Lagrangean relaxation two simple Lagrangean heuristics are provided. An effective *Branch and Bound* algorithm is then presented, which aims at reducing the gap between the above mentioned lower and upper bounds. Our Branch and Bound exploits the information gathered while going down in the enumeration tree in order to solve efficiently the subproblems related to other nodes. This is accomplished by using a *bundle method* to solve at each node the Lagrangean dual. Some variants of the proposed Branch and Bound method are defined in order to identify the best strategy for different classes of instances. A comparison of computational results relative to these variants is presented.

Keywords: Obnoxious location-routing; Lagrangean heuristic; Branch-and-Bound

* Corresponding author.
E-mail address: cappaner@di.unipi.it (P. Cappanera).

0166-218X/$ - see front matter
doi:10.1016/S0166-218X(03)00431-1

1. Introduction

The topic of *Obnoxious Facilities Location* [7,9,10,20] has been studied since the environmental impact has been considered as a real problem of the industrial societies. Before 1970s scientists have mainly focused their attention on classical location problems i.e. service facilities location such as hospitals, supermarkets, post-offices and warehouses.

Obnoxious facilities are those facilities which affect people as well as the environment i.e. dump sites, chemical industrial plants, electric power supplier networks and nuclear reactors. Hereafter by *affected sites* we mean all those sites which might be affected by the establishment of obnoxious facilities, such as for example built-up areas, schools, natural reserves or tourist zones.

As a consequence of the massive industrialization of the contemporary society, obnoxious facilities location has become a real risk to human being.

Moreover the location of an obnoxious or potentially dangerous facility usually determines either the origin or the destination of obnoxious materials shipments, and therefore interacts with the routing decisions: the facility location and transportation logistics decisions are strictly interrelated within the context of obnoxious materials management systems.

There are situations where the routing aspect prevails over the location one: such is the case for instance of electric power supplier network. Here the nuisance is mainly due to the electro-magnetic field the electricity induces while passing through the network. Nowadays the electro-magnetic pollution is felt as a real risk and people are very quick in raising opposition against the installation of such networks. In the last ten years for instance, the Italian electric society ENEL has attempted to locate at least four electric power supplier networks and failed in all the cases due to the public opposition.

Moreover recently increasing attention has been given to a particular kind of waste materials: dangerous or *hazardous materials*, also referred to as *hazmat* such as explosives, flammable liquids and solids, oxidizing substances, radioactive materials, corrosive substances, poisonous and infectious substances in general.

First of all we observe a lack of papers on this topic as the recent surveys by List et al. [17] and by Boffey and Karkazis [2] suggest; secondly almost all of the models presented in the literature have been developed from an analytical viewpoint [14,18,21] and deal with the minimization of risk.

Actually two aspects strictly connected with the location of obnoxious facilities can be outlined: the perceived risk due to the possibility of an accident to occur and the exposure that obnoxious facilities cause when settled nearby a built-up area. The risk is related to the probability of an accident but for instance, a chemical industrial plant based nearby a built-up area causes exposure because of its toxic emissions daily spilt over. In this regard we have focused on minimizing exposures rather than reducing risks.

Risk assessment is a very critical task and it is out of the aim of our work: the interested reader can find some references in [11]. As far as we know the earliest combined routing and location approach has been proposed by Shobrys [23] in spent nuclear fuel transportation setting. The model simultaneously locates the storage facilities and select paths for the spent fuel shipments so as to minimize the total cost and total risk of transportation.

However there is very little work on combined location and routing problems concerning the minimization of exposures: among these, the discrete models proposed by Zografos and Samara [24] and List and Mirchandani [16] are briefly described later; another combined model for planning *Urban Solid Waste Management System* in the Italian region of Lombardy is developed in Caruso et al. [8] where four different types of obnoxious facilities are considered and the multiobjective function consists of the following components: (i) *economic cost*, i.e. opening and management costs of new facilities as well as transportation costs are taken into account; (ii) *bad use of resources* measured by the amount of waste material which is disposed at the sanitary landfills (plants where waste from which no further recovery is possible is disposed); (iii) *environmental impact* such as pollution of air and water, soil impoverishment, negative impact on the landscape, public opposition and so on, measured for each possible type of plant and location.

This work is organized as follows: in Section 2 a discrete obnoxious location-routing model, referred to as *Obnoxious Facility Location and Routing* model (*OFLR*) is introduced; then it is compared with two other combined location-routing models proposed in the literature, respectively in [16,24]. *OFLR* is a general model which fits many real situations and Lagrangean heuristic approaches for it are proposed in Section 3; the test problems used are described in Section 4 while an analysis of computational results is shown in Section 5. Some conclusions are drawn in the final section.

2. The model

The *Obnoxious Facility Location and Routing* (*OFLR*) problem is formulated as a capacitated minimum cost network flow model and the following assumptions are done: (i) the model is single commodity, i.e. a single obnoxious material is considered; (ii) the affected sites are represented as point in the plane; (iii) sites which neither generate nor consume the obnoxious material might be affected by transportation activities; (iv) for each affected site, location and routing exposure thresholds are given.

Let now introduce the mathematical formulation of *OFLR*: given a directed graph $G=(V,A)$ where A is the set of arcs and the set of vertices V is the union of the following three sets

$R=\{1,2,\ldots,m\}$	the set of affected sites
$N=\{1,2,\ldots,n\}$	the set of candidate locations to establish the *new facilities*
$T=\{1,2,\ldots,t\}$	the set of transshipment nodes,

let us define

δ_i	the demand of vertex i ($\delta_i = 0\ \forall i \in T \cup N$). Each node $i \in R$ can be a source of obnoxious material ($\delta_i \leqslant 0$)
α_{ij}^k	the exposure caused by a unitary flow along the arc (i, j) to affected site k
τ_k	the threshold of affected site k relative to the exposure induced by the routing of obnoxious materials
a_{ij}	the exposure caused by the opening of a facility in location j to affected site i
t_i	the threshold of affected site i relative to the exposure induced by the establishment of obnoxious facilities
u_j	the capacity of a facility located in site j
c_j	the opening cost of a facility located in site j
γ_{ij}	the transportation cost of a unitary flow along the arc (i, j).

An artificial super destination node p is added to V, to which every location site $j \in N$ is connected via an artificial arc (j, p) of zero cost. The demand δ_p of node p is equal to $-\sum_{i \in R} \delta_i$ thus guaranteeing that the demand of each affected site $i \in R$ is disposed of entirely.

Let us denote by V' and A' the extended set of nodes and arcs respectively, i.e.
$V' = V \cup \{p\}$,
$A' = A \cup \{(j, p)\colon\ j \in N\}$.

OFLR problem can thus be formulated as

$$OFLR \quad \min \quad \sum_{j \in N} c_j y_j + \sum_{(i,j) \in A} \gamma_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{j:\ (j,i) \in A'} x_{ji} - \sum_{j:\ (i,j) \in A'} x_{ij} = \delta_i \quad \forall i \in V' \tag{1}$$

$$\sum_{(i,j) \in A} \alpha_{ij}^k x_{ij} \leqslant \tau_k \quad \forall k \in R \tag{2}$$

$$\sum_{j \in N} a_{ij} y_j \leqslant t_i \quad \forall i \in R \tag{3}$$

$$x_{jp} \leqslant u_j y_j \quad \forall j \in N \tag{4}$$

$$x_{ij} \geqslant 0 \quad \forall (i, j) \in A' \tag{5}$$

$$y_j \in \{0, 1\} \quad \forall j \in N. \tag{6}$$

There are two kind of variables in *OFLR*:

- the 0–1 *location variables* y_j's where

$$y_j = \begin{cases} 1 & \text{if a new facility is located in site } j, \\ 0 & \text{otherwise;} \end{cases}$$

- the continuous *routing variables* x_{ij}'s representing the quantity of flow along arc (i,j).

OFLR consists of selecting a set of locations among the candidate ones where opening the new facilities and routing the obnoxious flow from the affected sites to the opened facilities so as to minimize the opening cost of the facilities and the transportation cost and to satisfy a set of constraints we are going to explain.

Constraints (1) are the *flow conservation constraints* for each vertex of the network; they assure that the demand of each affected site $i \in R$ is disposed of entirely. Observe that because of the flow conservation constraint relative to facility j

$$x_{jp} = \sum_{i:\ (i,j)\in A} x_{ij} - \sum_{i:\ (j,i)\in A} x_{ji} \quad \forall j \in N$$

and x_{jp} represents the quantity of flow sent to site j to be disposed.

When dealing with transportation of obnoxious materials from a set of affected sites to the opened facilities, an attempt at constraining the exposure caused by the flow along the arcs of the network to the nearby affected sites should be made, besides minimizing the transportation costs as in a classical routing problem. Constraints (2), which we refer to as *routing exposure constraints*, state that for each affected site k the total exposure caused by the routing of obnoxious materials along the arcs of the network must not exceed the threshold τ_k; constraints (3), which we refer to as *location exposure constraints*, assert that for each affected site i the sum of exposures caused by all the opened facilities must not be over the fixed threshold t_i. In our combined location-routing problem there are thus two kinds of exposure: the exposure caused by the establishment of obnoxious facilities and the exposure caused by the routing of obnoxious flows along the network.

Constraints (4) say that if facility in location j is closed ($y_j = 0$) node j can be used as a transshipment node but you cannot route obnoxious flow to facility j in order to dispose it, i.e. $x_{jp} = 0$. On the other hand, if facility in location j is open, the total quantity of flow to be disposed at site j, namely x_{jp}, cannot exceed the capacity of the facility.

OFLR is a *NP*-hard problem: in fact, it is straightforward to show [12] that the *Maximum Independent Set* problem, a *NP*-complete problem asking for the existence in a graph of an independent set of k vertices, can be reduced to a simplified recognition form of *OFLR* concerning location aspect only.

In the remainder of this section *OFLR* is compared with the models by Zografos and Samara [24] and by List and Mirchandani [16]: in [24] a goal programming model is presented which simultaneously minimizes (i) routing risk, (ii) location risk and

(iii) routing cost according to user-controlled priorities. The main differences between *OFLR* and the model by Zografos and Samara are the following:

- in [24] each affected site suffers from the nearest open facility only, whilst in *OFLR*, each affected site is exposed to a set of open obnoxious facilities, according to the value of the a_{ij}'s
- in [24] the total risk given by location and routing activities is minimized, i.e. risks are summed up over all of the affected sites differently from *OFLR* where, for each affected site, given exposure thresholds must not be exceeded
- in [24] the number of facilities to open is fixed while in *OFLR* is variable
- in [24] there are capacity constraints on the arcs of the network versus uncapacitated arcs in *OFLR*.

In [16] a multiobjective model is proposed in which risk, cost and risk equity are considered jointly. A set of non-overlapping zones defines the region of concern on which a transportation network is superimposed. For each zone, the risk from the routing of obnoxious materials along nearby arcs and the establishment of nearby facilities is taken into account and the total risk is minimized, which is given by the sum of the zonal risks. Equity is regarded by minimizing the maximal zonal risk. The main issues that characterize the model proposed by List and Mirchandani are the following:

- it is a multicommodity flow model in which different types of obnoxious wastes and materials are considered; on the contrary *OFLR* is single commodity. Besides in [16] two aspects strictly concerned to location decisions are outlined: where to locate an obnoxious facility and which kind of treatment technology has to be employed, while in *OFLR* a single type of facility is assumed
- impacts from both routing and location are assumed to be additive while in *OFLR* they are considered separately
- the model in [16] is uncapacitated, while in *OFLR* there is a capacity constraint on the facilities
- in [16] a path formulation is given: disutilities functions are defined arc by arc as a risk measure from the fatality, property damage and so forth; then such functions are combined together to produce path-specific functions. On the other hand in *OFLR* an arc formulation is given.

Heuristic algorithmic approaches for *OFLR* are provided in the next section.

3. The algorithmic approach

3.1. The lower bound

By dualizing the capacity constraints (4) on the facilities, *OFLR* is decomposable in two subproblems: one involving the location variables y_j only, which we refer to as the *Location Problem*; and the other one in the routing variables x_{ij}, the *Routing Problem*.

For a given non-negative vector λ the Lagrangean relaxation accounts for the evaluation of the function $\varphi()$ in point λ, i.e.

$$\varphi(\lambda) = \min\left\{\sum_{j\in N} c_j y_j + \sum_{(i,j)\in A} \gamma_{ij}x_{ij} + \sum_{j\in N}\lambda_j(x_{jp} - u_j y_j)\colon (1),(2),(3),(5),(6)\right\}$$

$$= \min\left\{\sum_{j\in N}(c_j - \lambda_j u_j)y_j\colon (3),(6)\right\}$$

$$+ \min\left\{\sum_{(i,j)\in A}\gamma_{ij}x_{ij} + \sum_{j\in N}\lambda_j x_{jp}\colon (1),(2),(5)\right\}.$$

Since $\varphi()$ is a separable function evaluating it for a given vector λ amounts to solve a Location problem and a Routing one. Let us examine more closely the two subproblems.

Because of the non-negativity of data and because of the decision variables y_j, the location subproblem

$$\min\left\{\sum_{j\in N}(c_j - \lambda_j u_j)y_j\colon \sum_{j\in N} a_{ij}y_j \leqslant t_i \ \forall i\in R,\ \ y_j\in\{0,1\}\ \forall j\in N\right\}$$

can be preprocessed as follows:

- if the updated cost $c'_j = c_j - \lambda_j u_j$ of the variable y_j is greater or equal than 0, then $y_j = 0$.

Let N^- be the set of variables with updated negative cost; we can then state the following problem

$$MKP \quad \max\left\{\sum_{j\in N^-} -c'_j y_j\colon \sum_{j\in N^-} a_{ij}y_j \leqslant t_i \ \forall i\in R,\ \ y_j\in\{0,1\}\ \forall j\in N^-\right\}$$

which is equivalent to our Location problem provided we put a minus sign in front of its objective function. The Location problem results thus in a 0–1 *Multidimensional Knapsack Problem*.

On the other hand the routing subproblem is a linear programming problem. For a given vector of Lagrangean multipliers λ, the two subproblems are only weakly correlated: for instance on one side the Location subproblem lacks information about how many facilities have to be opened in order to satisfy the flow conservation constraints; and on the other side the Routing subproblem routes flow to facilities disregarding their capacity and how much they disturb the affected sites of the network. So it makes sense to introduce into *OFLR* constraints that are superfluous as far as *OFLR* itself is concerned but meaningful for the subproblems thus strengthening them.

The following constraints have been added to *OFLR*:

$$\sum_{j\in N} u_j y_j \geqslant \delta_p \tag{7}$$

$$x_{jp} \leqslant u_j \quad \forall j \in N \tag{8}$$

$$\sum_{j\in N} \frac{a_{ij}}{u_j} x_{jp} \leqslant t_i \quad \forall i \in R. \tag{9}$$

Constraint (7) asserts that as many facilities have to be opened as to satisfy the flow conservation constraint of the affected sites.

Constraints (8) are weaker than the facilities capacity constraints (4) since $y_j \leqslant 1$. Imposing them is equivalent to set an upper capacity equal to u_j on each arc (j, p).

Finally, constraints (9) are a relaxation of the location exposure constraints (3) since $x_{jp}/u_j \leqslant 1\ \forall j \in N$. As a matter of fact from the relaxed constraints (4) we have $y_j \geqslant x_{jp}/u_j$ and this expression for y_j has been substituted in location exposure constraints. The quantity x_{jp}/u_j (a value between 0 and 1) measures the utilization factor of a facility located in site j.

By adding the above inequalities to *OFLR*, the two subproblems are strengthened as follows. The Location subproblem results in

$$IP \quad \min\left\{\sum_{j\in N} c'_j y_j\colon \sum_{j\in N} a_{ij} y_j \leqslant t_i \ \forall i \in R,\ \sum_{j\in N} u_j y_j \geqslant \delta_p,\ \ y_j \in \{0,1\}\ \forall j \in N\right\}$$

which is a general 0–1 *Integer Programming* problem. Computational results show that *IP* is much more difficult to solve than *MKP* where all constraints are less than or equal constraints.

On the other hand the routing problem strengthened with inequalities (8) and (9) is still a linear programming problem.

To find the best lower bound among all the lower bounds given by the Lagrangean relaxation w.r.t. the facility capacity constraints, the corresponding Lagrangean dual, $\max_{\lambda\geqslant 0} \varphi(\lambda)$ has to be solved where $\varphi()$ is the polyhedral (concave, non-differentiable) function defined above.

3.2. The upper bounds

When evaluating function $\varphi()$ in point λ, there is no precedence order in solving the two subproblems: we can indifferently solve the Location problem first and then the Routing one or vice-versa. This straightforward observation suggests two simple heuristics based on information given by the Lagrangean relaxation and known in the literature as *Lagrangean heuristics* [1,22].

The former, which we refer to as *Location–Routing heuristic*, takes the solution to the Location problem that indicates which facilities are to be opened and which not and force it in the Routing problem. If Routing problem is feasible, also a feasible solution to *OFLR* has been obtained.

On the contrary the latter one, referred to as *Routing–Location heuristic*, takes the solution to the Routing problem, opens the facilities to which flow has been sent and verifies if this solution is feasible for the Location problem. When this happens, also a feasible solution to *OFLR* has been built up, otherwise our heuristic chooses a facility to close according to some criterion discussed below, and solve a new Routing problem until a feasible solution to *OFLR* has been obtained or a maximum number of iterations has been reached.

In the following the Lagrangean heuristics are described in detail.

We begin with the Location–Routing heuristic and let y^* be the optimal solution to Location problem, i.e. to *MKP* or to *IP* according to which subproblem has been solved.

The constraints $x_{jp} \leqslant u_j y_j^* \; \forall j \in N$ are added to the Routing problem which is solved to get a solution $\bar{x}$.

Observe that if $y_j^* = 0$, i.e. a facility is not established in site j, the constraint j above stated ensures that obnoxious flow cannot be routed to site j to be disposed. Site j can instead be used as a transshipment node. On the other side there is no way to assure that y_j^* is equal to 1 in the Routing problem, i.e. the flow along arc (j, p) may be zero even if $y_j^* = 1$. Sending flow to j is however encouraged by giving a zero cost to arc (j, p).

If an optimal solution $\bar{x}$ to the modified Routing problem exists, then $(\bar{x}, y^*)$ is also feasible to the original problem *OFLR*.

On the other side the Routing–Location heuristic constructs a solution $\bar{y}$ to the Location problem starting from the Routing one x^* (*Constructive Phase*) and if $\bar{y}$ is not feasible, iteratively performs a *Destructive Phase* until a feasible solution has been obtained or a maximum number of iterations has been reached.

- *Constructive Phase:* opens all those facilities j to which flow has been sent to be disposed, i.e. $\forall j \in N$

$$\bar{y}_j = \begin{cases} 1 & \text{if } x_{jp}^* > 0, \\ 0 & \text{otherwise.} \end{cases}$$

- *Feasibility Test:* tests if $\bar{y}$, obtained at previous step satisfies the location exposure constraints, i.e. if

$$\sum_{j \in N} a_{ij} \bar{y}_j \leqslant t_i \quad \forall i \in R.$$

  In such a case a feasible solution $(\bar{y}, x^*)$ to *OFLR* has been obtained; otherwise a (series of) Destructive Phase must be performed.

- *Destructive Phase*: if feasibility test fails a facility has to be closed, namely facility k; k is chosen as the less used facility among the ones to which flow has been sent, i.e.

$$k = \arg\min_{j \in N} \{x_{jp}^*/u_j \colon x_{jp}^* > 0\}.$$

  Obnoxious flow is then re-routed in order to get a new solution x^* and to come back to a Constructive Phase.

3.3. The Branch and Bound method

Preliminary computational results have shown that the gap between lower and upper bounds given by the Lagrangean heuristics presented above is not very tight: on some instances the gap is over 25%. On the same instances the gap between the optimal solution value (where known) and the bound given by the linear programming relaxation is over 40%.

In order to reduce quickly such gaps, an effective Branch and Bound (B&B) algorithm is designed, which is based on the lower and upper bounds described above. Branching is done on the 0–1 location variables according to the general rule followed when dealing with a 0–1 Mixed Integer Programming problem; without loss of generality we assume that the left branch corresponds to close a facility.

Among the issues that make a B&B efficient the followings seem to be particularly important: (i) encourage the flowing of information from a node to another node of the B&B tree: restarting from scratch is in fact not desirable in a situation like this. To this end when computing the lower bound of a subproblem, information should be gathered to be successively screened and passed to the next subproblem if still valid; (ii) get an optimal (fractional) solution y^* to the continuous relaxation of *OFLR* in order to select a good branching variable.

In our B&B method the lower bound is given by the Lagrangean relaxation proposed in Section 3.1: deciding which non-differentiable optimization method should be used to solve the Lagrangean dual is thus a critical task and requires a careful selection.

A *bundle* method in which by definition, information (subgradients) gathered during the steps are maintained to guide the search toward optimality seems to be more suitable than a classical subgradient method to build up a B&B with the properties above mentioned.

Many algorithms fall in the class of bundle methods: the first ones have been proposed independently by Lemaréchal and Wolfe; Lemaréchal [15] explored and stressed the potentialities of a bundle method and put the basis for future developments.

The bundle methods are iterative ascent algorithms: they compute a sequence of points $\{\lambda^i\}$ for which the concave non-differentiable objective function $\varphi()$ does not decrease. At iteration i the following steps are performed:

(1) finding an ascent direction d^i
(2) deciding whether a step along d^i should be taken or not.

The ascent direction is obtained by maximizing a local model of the objective function $\varphi()$. This model, differently from what happens in the classical algorithms, is built upon a set of disaggregated information (subgradients) corresponding to points generated at the previous steps of the algorithm; such information cannot be put together to compute an approximating function since $\varphi()$ is a non-differentiable function. This is the main characteristic of a bundle method.

Step 2 is a test on the goodness of the current local model: if it is quite good, or equivalently if along the direction computed at Step 1 a sufficient increase of the objective function is obtained, a step along d^i is performed and the next point λ^{i+1} becomes a better estimate of the maximizer.

Otherwise (direction d^i is not an ascent direction) the obtained information are gathered to enrich the model and generate at the next iteration a new research direction starting from λ^i.

This is the second main feature of a bundle algorithm whether the next point is given by a *line-search* along the current direction or it is computed by a *trust region* approach.

For a detailed description of the bundle method used in the B&B framework the reader is referred to [4,6]; here the main issues we exploit in the B&B method are pointed out:

(1) the way information gathered at each iteration are reused
(2) the availability of an optimal solution to (a problem stronger than) the continuous relaxation of *OFLR* at the termination of the algorithm.

It is easy to show that

$$g_j(\lambda) = x_{jp} - u_j y_j \quad \forall j \in N$$

is the jth component of a subgradient $g(\lambda)$ for $\varphi()$ in λ where x and y are the optimal solutions to the location and routing problems for a given vector λ.

In the following we denote by g^i the subgradient computed at iteration i and by (y^i, x^i) the corresponding solution.

According to the ε-subgradient theory, a subgradient $g(\lambda)$ for $\varphi()$ in a point λ is also an ε-subgradient for $\varphi()$ in a completely different point for a proper value of ε. This fact can be used to transfer the information (subgradients) from a node to another node of the enumeration tree as the following property shows:

Proposition 1. *Let*
j *be the index of the branching variable and*
B *be the set of subgradients*
computed at the current node. Then

$$B^0 = \{g^i \in B\colon\ y^i_j = 0 \text{ and } x^i_{jp} = 0\}$$

is a set of valid subgradients for the left subproblem, obtained from the current one by adding the constraint $y_j = 0$, *while*

$$B^1 = \{g^i \in B\colon\ y^i_j = 1\}$$

is a set of valid subgradients for the right subproblem ($y_j = 1$).

On the other side an optimal solution to a problem stronger than the linear relaxation of *OFLR* is obtained by using a bundle method. When a bundle method converges a set of convex multipliers θ_i is given such that $g(\theta) = \sum \theta_i g^i$ is the zero subgradient and $(y(\theta), x(\theta)) = (\sum \theta_i y^i, \sum \theta_i x^i)$ is an optimal solution to the following problem

$$\overline{OFLR} \quad \min\{cy + \gamma x\colon\ (x, y) \in \{(4) \cap \mathrm{conv}\{(1), (2), (3), (5), (6)\}\}.$$

which is stronger than the linear relaxation of *OFLR* (c and γ are the location and routing cost vectors respectively).

So at each node, when the bundle algorithm terminates the (fractional) solution $y(\theta)$ can be used to select a branching variable.

According to the dimension of the instance to be solved, computing a series of 0–1 *MKP* at each node of the enumeration tree may require a great computational burden: in such cases a weaker lower bound to *OFLR* is provided by computing an upper bound to *MKP*; there is thus a tradeoff between the quality of the bound and the computational effort to compute it.

In our approach an upper bound to *MKP* is given by the surrogate relaxation obtained by using the dual optimal variables to the linear relaxation of *MKP* as multipliers. It is easy to show that such an upper bound results in a 0–1 *Knapsack Problem*, for which efficient algorithms based on *Dynamic Programming* are known (see for instance the one by Pisinger [19]).

Computational results show that solving *IP* to optimality instead of *MKP* is a difficult task; on the other hand it is easy to show that the surrogate relaxation of *IP* is still a knapsack problem: constraint (7) is thus successfully used to strengthen the location problem.

When using one or the other of the surrogate relaxations above described to compute an upper bound to the location problem, a greater attention should be paid in conveying the subgradients from a node to another node of the enumeration tree.

At each node, at the first iteration of the algorithm which computes a lower bound to *OFLR*, a continuous location problem is solved and its dual optimal solution is then used to compute the surrogate constraint; then a series of 0–1 knapsack problems is computed until the Lagrangean dual is solved: such knapsack problems have all the same constraint and different objective functions depending on the Lagrangean multipliers.

The way the subgradients generated at a particular node of the enumeration tree are re-used at the other nodes is described by the following property.

Proposition 2. *Let*

j *be the index of the branching variable at current node and*
B *be the bundle of subgradients associated with the parent node.*
B *is split up in the following two subsets*

$$B^0 = \{g^i \in B\colon\ y^i_j = 0,\ x^i_{jp} = 0 \text{ and } a_0 y^i \leqslant b_0\}$$

where $a_0 y \leqslant b_0$ *denotes the surrogate constraint associated with the left son of the current node and*

$$B^1 = \{g^i \in B\colon\ y^i_j = 1 \text{ and } a_1 y^i \leqslant b_1\}$$

where $a_1 y \leqslant b_1$ *is the surrogate constraint related to the right son.*

B^0 is thus a set of feasible subgradients for the left son while B^1 is feasible for the right one.

So when using surrogate relaxation, at each node also the subgradients coming from the "grandfather" have to be examined: as a matter of fact, there may exist subgradients discarded by the parent node because they do not satisfy the aggregate constraint, which however may be still valid as far as the current node in concerned.

The Location–Routing Lagrangean heuristic described in previous section is used inside the B&B algorithm to get an upper bound to *OFLR*; in fact computational results (see Table 6) show that this heuristic performs better than the other Lagrangean heuristic proposed. Recall that an optimal solution to 0–1 *MKP* is required by the Lagrangean heuristic and usually such a solution is not available when computing an upper bound to *MKP* instead of solving *MKP*.

The tabu search heuristic described in [5] is thus used to generate good feasible solutions to *MKP* starting from which the Location–Routing heuristic attempts to get a feasible solution to the original problem. Besides being used as starting point to the Lagrangean heuristic, the best feasible solutions returned by the tabu search are also conveyed in subgradients to be passed to the bundle algorithm as the property below shows.

Proposition 3. *Let λ be the current vector of Lagrangean multipliers and let (y,x) be the corresponding primal solution. Let then $\{y^1,\dots,y^K\}$ be the set of approximate solutions given by the tabu search approach in λ. Then for each k, (y^k,x) is an approximate solution to $\varphi()$ in λ and g^k is an α^k-subgradient of $\varphi()$ in λ where*

$$g_j^k = x_{jp} - u_j y_j^k \quad \forall j \in N$$

and

$$\alpha^k = \varphi^k(\lambda) - \varphi(\lambda)$$

where $\varphi^k(\lambda)$ is the cost of the approximate solution (y^k,x).

Some straightforward rules are used which aims at enumerating implicitly a (hopefully large) number of nodes of the B&B tree.

Given a node of the enumeration tree, we denote by I the set of indices of variables which have not yet been fixed; let then $k \in I$ be the index of the branching variable and y be the partial solution corresponding to non-active variables. The left subtree of the current node (corresponding to $y_k = 0$) is pruned if the following conditions hold:

$$\sum_{j \in I \setminus \{k\}} u_j < \delta_p - \sum_{j \in N \setminus I} u_j y_j.$$

Such condition states that if y_k was fixed to 0 constraint $\sum_{j \in N} u_j y_j \geqslant \delta_p$ would not be satisfied even if the rest of the active variables were set to 1.

On the other hand the right subtree (corresponding to $y_k = 1$) is pruned when

$$\exists i \in R\colon\ a_{ik} > t_i - \sum_{j \in N \setminus I} a_{ij} y_j.$$

i.e. when at least a location exposure constraint i would be violated even if all of the active variables except variable k were fixed to 0.

The two rules above mentioned allow to prune a node by infeasibility but a node can also be pruned by optimality in a situation like this: suppose that there exists an affected site for which the location exposure constraint would be violated if anyone of the not yet opened facilities was established, or more formally stated

$$\exists i \in R\colon \min_{j \in I} \{a_{ij}\} > t_i - \sum_{j \in N \setminus I} a_{ij} y_j.$$

Then

$$y_j^* = \begin{cases} 0 & \text{if } j \in I, \\ y_j & \text{otherwise,} \end{cases}$$

is the optimal solution to the current problem which is thus closed by optimality.

As anticipated when dealing with the description of a bundle method, at each node of the B&B tree the optimal solution y^* to $\overline{OFLR}$ together with other information such as the optimal Lagrangean multipliers λ^* is helpful in selecting the branching variable. Two branching rules are proposed: the index of the branching variable j^* is selected as

$$j^* = \arg\max_j \{(\lambda_j^* u_j) \min\{y_j^*, 1 - y_j^*\} \ \forall j \in N\colon\ 0 < y_j^* < 1\} \tag{a}$$

or

$$j^* = \arg\max_j \{(|c_j - \lambda_j^* u_j|) \min\{y_j^*, 1 - y_j^*\} \ \forall j \in N\colon\ 0 < y_j^* < 1\}. \tag{b}$$

The rationale is that the term $\min\{y_j^*, 1 - y_j^*\}$ chooses the most fractional variable j^* among the integer variables that are fractional in the LP solution y^*: this is the classical rule usually used in a LP-based B&B method.

Furthermore in our case, the decision regarding branching variable j^* is also guided by the quantities $\lambda_j u_j \ \forall j$ or by the optimal Lagrangean cost of the variables, namely $c_j - \lambda_j^* u_j, \ \forall j$.

4. The test problems

There is a lack of works dealing with combined obnoxious location and routing problems and we were not able to find publicly available instances on which testing our exact as well as heuristic approaches. We have thus decided to design an $OFLR$-instances generator which according to a set of parameters computes exposure factors, thresholds and other data upon a planar graph.

Let i be the index of a node in V; hereafter we denote by x^i the corresponding point in the plane and by $d(x, y)$ the Euclidean distance between points x and y.

The demand δ_i of each affected site $i \in R$ is a non-negative integer randomly generated in the range $[\delta_{\min}, \delta_{\max}]$. We assume that there is a weight $w_i = \delta_i/\delta_{\max}$ associated with affected site i which stands for the importance of affected site i; the rationale is that the bigger the demand of affected site i is, the more i is important.

Given the fraction k_c of facilities to be opened,

$$\Delta u = \frac{\sum_{i\in R}\delta_i}{k_c * |N|}$$

represents the average capacity of a facility in order to satisfy the demand constraints; then the capacity u_j and the cost c_j of facility $j \in N$ are computed as a perturbation of Δu and u_j respectively. We assume also that there is a weight $w_j = u_j/u_{\max}$ associated with facility j which stands for the importance of facility j where $u_{\max}$ is defined as the maximum of u_j's.

The transportation cost γ_{ij} of arc $(i,j)\in A$ is a perturbation of the Euclidean distance between x^i and x^j.

It is reasonable to assume that location and routing exposures depend on the Euclidean distance and on the weight of the involved sites; consider the location exposure a_{ij} for instance: the more affected site i is far from facility j the less i is disturbed by j. Here we assume that exposures are an exponential function of the distance but any function decreasing with the distance may be used as well. As far as the location exposure is concerned we have

$$a_{ij} = w_i w_j \exp(-k_e d(x^i, x^j)^2) \quad \forall i \in R,\ j \in N.$$

On the contrary computing routing exposures is a little more complex: in this case the exposure caused by a flow along arc (i,j) to affected site k depends either on the Euclidean distance between (i,j) and k or on the length of the path Γ_{ij} between x^i and x^j in the region under study. We thus define:

$$\alpha_{ij}^k = w_k \int_{\Gamma_{ij}} \exp(-k_e d(x, x^k)^2)\,\mathrm{d}x \quad \forall (i,j)\in A,\ \forall k \in R.$$

Indeed Γ_{ij} is approximated in the following way: for each arc $(i,j)\in A$ a point P is randomly taken in a square box centered in the middle point of the line from x^i to x^j. The path Γ_{ij} is thus approximated by the two segments connecting P with x^i and x^j respectively. Location and routing exposures are then normalized in the ranges $[0, a_{\max}]$ and $[0, \alpha_{\max}]$ respectively.

Given the parameter k_n, location thresholds are computed as follows:

$$t_i = k_n * \sum_{j\in N} a_{ij} \quad \forall i \in R,$$

i.e. for each affected site i, t_i is equal to a fraction k_n of the sum of exposures caused by all the facilities to affected site i.

Likewise given the parameter k_a, routing thresholds are computed as follows:

$$\tau_k = k_a * \sum_{(i,j)\in A} \alpha_{ij}^k \delta_k \quad \forall k \in R,$$

i.e, for each affected site k, τ_k is equal to a fraction k_a of the sum of exposures caused by the routing to k; here we assume δ_k as estimate of the flow along arc (i,j).

Finally the objective function of *OFLR* is chosen as a convex combination of location and routing costs.

5. Computational results

In this section computational results relative to several C++ versions of the B&B algorithm described in Section 3.3 are presented; in all the variants of the B&B code the Routing problem is the linear programming problem obtained from *OFLR* decomposition and tightened with inequalities coming from the Location problem. The main features that characterize the different versions of the B&B code are the following:

- which and how location problem is solved
- which branching rule is used
- how many times and starting from which solution the Lagrangean heuristic is executed.

Let us explain such differences in detail. Decisions concerning the location problem are controlled via two parameters: `which` and `how`. Parameter `which` controls which location problem is solved: '`p`' stands for *MKP*; '`m`' for *IP*. According to the value of parameter `how`, the location problem can be solved either to optimality ('`o`') via the CPLEX MIP solver [13] or via a surrogate relaxation ('`r`') by mean of Pisinger's code [19]. In the remainder of this section combinations `o`–`p`, `r`–`p` and `r`–`m` are analyzed; computational results concerning the combination `o`–`m` are not given since preliminary tests have shown that *IP* becomes much more difficult to solve than *MKP*: there are many *IP*-instances CPLEX is not able to solve within reasonable computational time, although the original *MKP* is quite easy.

The way the Location–Routing heuristic is used characterizes the several versions of the B&B code as follows:

- `bb` version: at each iteration of the bundle method, given the current vector λ of Lagrangean multipliers, the Location–Routing heuristic is executed starting from the optimal solution y to the Location problem if such a solution is feasible and if it guarantees that constraint (7) is satisfied. Clearly the feasibility of y has to be controlled only if *MKP* is not solved to optimality
- `bb1` version: the heuristic is run only once at every node of the enumeration tree starting either from the primal solution given by the bundle if integer or from a feasible solution to *MKP* obtained via the tabu search heuristic
- `bb2` version: such a version attempts to improve the performance of `bb`. When the optimal solution to the Location problem is not feasible to *MKP* a trial to solve *MKP* approximately is made by invoking the tabu search.

The following measures are used to test the effectiveness of the B&B method:

$$\varepsilon_{\mathrm{LB}} = \frac{z^* - z_{\mathrm{LB}}}{z^*}, \quad \varepsilon_{\mathrm{FS}} = \frac{z^* - z_{\mathrm{FS}}}{z^*} \quad \text{and} \quad \varepsilon_{\mathrm{UB}} = \frac{z_{\mathrm{UB}} - z^*}{z^*},$$

Table 1
Location–Routing results

	o–p				r–m				r–p				
	$\varepsilon_{\mathrm{LB}}$	$\varepsilon_{\mathrm{UB}}$	FiCall	T_{TOT}	$\varepsilon_{\mathrm{LB}}$	$\varepsilon_{\mathrm{UB}}$	FiCall	T_{TOT}	$\varepsilon_{\mathrm{LB}}$	$\varepsilon_{\mathrm{UB}}$	FiCall	T_{TOT}	T_{CPLEX}
60–15–15	18.47	0.00	33	0.6	31.81	24.89	13	0.06	31.99		12	0.04	0.18
	20.80	0.00	26	0.53	29.70	9.52	13	0.1	31.09		13	0.04	0.17
	13.35	14.31	36	0.69	13.24	1.79	40	0.2	16.93		37	0.13	0.29
50–15–15	33.68	4.59	15	0.14	33.67	7.87	19	0.13	33.68	19.83	15	0.05	0.75
	21.85	1.57	12	0.12	21.85	29.99	15	0.07	21.85	12.10	11	0.04	0.45
	31.19	2.49	11	0.15	31.18	13.81	20	0.17	31.19	9.07	13	0.08	0.42
	32.32	1.39	28	0.22	31.81	0.69	28	0.18	32.31	0.69	33	0.11	0.7
	24.38	0.54	21	0.21	24.05	16.30	25	0.1	24.38	10.26	24	0.08	0.44
	29.80	5.00	19	0.19	29.17	16.01	16	0.12	29.80	10.78	21	0.06	0.37
	1.41	0.00	54	0.66	0.78	0.00	101	0.46	1.43	0.00	39	0.24	0.18
	5.58	8.52	64	0.62	0.39	0.00	51	0.22	6.08	5.41	27	0.14	0.61
	2.79	3.28	54	0.7	1.98	0.27	113	0.48	2.77	5.12	74	0.38	0.28
90–30–30	22.25	5.86	52	1.92	22.27	13.76	105	1.45	22.26	8.17	43	1.63	60.46
	17.91	6.70	47	1.68	11.03	8.38	105	1.88	17.91	4.53	45	1.59	47.05
	22.83	8.95	51	1.64	22.83	7.49	99	1.9	22.83	9.22	38	1.18	41.16
	18.14	4.88	46	1.58	18.16	8.13	66	1.09	18.14	5.11	53	1.76	7.84
	21.91	9.23	70	2.75	16.37	9.52	73	1	21.91	3.70	71	2.44	20.6

where z_{LB} is the lower bound value, z_{UB} the upper bound value, z_{FS} the lower bound value relative to the first feasible son of the root node and z^* is the optimal value (given by CPLEX). $\varepsilon_{\mathrm{FS}}$ measures the reduction of the gap between lower and upper bounds obtained just by fixing an integer variable.

From now onwards per cent relative errors are always reported and computational times are expressed in CPU seconds; the machine used to test the code is a Pentium II 400 MHz with 128 MB of main memory.

The instances generated in order to validate the efficiency of the approach are identified by a triple v–m–n where v is the total number of nodes of the underlying graph, m is the number of affected sites and n is the number of candidate locations to open the new facilities. For all the instances generated the parameters k_n, k_a and k_c are set to 0.5, 0.35 and 0.3 respectively.

One of the main aims of this work is to show that the approach proposed is able to take advantage of a set of information gathered while visiting the enumeration tree and can thus be used successfully to find out quickly quite good solutions. This is the reason why we start reporting on computational results relative to the quality of the solution obtained by using a B&B truncated at the root node. In Table 1 the performance of the Location–Routing heuristic is reported for the following instances:

- three 60–15–15 instances which differ only as far as the weight of the two subproblems in the objective function is concerned: in the first instance the routing cost prevails over the location one in the optimal solution (by one order of magnitude); in the second instance, location and routing costs have the same order of magnitude, while for the third one the location cost prevails over the routing one. The demand of each affected site is an integer uniformly drawn in the range [1500, 2000]; location sites are destination nodes and they cannot be used as transshipment nodes
- nine 50–15–15 instances: in the first 3 instances the routing cost prevails over the location one by two orders of magnitude; in the next 3 instances the two subproblems have the same weight, while in the latter ones the location cost dominates the routing one, again by two orders of magnitude instead of one as in 60–15–15 problems. The demand of each affected site is in [500, 2000]; location sites can be used as transshipment nodes
- five 90–30–30 instances in which location and routing costs have the same order of magnitude, the demand of each affected site is in [1500, 2000] and location sites can be used as transshipment nodes.

For each combination `how–which` (namely `o–p`, `r–p`, `r–m`), besides ε_{LB} and ε_{UB}, the total number of iterations performed by the bundle algorithm (FiCall) and the total execution time (T_{TOT}) are reported. In the last column the time T_{CPLEX} spent by CPLEX (version 6.0) to solve the problem is given. Observe that CPLEX is able to solve 60–15–15 and 50–15–15 problems in less than 1 s. On the contrary the running time T_{CPLEX} rapidly increases on 90–30–30 problems: on average the time required by CPLEX to solve 90–30–30's is over 85 times bigger than the time required to solve 50–15–15's (see also Fig. 1). On the other hand, on 90–30–30 instances, our truncated B&B gives solution for which ε_{UB} is not bigger than 10% in less than 3 seconds and the increase of the computational time while passing from 50–15–15's to 90–30–30's is limited by a factor smaller than 3. Such a behaviour comes into evidence even more as the dimension of instances increases (a proof of this fact will be given later).

Let us now analyze computational results relative to `bb` and `bb1` versions on 60–15–15 problems, given in a disaggregated form in Table 2. The first two columns identify the approach used: in column 2 the triple `how–which–BranchingRule` is reported. Columns 3 and 4 report on relative errors above mentioned. In the next two columns the total number of explored nodes (Nodes) and the node where the best solution is found (BestNode) are reported. The last columns give the total running time of the B&B (T_{TOT}), the time spent in the location problem (T_{LOC}), the time spent while solving the routing problem (T_{ROUT}), the time spent in the Lagrangean heuristic (T_{HEUR}) and finally the time spent in the tabu search (T_{TS}) when such heuristic is used to solve *MKP* approximately.

The rows of Table 2 are grouped by the approach used and inside each group data relative to the 3 problems described above are reported.

The two branching rules used are equivalent in terms of number of explored nodes when *MKP* is solved to optimality (combination `o–p`); on the contrary Rule `b` dom-

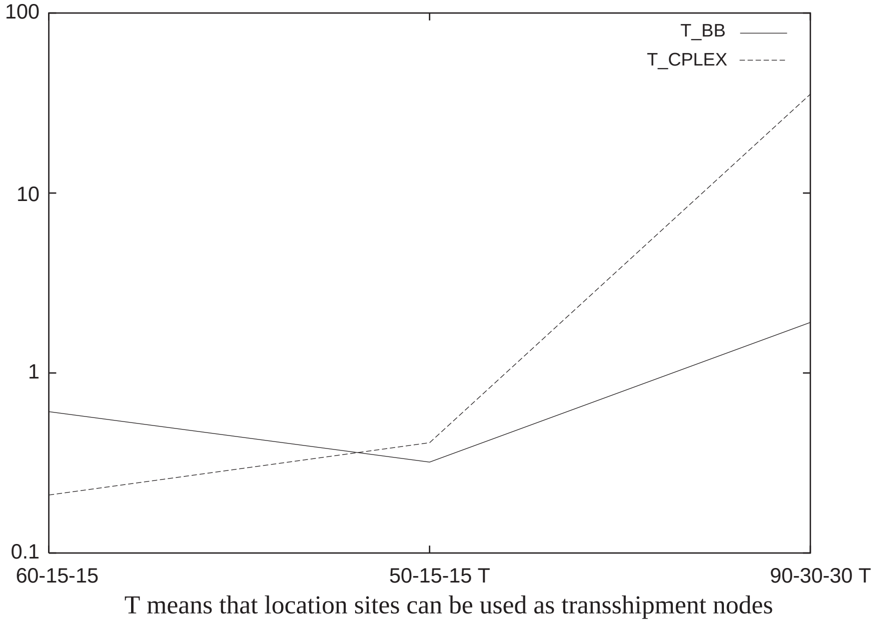

Fig. 1. Comparison between CPLEX and the B&B.

inates Rule `a` for the combination `r`–`p` and is dominated when combination `r`–`m` is used.

The number of nodes explored is lower in `bb` than in `bb1` since the optimal solution is found first when the former approach is used as the column BestNode clearly shows. For such instances invoking the Location–Routing heuristic for each value of the Lagrangean multipliers given by the bundle method does not involve a great computational effort; indeed the total time spent while running `bb` is lower, on average, than the one given by `bb1`. However the number of nodes explored is quite low in both the versions of the B&B. Observe that for all the combinations `how`–`which` (`o`–`p`, `r`–`p` and `r`–`m`) the smallest ε_{LB} is always given by the third problem: for such problem the bound given by the combination `r`–`m` is even better than the one given by `o`–`p`. The improvement obtained by using `r`–`m` with respect to `r`–`p` is quite small for the first instance and progressively increases for the other instances. Observe that as far as the first instance is concerned the deterioration of the lower bound given by `r`–`m` or `r`–`p` with respect to `o`–`p` is quite high. Anyway for the first two problems the gap between lower and upper bounds decreases meaningfully (reduced to a half and even more when `o`–`p` is used) just by fixing an integer variable, as column relative to ε_{FS} in Table 2 clearly shows. Computational results on these instances seem to show that the B&B is quite effective since the optimal solution is found quickly and the combination `r`–`m` seems to be a good alternative to `o`–`p` at least when the location cost prevail over the routing one.

Table 2
`bb` and `bb1` results on 60–15–15

	Alg	ε_{LB}	ε_{FS}	Nodes	BestNode	T_{TOT}	T_{LOC}	T_{ROUT}	T_{HEUR}	T_{TS}
bb	o–p–a	18.47	5.53	16	1	1.74	0.9	0.37	0.44	
		20.80	10.65	27	1	2.07	0.98	0.48	0.56	
		13.35	10.20	49	2	5.03	2.65	1.11	1.17	
	o–p–b	18.47	5.53	14	1	1.88	1.08	0.31	0.47	
		20.80	10.65	25	1	1.79	0.83	0.35	0.55	
		13.35	10.20	64	2	3.58	1.23	1.17	1.12	
	r–p–a	31.99	24.16	94	2	1.22	0.01	0.84	0.28	
		31.09	25.00	133	62	2.86	0.04	1.89	0.73	
		16.93	14.56	87	28	1.8	0.02	1.37	0.25	
	r–p–b	31.99	24.16	75	2	1.06	0.01	0.73	0.27	
		31.09	28.55	96	37	1.93	0.04	1.2	0.54	
		16.93	12.92	84	3	1.89	0.01	1.27	0.5	
	r–m–a	31.81	24.08	112	38	1.78	0	1.08	0.59	
		29.70	24.41	96	45	2.64	0.03	1.58	0.86	
		13.24	12.30	52	2	3.01	0.12	1.58	1.08	
	r–m–b	31.81	25.98	226	200	3.44	0.11	2.17	0.9	
		29.70	25.14	123	104	3.06	0.05	1.93	0.89	
		13.24	12.09	95	64	4.37	0.07	2.52	1.62	
bb1	r–p–a	31.99	24.16	209	140	2.68	0.05	1.94	0.34	0.07
		31.09	25.00	134	57	2.25	0.03	1.78	0.27	0.11
		16.93	14.56	125	78	2.58	0.03	2.12	0.21	0.11
	r–p–b	31.99	24.16	178	111	1.98	0.03	1.5	0.28	0.05
		31.09	28.55	112	24	1.8	0.01	1.44	0.24	0.08
		16.93	12.92	154	108	2.71	0.05	2.08	0.26	0.11
	r–m–a	31.81	24.08	165	93	2.15	0.08	1.42	0.34	0.1
		29.70	24.41	151	105	2.58	0.04	2.06	0.23	0.04
		13.24	12.30	108	84	3.01	0.11	2.45	0.09	0.05
	r–m–b	31.81	25.98	246	241	3.05	0.09	1.94	0.49	0.12
		29.70	25.14	285	268	4.74	0.15	3.5	0.42	0.17
		13.24	12.09	183	174	4.38	0.21	3.58	0.12	0

In Tables 3 and 4 computational results relative to the 50–15–15 problems are reported in aggregated form.

Again the rows of these tables are grouped by version of B&B used and inside a group each row reports average results obtained on the 3 instances in the order given above (i.e. the first row corresponds to the first 3 instances, the second one to the next 3 and finally the third row corresponds to the last 3 instances).

Table 3
bb results on 50–15–15

Alg	ε_{LB}	ε_{FS}	Nodes	BestNode	T_{TOT}	T_{LOC}	T_{ROUT}	T_{HEUR}
o–p–a	28.91	21.18	89	69	13.34	5.28	2.10	5.65
	28.83	23.25	112	66	14.30	5.26	2.61	6.04
	3.26	2.91	106	42	8.74	3.07	2.57	2.81
o–p–b	28.91	21.32	125	47	13.79	4.59	2.30	6.50
	28.83	24.92	162	73	16.52	4.90	3.12	7.98
	3.26	3.11	105	60	7.85	2.47	2.42	2.66
r–p–a	28.91	21.18	258	222	5.08	0.06	2.40	2.09
	28.83	23.24	228	162	5.15	0.05	2.60	2.10
	3.42	3.18	231	202	6.03	0.08	3.77	1.80
r–p–b	28.91	24.01	332	245	6.09	0.07	3.10	2.23
	28.83	26.10	295	136	7.27	0.08	3.53	2.97
	3.42	3.18	204	169	5.15	0.07	3.19	1.56
r–m–a	28.90	24.52	248	223	5.72	0.07	2.22	2.89
	28.34	25.56	180	96	5.80	0.05	2.50	2.82
	1.05	0.76	16	1	1.95	0.06	1.00	0.82
r–m–b	28.90	21.14	399	307	8.24	0.13	3.92	3.37
	28.34	22.23	338	219	10.14	0.13	4.44	4.83
	1.05	0.44	17	5	2.17	0.05	1.08	0.95

Table 4
Comparison between bb and bb2 on 50–15–15

	bb	bb2
	BestNode	BestNode
r–p–1	222	162
	162	102
	202	165
r–p–2	245	151
	136	48
	169	155
r–m–1	223	166
	96	91
	1	1
r–m–2	307	248
	219	187
	5	5

These results show that the bound obtained by our approach is very small when the location problem prevails over the routing one compared with the bound obtained on the other instances. In the former case the best performance is obtained with bb and the combination r–m, both in terms of total execution time and number of nodes explored. By introducing the total demand constraint (7) the bound given by our approach is tight and the optimal solution is obtained at the root node when using the branching rule a. Such results seem to show that on such instances our approach can be successfully used to compute good solutions to *OFLR*.

On the contrary the gap obtained on the other instances is higher and on some instances our approach does not take advantage neither from the introduction of the total demand constraint (version r–m) nor from the optimal solution of *MKP* (version o–p).

A comparison between bb and bb2 in terms of BestNode is pointed out in Table 4. As well as for the other classes of instances the following trend emerges: the enhanced version bb2 allows to compute an optimal solution to *OFLR* earlier than bb and these results are hence encouraging particularly with the view of using the B&B to compute quickly good solutions.

Computational results, reported here in a qualitative way (see [3] for a more in depth analysis) have also shown that the efficiency of the B&B can be improved by implementing simple accelerating strategies. Among these the following seem to be worth considering:

- using approximated versions of the several B&B variants in which a node is closed as soon as $(z_{\mathrm{UB}} - z_{\mathrm{LB}})/z_{\mathrm{UB}} \leqslant \varepsilon$ for a given $\varepsilon > 0$. On the 60–15–15 problems computational results have shown that the best solution given by the approximated B&B ($\varepsilon = 0.01$) is still optimal to *OFLR* and the number of nodes explored is almost everywhere (27 times on 30 trials) smaller than the one reported by the corresponding non-approximated versions: the minimum decrease of explored nodes is 0.93% while the maximum is 37.5% (the average decrease is equal to 13.78%)
- adopting the *close by optimality* pruning rule: on 60–15–15 instances the average number of cut off nodes is equal to 8.65% with the bb version and 12.58% with bb1 and the reduction in the number of pruned nodes can be remarkable (22.88% at most) especially as the number of nodes of the enumeration tree grows over 200
- considering an extension of bb1: at each node when the bundle algorithm converges, *MKP* is solved approximately by using the tabu search as in bb1. However the solution returned by the tabu search may fail to satisfy constraint (7); when this happens all the best feasible solutions given by the tabu search are used in turn as starting point to the Lagrangean heuristic until the first one is found out that satisfies constraint (7) if such a solution exists. Computational results on 50–15–15 instances have disclosed that this enhanced version of bb1, namely bb3, performs better than bb1 almost everywhere when the combination r–p is used. Even more promising results seem to be obtained when the approximated versions of these two variants, namely abb1 and abb3, are considered: usually abb3 outperforms abb1 and sometimes allows to find an optimal solution to *OFLR* when abb1 fails: this

Table 5
bb1 results on 120–50–50

Nodes	$100 \cdot \frac{z_{UB} - z_{LB}}{z_{UB}}$
1	27.48
10	23.09
50	19.27
100	19.27
500	18.64
1000	18.64
2000	18.04
5000	15.73
10 000	15.73
15 000	15.73
25 000	15.73
50 000	15.46

means that information lost due to the approximated B&B are gained again by the tabu search heuristic.

Moreover also several instances with $|R| = |N| = 50$ have been generated which CPLEX is not able to solve to optimality within reasonable computational times. These “big” instances have a total number $|V|$ of nodes equal to 120 and a number of arcs less than 800. On these instances a careful selection of CPLEX parameters and strategies is required in order to avoid CPLEX running often out of memory without producing good feasible solutions to *OFLR*. In order to avoid the computational burden due to many calls to the Location–Routing heuristic and to the CPLEX MIP solver, a truncated version of bb1 is then run in conjunction with the combination r–m–1 on those 120–50–50 instances which CPLEX was not able to solve. Average computational results are given in Table 5. A maximum number of B&B nodes (reported in column Nodes) is fixed and for each value of Nodes the percentage gap between the lower bound computed at root node and the best solution found is given: in fact the optimal solution is not known. Observe that a notable reduction of such gaps is obtained after 5000 nodes have been explored, while appreciable improvements are not reported by increasing the maximum number of nodes up to 50 000. Such gaps compared with the ones obtained on the other instances at the root node (i.e. $(z^* - z_{LB})/z^*$ is compared with $(z_{UB} - z_{LB})/z_{UB}$ since the optimal value z^* is not known) allow to believe that quite good solutions have been obtained with a maximum number of nodes fixed to 5000; moreover it is important to remark here that such hopefully good solutions are computed in less than 30 min.

Finally in Table 6 a comparison is made between the two Lagrangean heuristics proposed, namely Location–Routing and Routing–Location. Average results are given here for matter of evidence but it is quite interesting to report that Routing–Location fails to provide a feasible solution 9 times on 66 trials against the 3 unsuccessful trials of Location–Routing.

Table 6
Comparison between the two Lagrangean heuristics (aggregate results)

	o–p				r–m				r–p			
	LocRout		RoutLoc		LocRout		RoutLoc		LocRout		RoutLoc	
	ε_{UB}	T_{TOT}	ε_{UB}	T_{TOT}	ε_{UB}	T_{TOT}	ε_{UB}	T_{TOT}	ε_{UB}	T_{TOT}	ε_{UB}	T_{TOT}
60–15–15	4.77	0.61	25.27	1.76	12.06	0.12		0.84		0.07		0.81
50–15–15	2.88	0.14	16.82	0.75	17.22	0.12	**16.82**	0.99	13.67	0.06	16.82	0.72
	2.31	0.21	6.47	1.32	11.00	0.13	**4.44**	1.19	7.24	0.08	**4.44**	1.40
	3.93	0.66	12.70	2.68	0.09	0.39	12.76	4.07	3.51	0.25	13.45	1.96
90–30–30	7.13	1.91	13.88	4.99	9.46	1.46	13.88	8.38	6.15	1.72	14.05	4.49

The following conclusions can be drawn: Routing–Location is outperformed by Location–Routing both in terms of quality of the best solution found and in terms of time on all the instances when o–p strategy is used. On the contrary Routing–Location gives better solution than Location–Routing on some 50–15–15 instances as pointed out in Table 6 (bold entries). There is however a tradeoff between the quality of the solution returned by Routing–Location and the computational time it requires. As a matter of fact when the Routing–Location performs better than Location–Routing the time it requires is about 10 times the time spent by the other heuristic. For these reasons we have decided to use Routing–Location in our experiments.

6. Conclusions

It seems quite difficult to draw general conclusions about the efficiency of our approaches since, as the computational results show, a best strategy seems not to have emerged.

However the following trends have come into evidence:

- the number of nodes explored by all the versions of our B&B is quite low: information gathered by our approach while going down in the B&B tree have thus been used successfully
- when the location cost prevails over the routing one the lower bound obtained is quite good and in such cases strengthening the location problem is particularly important. Anyhow on all the instances the gap between lower and upper bounds decreases quickly as very few variables are fixed
- Cplex is able to solve problems with up to 15 location sites quickly (less than 1 second) and on such instances it makes no sense to compare our Lagrangean heuristic approach with a finely tuned package as Cplex. On the other hand *OFLR*-instances become much more difficult to solve as the number of location sites reaches 30 or 50 and on such instances the Lagrangean heuristic seems to be a good choice. Computational experiments seem to show that the best results are obtained by strengthening the location subproblem with information coming from *OFLR* and solving it via

a surrogate relaxation. Moreover branching rule a, which considers the obnoxious facilities near to saturation first, seems to dominate branching rule b.

These observations are encouraging particularly with the view of using the B&B to compute quickly good solutions rather than solving *OFLR* to optimality. In fact, by simply fixing a maximum number of explored nodes the truncated versions of our B&B are able to find out solutions of very good quality.

References

[1] C. Barnhart, Dual ascent methods for large-scale multicommodity flow problems, Naval Res. Logist. 40 (1993) 305–324.

[2] T.B. Boffey, J. Karkazis, Models and methods for location and routing decisions relating to hazardous materials, Stud. Locational Anal. 4 (1993) 149–166.

[3] P. Cappanera, Discrete facility location and routing of obnoxious activities, Ph.D. Thesis, University of Milano, 1999.

[4] P. Cappanera, A. Frangioni, Embedding a bundle method in a branch and bound framework: an application-oriented development, Technical Report TR-00-09, Dipartimento di Informatica, Università di Pisa, 2000.

[5] P. Cappanera, M. Trubian, A local-search-heuristic for the demand-constrained multidimensional knapsack problem, INFORMS JOC, to appear.

[6] P. Carraresi, A. Frangioni, M. Nonato, Applying bundle methods to the optimization of polyhedral functions: an applications-oriented development, Technical Report TR-96-17, Dipartimento di Informatica, Università di Pisa, 1996.

[7] E. Carrizosa, F. Plastria, Location of semi-obnoxious facilities, Stud. Locational Anal. 12 (1999) 1–27.

[8] C. Caruso, A. Colorni, M. Paruccini, The regional urban solid waste management system: a modelling approach, European J. Oper. Res. 70 (1993) 16–30.

[9] H.A. Eiselt, G. Laporte, Objectives in location problems, in: Z. Drezner (Ed.), Facility Location, a Survey of Applications and Methods, Springer Series in Operations Research, Springer, Berlin, 1995 (Chapter 8).

[10] E. Erkut, S. Neuman, Analytical models for locating undesirable facilities, European, J. Oper. Res. 40 (1989) 275–291.

[11] E. Erkut, V. Verter, Hazardous materials logistics, in: Z. Drezner (Ed.), Facility Location, a Survey of Applications and Methods, Springer Series in Operations Research, Springer, Berlin, 1995 (Chapter 20).

[12] G. Galbiati, On the approximation of the minimum disturbance p-facility location problem, Discrete Appl. Math. 118 (2002) 73–83.

[13] ILOG, Inc. Using the CPLEX Callable Library, Version 6.5, 1999.

[14] A. Jennings, R. Solar, Hazardous waste disposal network analysis, J. Environ. Eng. ASCE 110 (1984) 325–342.

[15] C. Lemaréchal, Bundle methods in nonsmooth optimization, in: C. Lemaréchal, R. Miffin (Eds.), Nonsmooth Optimization, IIASA Proceedings Series, Vol. 3, Pergamon Press, Oxford, 1977.

[16] G.F. List, P.B. Mirchandani, An integrated network planar multiobjective model for routing and siting for hazardous materials and wastes, Transp. Sci. 25 (1991) 146–156.

[17] G.F. List, P.B. Mirchandani, M.A. Turnquist, K.G. Zografos, Modeling and analysis for hazardous materials transportation: risk analysis, routing/scheduling and facility location, Transp. Sci. 25 (1991) 100–114.

[18] J. Mahoney, A location–allocation perspective on the hazardous waste management problem: an application to the electroplating industry in New England, Ph.D. Thesis, Department of Geography, Boston University, 1991.

[19] D. Pisinger, A minimal algorithm for the 0–1 knapsack problem, Oper. Res. 45 (1997) 758–767.

[20] F. Plastria, Optimal location of undesirable facilities: an overview, Belg. J. Oper. Res. Statist. Comput. Sci. 36 (1996) 109–127.

[21] S.I. Schwartz, R.A. McBride, R.L. Powell, Models for aiding hazardous waste facility siting decisions, J. Environ. Systems 18 (1989) 97–122.

[22] P.J. Shea, Lagrangian relaxation based methods for data association problems in tracking, Ph.D. Thesis, Colorado State University, 1998.

[23] D. Shobrys, A model for the selection of shipping routes and storage locations for a hazardous substance, Ph.D. Thesis, Johns Hopkins University, Baltimore, 1981.

[24] K.G. Zografos, S. Samara, Combined location–routing model for hazardous waste transportation and disposal, Transp. Res. Record 1245 (1989) 52–59.