Università di Pisa

Dipartimento di Informatica

Technical Report: TR-00-09

Embedding a Bundle Method in a Branch and Bound Framework: an Application-Oriented Development

P. Cappanera A. Frangioni

26 July 2000

ADDRESS: Corso Italia 40, 56125 Pisa, Italy. TEL: +39 050 887111 FAX: +39 050 887226

Embedding a Bundle Method in a Branch and Bound Framework: an Application-Oriented Development

P. Cappanera*   A. Frangioni*

26 July 2000

Abstract

An effective *Branch and Bound* algorithm based on Lagrangean heuristic is presented, which aims at reducing the gap between the lower and upper bounds. The Branch and Bound method proposed 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.

A discrete combined location-routing model, which we refer to as *Obnoxious Facility Location and Routing* model (*OFLR*), is used as a benchmark to validate the efficiency of the method proposed. Such model simultaneously locates obnoxious facilities and routes obnoxious materials between a set of built-up areas and the facilities. Obnoxious facilities are those facilities which cause nuisance to people as well as to the environment i.e. dump sites, chemical industrial plants, electric power supplier networks, nuclear reactors and so on.

OFLR is a particular case of the *Obnoxious Network Design* problem which occurs when also an obnoxious network has to be designed such as the case of electric power supplier networks. These are meaningful problems given the industrial development and the increasing importance of environmental issues in our society.

The use of the B&B to solve this latter problem is under development.

*Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy

1 Introduction

When dealing with big and difficult problems such as many 0-1 *Mixed Integer Programming* (*MIP*) problems it is important to look at their very structure and propose approaches which aim at exploiting their characteristics: in this work such a goal is accomplished by using a Lagrangean heuristic approach which gives us both a lower and an upper bound. We then present an effective Branch and Bound algorithm in order to reduce the gap between lower and upper bounds when they are not very tight.

Among the issues that make a Branch and Bound efficient the followings seem to be particularly important: (i) encourage the flowing of information from a node to another node of the Branch and Bound 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 to the continuous relaxation of the problem studied in order to select a good branching variable.

In the B&B method proposed, the lower bound asks for solving the Lagrangean dual: deciding which non-differentiable optimization method should be used 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 [17] 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 can not 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.

The structure of the work is the following: in Section 2 a B&B algorithm to solve a general 0-1 Mixed Integer Programming problem is presented which is based on Lagrangean heuristic. An *Obnoxious Network Design* (*OND*) model which we would tackle using the B&B proposed is presented in Section 3. Being *OND* a difficult problem to start with, a simpler problem has been defined which however preserves the same structure than *OND* and is worth considering because it is of great interest, namely the *Obnoxious Facility Location and Routing* (*OFLR*) problem. Section 4 deals with the algorithmic details of the B&B method when applied to *OFLR*. In Section 5 some computational results relative to *OFLR* are discussed. The paper concludes with some suggestions for future search.

2 A branch and bound method for MIP

Let consider a 0-1 *Mixed Integer Programming* problem

$$MIP \quad \min\{cx + fy : Ax \leq b, Dy \leq e, Gx + Hy \leq l, y \in \{0,1\}^n\}$$

where x and y are the continuous and decision variables respectively, A, D, G, H are the coefficients matrices and c, f, b, e, l are vectors of right dimension.

First the problem of obtaining lower bounds is addressed: a Lagrangean relaxation is considered in which constraints $Gx + Hy \leq l$ connecting variables x and y are dualized. The problem then decomposes in two subproblems, one involving only the decision variables and the other one involving only the continuous variables. For a given non-negative vector λ the Lagrangean relaxation w.r.t. constraints $Gx + Hy \leq l$ accounts for the evaluation of the separable (polyhedral, concave, non-differentiable) function $\varphi()$ in point λ, i.e.

$$\begin{aligned}\varphi(\lambda) &= \min\{cx + fy + \lambda(Gx + Hy - l) : Ax \leq b, Dy \leq e, y \in \{0,1\}^n\} \\ &= \min\{(f + \lambda H)y : Dy \leq e, y \in \{0,1\}^n\} + \\ &= \min\{(c + \lambda G)x : Ax \leq b\} - \lambda l\end{aligned}$$

and

$$g = Gx + Hy - l$$

is a subgradient for $\varphi()$ in λ where x and y are the optimal solutions to the subproblems for a given vector λ.
Among all the lower bounds given by the Lagrangean relaxation defined above the best lower bound is obtained by solving the Lagrangean dual $\max_{\lambda \geq 0} \varphi(\lambda)$, which by the primal interpretation of the Lagrangean dual [1],[2] is equivalent to the following problem:

$$\overline{MIP} \quad \min\{cx + fy : (x,y) \in \{Gx + Hy \leq l \cap \mathrm{conv}\{Ax \leq b, Dy \leq e, y \in \{0,1\}^n\}\}$$

which is stronger than the continuous relaxation of *MIP*.
The Lagrangean dual is then tackled using a bundle method: we stress the fact that the Lagrangean dual can be solved with a subgradient method as well, but a bundle method is certainly more suitable to our purposes since it allows to

1. exploit the information gathered at each iteration
2. get an optimal solution to $\overline{MIP}$.

The interested reader is referred to [8] for details.

Since an optimal solution to the Lagrangean relaxation is available a natural choice to get upper bounds seems to be given by Lagrangean heuristics: the main idea [3],[24] consists in taking the optimal solution to one of the two subproblems coming from the Lagrangean relaxation and then to complete it in order to obtain a feasible solution to the original problem. Hence two different heuristics are proposed: the former gets the optimal solution y^* to the 0-1 subproblem and forces it in the continuous problem by imposing constraints $Gx \leq l - Hy^*$. If the continuous subproblem is feasible, a feasible solution to the original problem has been obtained. On the other hand the latter heuristic gets the optimal solution to the continuous subproblem and forces it in the 0-1 problem.

A Branch and Bound algorithm is then designed which is based on the lower

and upper bounds described above.
Branching is done on the 0-1 decision variables according to the general rule followed when dealing with a 0-1 *MIP* problem; without loss of generality we assume that the left branch corresponds to set a variable to zero.
Let now explain the way the information gathered while going down in the enumeration tree are exploited in order to solve efficiently the subproblems related to other nodes.
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:

Property 2.1. *Let*

j	*be the index of the branching variable and*
B	*be the set of subgradients*

computed at the current node. Then

$$B^0 = \{g^i \in B : y^i_j = 0, Gx^i + Hy^i \leq l\}$$

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

$$B^1 = \{g^i \in B : y^i_j = 1, Gx^i + Hy^i \leq l\}$$

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

The optimal solution y^* to $\overline{MIP}$ computed at each node of the B&B tree can then be used together with other information such as the optimal Lagrangean multipliers λ^* in order to select the branching variable. Two branching rules are proposed: the index of the branching variable j^* is selected as

$$j^* = \arg\max_j \{|\lambda H^j| \min\{y^*_j, 1 - y^*_j\} \ \forall j \in \{1, \ldots, n\} : 0 < y^*_j < 1\} \tag{a}$$

or

$$j^* = \arg\max_j \{|f_j + \lambda H^j| \min\{y^*_j, 1 - y^*_j\} \ \forall j \in \{1, \ldots, n\} : 0 < y^*_j < 1\} \tag{b}$$

where H^j denotes the j-th column of matrix H.
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 the decision regarding branching variable j^* is also guided by the non-negative quantities $|\lambda H^j|$'s or by the (absolute value of the) optimal Lagrangean cost of the variables j's, namely $|f_j + \lambda H^j|$.

Solving a (series of) 0-1 subproblem at each node of the enumeration tree can require a great computational burden: in a situation like this a weaker lower bound to the original *MIP* problem may be preferred in which a relaxation of the subproblem involving only decision variables is solved instead of the 0-1 problem itself. When this happens solving the 0-1 subproblem heuristically is particularly important since such approximate solutions have a twofold function

- they can be used as starting points for the Lagrangean heuristic
- they are easily conveyed in information (subgradients) to be passed to the bundle method as the property below shows.

Property 2.2. *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 a set of approximate solutions to the problem involving only decision variables in point 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*

$$g^k = Gx + Hy^k - l$$

and

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

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

3 The Obnoxious Network Design problem

In some real situations an obnoxious network has to be designed such as the case of electric power supplier networks. This topic raises the public opposition due to the electro-magnetic pollution and to the environmental impact such networks produce.
For instance, in the last ten years Italian electric society *ENEL* has attempted to install at least four such networks and failed.
Moreover in the United States there are many electric societies which have to deal with the location of power plants and electric power supplier networks in very

big areas. This is the case for instance, of the Associated Electric Cooperative, Inc. (*AECI*), headquartered in Springfield, Missouri. *AECI* provides power to six regional generation and transmission cooperatives, who in turn supply power to thirty-nine Missouri, three southeast Iowa and nine northwest Oklahoma distribution cooperatives. These cooperatives serve over 680.000 homes and businesses representing 1.8 million individual consumers. The interested reader can find more details at their web page `http://www.aeci.org/aeci/info/company.html`. Usually in such situations power plants are already operative but new distribution plants and the electric power supplier network should be located.
Here the following problem is considered: a set of client sites such as cities or other built-up areas asks for the delivery of electric power; a number of distribution plants has to be opened so as to satisfy the demand constraint at the client sites and an electric supplier network has to be designed which links the clients with the distribution plants. In a situation like this two kinds of nuisance are identified: the nuisance given by the establishment of obnoxious facilities (the distribution plants) and the nuisance due to the electro-magnetic field the electricity induces while passing through the network. The client sites thus ask for a service and at the same time they suffer from the establishment of obnoxious facilities and links. This is the reason why they are referred to as *affected sites* and electricity is to be considered an obnoxious material.
Let now go into details concerning the formulation of such problems. 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

$R = \{1, 2, \dots, m\}$ the set of affected sites

$N = \{1, 2, \dots, n\}$ the set of candidate locations to establish the *new facilities*

$T = \{1, 2, \dots, t\}$ 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 $i \in R$ is a sink of obnoxious material ($\delta_i > 0$)

α_{ij}^k the nuisance given by the establishment of obnoxious link (i, j) to affected site k

τ_k the threshold of affected site k relative to the nuisance given by the routing of obnoxious materials

a_{ij} the nuisance given by the opening of a facility in location j to affected site i

t_i the threshold of affected site i relative to the nuisance given 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)

π_{ij} the opening cost of link (i,j).

An artificial super source node p is added to V, which is connected to every location site $j \in N$ via an artificial arc (p,j) 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 satisfied.
Let denote by V' and A' the extended set of nodes and arcs respectively, i.e.

$$V' = V \cup \{p\}$$
$$A' = A \cup \{(p,j) : j \in N\}.$$

The *Obnoxious Network Design* problem (OND), can be formulated as

$$OND \qquad \min \sum_{j\in N} c_j y_j + \sum_{(i,j)\in A} \gamma_{ij} x_{ij} + \sum_{(i,j)\in A} \pi_{ij} w_{ij}$$

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

$$\sum_{(i,j)\in A} \alpha_{ij}^k w_{ij} \le \tau_k \qquad \forall k \in R \tag{2}$$

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

$$x_{jp} \le u_j y_j \qquad \forall j \in N \tag{4}$$

$$x_{ij} \le M w_{ij} \qquad \forall (i,j) \in A \tag{5}$$

$$x_{ij} \ge 0 \qquad \forall (i,j) \in A' \tag{6}$$

$$y_j \in \{0,1\} \qquad \forall j \in N \tag{7}$$

$$w_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{8}$$

There are three kind of variables in *OND*:

- 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 0-1 *network design variables* w_{ij}'s with the following meaning:

$$w_{ij} = \begin{cases} 1 & \text{if the link } (i,j) \text{ is settled} \\ 0 & \text{otherwise;} \end{cases}$$

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

OND consists of selecting a set of locations among the candidate ones where opening the new facilities and a set of links along which obnoxious flow is routed, so as to minimize the opening cost of the facilities and the links besides 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 satisfied. Observe that because of the flow conservation constraint relative to facility j

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

and x_{pj} represents the quantity of flow outgoing from site j to be sent to the affected sites for consumption.
When dealing with transportation of obnoxious materials from the opened facilities to a set of affected sites, an attempt at minimizing the nuisance given by the flow along the arcs of the network to the nearby sites should be made, besides minimizing the transportation costs as in a classical routing problem.
Constraints (2), which we refer to as *routing nuisance constraints*, state that for each affected site k the total nuisance given by the establishment of obnoxious links must not exceed the threshold τ_k.
Constraints (3), which we refer to as *location nuisance constraints*, assert that for each affected site i the sum of nuisances given by all the opened facilities must not be over the fixed threshold t_i.
As anticipated before, in *OND* two kinds of nuisance are considered: the nuisance given by the establishment of obnoxious facilities and the nuisance given by the establishment of obnoxious links.
Constraints (4) say that if facility in location j is closed ($y_j = 0$) node j can

be used as a transshipment node but it is not a source of obnoxious flow, i.e. $x_{pj} = 0$. On the other hand, if facility in location j is opened, the total quantity of obnoxious material produced at site j, namely x_{pj}, can not exceed the capacity of the facility.
Constraints (5) assert that if link (i, j) is not established, i.e. $w_{ij} = 0$, flow can not be routed along it, i.e. $x_{ij} = 0$. On the contrary if link (i, j) is established, i.e. $w_{ij} = 1$, arc (i, j) is uncapacitated since M is a very big number.

4 A particular case: Obnoxious Facility Location and Routing

In this section a particular case of *OND* is discussed which we refer to as *Obnoxious Facility Location and Routing* (*OFLR*) problem.
Such a problem is used as a benchmark to validate the efficiency of the B&B proposed: as a matter of fact it is simpler than *OND* and yet it preserves the same structure.
However *OFLR* is also worth considering because it is of great interest: *OFLR* is a combined location and routing problem involving obnoxious activities and products. Such problems correspond to situations where polluting facilities must be established and some obnoxious material must be transported. These are meaningful problems given the industrial development and the increasing importance of environmental issues in our society. Research on such problems is relatively recent and, as pointed out in [5],[9],[11],[12],[19],[22] adequate models and solution methods are still to be provided. Moreover almost all of the models presented in the literature have been developed from an analytical viewpoint ([16],[20],[23]) and deal with the minimization of risk [18],[25],[26],[27],[28]. 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 nuisance that obnoxious facilities cause when settled nearby a built-up area. In this regard we have focused on minimizing nuisances rather than reducing risks.
There are very few works on combined location and routing problems concerning the minimization of nuisances: two discrete models are briefly described which have been presented in the literature.
The first work by Caruso *et al.* [10] develops a combined location and routing model for planning *Urban Solid Waste Management System* in the Italian region of Lombardy; an *Add and Drop* heuristic approach is presented. Four different

types of obnoxious facilities are considered. Their model is characterized by a multiobjective function, with 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) *waste of resources* measured by the amount of refuse which is disposed in 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.
The other work by Belotti *et al.* [4] deals with the discrete *repulsive p-median* problem which consists of finding p facility locations which will maximize the sum of distances between each affected site and the nearest obnoxious facility.
Let $G = (V, E)$ be a complete and Euclidean graph and let (R, N) be a partition of V where R denotes the set of affected sites and N denotes the set of potential obnoxious facility location sites; the problem can thus be formulated as

$$\text{Max}\left\{\sum_{i\in R}\ \min_{j\in N:\ y_j=1}\left\{d_{ij}y_j : \sum_{j\in N} y_j = p,\ y\in\{0,1\}^{|N|}\right\}\right\}$$

where d_{ij} is the Euclidean distance between the affected site i and the facility located in site j, and y_j is the decision variable relative to facility j. The authors [4] show that the repulsive p-median problem is $\mathcal{NP}$-hard, and propose several families of cutting planes and a local search heuristic to solve the problem. An exact branch and cut approach is developed and tested.

OFLR is formulated as follows

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

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

$$\sum_{(i,j)\in A} \alpha^k_{ij} x_{ij} \le \tau_k \qquad \forall k \in R \qquad (2)$$

$$\sum_{j\in N} a_{ij} y_j \le t_i \qquad \forall i \in R \qquad (3)$$

$$x_{jp} \le u_j y_j \qquad \forall j \in N \qquad (4)$$

$$x_{ij} \ge 0 \qquad \forall (i,j) \in A' \qquad (6)$$

$$y_j \in \{0,1\} \qquad \forall j \in N \qquad (7)$$

Here the affected sites are assumed to be sources of obnoxious material; node p is thus an artificial super destination node.

OFLR is a general model which fits many real situations: let us briefly focus on the following problem which is but one example of combined location-routing problem. We are given a network where three sets of nodes are identified: the set of built-up areas or affected sites R, the set of transshipment nodes T and the set of candidate locations to establish the new obnoxious facilities N. Waste material is produced at the built-up areas and their supply must be disposed of entirely and routed to the obnoxious facilities such as dump sites, incinerators and recycling centers, for disposal. A certain number of new facilities must be opened in order (i) to satisfy the demand constraint at the affected sites, (ii) to minimize the opening cost of the facilities, and (iii) to minimize a function of the nuisance due to the new facilities. These three objectives concern the location aspect but in a situation such as the one mentioned above, it is very important to consider the routing aspect as well. The location of an obnoxious or potentially dangerous facility determines either the origin or the destination of obnoxious materials shipments, and therefore interacts with the routing decisions: routing decisions must be taken which are not independent from the location ones. Moreover the transportation of waste material from the affected sites to the disposal sites along the links of the network disturbs the cities. Also this kind of nuisances (mainly smell in this case) should be taken into account and an attempt in minimizing their effect should be made. Thus obnoxious flows must be routed on the network and obnoxious facilities must be established on some of its nodes.

OFLR and *OND* differ only as far as two set of constraints are concerned. Observe that in *OND* the routing nuisance constraints (2) depend on the w_{ij}'s variables rather than on the flow variables x_{ij}'s such as the case of *OFLR*. It is in fact reasonable to assume that in *OND* the nuisance depends on the establishment of the obnoxious link, while in *OFLR* nuisance are proportional to the quantity of obnoxious flow passing through it. Constraints involving variables w_{ij}'s are not present in *OFLR* since a network must not be designed.
OFLR is a $\mathcal{NP}$-hard problem: in fact, it is straightforward to show [14] that the *Maximum Independent Set* problem, a $\mathcal{NP}$-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 following three sections lower bound, upper bounds and a B&B method for *OFLR* are discussed which reflect the general issues discussed in Section 2.

4.1 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.

$$\begin{aligned}\varphi(\lambda) &= \min\{\sum_{j\in N} c_j y_j + \sum_{(i,j)\in A} \gamma_{ij}x_{ij} + \sum_{j\in N}\lambda_j(x_{jp} - u_j y_j) : (1),(2),(3),(6),(7)\}\\ &= \min\{\sum_{j\in N}(c_j - \lambda_j u_j)y_j : (3),(7)\} + \min\{\sum_{(i,j)\in A}\gamma_{ij}x_{ij} + \sum_{j\in N}\lambda_j x_{jp} : (1),(2),(6)\}.\end{aligned}$$

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

$$\min\{\sum_{j\in N}(c_j - \lambda_j u_j)y_j : \sum_{j\in N} a_{ij}y_j \le t_i \ \forall i \in R,\ y_j \in \{0,1\}\ \forall j \in N\}.$$

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

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

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.

Location and routing subproblems are then strenghtened by adding suitable inequalities; observe that 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 without taking into account 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*:

$$\sum_{j \in N} u_j y_j \geq \delta_p \tag{9}$$

$$x_{jp} \leq u_j \qquad \forall j \in N \tag{10}$$

$$\sum_{j \in N} \frac{a_{ij}}{u_j} x_{jp} \leq t_i \qquad \forall i \in R. \tag{11}$$

Constraint (9) asserts that as many facilities have to be opened as to satisfy the flow conservation constraint of the affected sites.
Constraints (10) are weaker than the facilities capacity constraints (4) since $y_j \leq 1$. Imposing them is equivalent to set an upper capacity equal to u_j on each arc (j, p).
Finally, constraints (11) are a relaxation of the location nuisance constraints (3) since $x_{jp}/u_j \leq 1 \ \forall j \in N$. As a matter of fact from the relaxed constraints (4) we have $y_j \geq x_{jp}/u_j$ and this expression for y_j has been substituted in location nuisance 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

$$IP \quad \min\{\sum_{j \in N} c'_j y_j : \sum_{j \in N} a_{ij} y_j \leq t_i \ \forall i \in R, \ \sum_{j \in N} u_j y_j \geq \delta_p, \ y_j \in \{0,1\} \ \forall j \in N\}$$

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 enforced with inequalities (10) and (11) is still a linear programming problem.

4.2 The upper bound

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 Lagrangean heuristics: the former, which we refer to as *Location-Routing heuristic*, first takes the solution y^* to the Location problem that indicates which facilities are to be opened; constraints $x_{jp} \leq u_j y_j^*$ $\forall j \in N$ are then added to the Routing problem which is solved to get a solution $\overline{x}$.
Observe that if $y_j^* = 0$, i.e. a facility is not established in site j, the constraint j above stated assures that obnoxious flow can not 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 $\overline{x}$ to the modified Routing problem there exists, then $(\overline{x}, y^*)$ is also feasible to the original problem *OFLR*.
On the other side the latter heuristic, referred to as *Routing-Location heuristic*, constructs a solution $\overline{y}$ to the Location problem starting from the Routing one x^* (*Constructive Phase*) and verifies if $\overline{y}$ is feasible. If $\overline{y}$ is feasible, also a feasible solution to *OFLR* has been built up, otherwise a facility is iteratively closed and a new Routing problem is solved until a feasible solution to *OFLR* has been obtained or a maximum number of iterations has been reached (*Destructive Phase*).
More formally stated

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

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

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

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

  In such a case a feasible solution $(\overline{y}, x^*)$ 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.

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

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

4.3 The branch and bound

It is easy to show that

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

is the j-th 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.
The set B of subgradients available at the current node of the enumeration tree is transfered to the son nodes according to Property 2.1. In particular

$$B^0 = \{g^i \in B : y_j^i = 0 \text{ and } x_{jp}^i = 0\}$$

is a set of valid subgradients for the left subproblem while

$$B^1 = \{g^i \in B : y_j^i = 1\}$$

is a set of valid subgradients for the right subproblem ($y_j = 1$), where j is the branching variable computed at the current node.

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 big computational effort: 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 burden 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 [21]).

Computational results have shown 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 (9) is thus successfully used to enforce the surrogate relaxation of 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.

Property 4.1. *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

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

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

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

where $a_1 y \leq b_1$ 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 5) 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 Location-Routing 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 [7] 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 Property 2.2 suggests.

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 (the *active set*) ; 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 holds

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

Such condition states that if y_k was fixed to 0 constraint $\sum_{j \in N} u_j y_j \geq \delta_p$ 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

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

i.e. when at least a location nuisance 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 nuisance constraint would be violated if anyone of the not yet opened facilities was established, or more formally stated

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

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.

According to the general considerations stated in Section 2 the following branching rules are proposed: the index of the branching variable j^* is selected as

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

or

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

where λ^* is the optimal Lagrangean multipliers vector and y^* is the optimal solution to the following problem

$$\overline{OFLR} \quad \min\{cy + \gamma x : (x,y) \in \{(4) \cap \mathrm{conv}\{(1),(2),(3),(6),(7)\}\}\}.$$

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

5 Computational results

In this section preliminary computational results concerning the use of the B&B method proposed to heuristically solve *OFLR* are shown. These seem to indicate that the method provided is well able to exploit the information gathered along the enumeration tree to compute quickly quite good solutions, but that there is still room for improvement.

There is a lack of works dealing with combined obnoxious location and routing problems and consequently, as far as we know, there are no instances, publically available, on which testing our approach. An *OFLR*-instances generator has thus been designed [6] which according to a set of parameters computes nuisances, thresholds and other data upon a planar graph.

The demand δ_i of each affected site $i \in R$ is a non-negative integer randomly generated in the range $[\delta_{min}, \delta_{max}]$. Associated with affected site i there is a weight $w_i = \delta_i/\delta_{max}$ 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 the capacity u_j of facility j is computed as a perturbation of the average capacity a facility must have in order to satisfy the demand constraints, namely $\sum_{i \in R} \delta_i/(k_c * |N|)$. The cost c_j of facility j is a perturbation of u_j. As well as for the affected sites, 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 sites i and j.

It is reasonable to assume that location and routing nuisances depend on the Euclidean distance and on the weight of the involved sites; consider the location nuisance a_{ij} for instance: the more affected site i is far from facility j the less i

is disturbed by j. On the contrary evaluating routing nuisances is a little more complex: in this case the nuisance given by 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 between sites i and j. Here we assume that nuisances are exponential functions of the distance opportunely defined but any other reasonable function of the distance may be used as well: the interested reader in referred to [6] for more details concerning the way the Euclidean distance becomes a disutility measure. Location and routing nuisances 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 nuisances given to affected site i by all the facilities. This procedure has been suggested by Freville and Plateau [13] for the *MKP*.
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 nuisances given 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.
The instances generated 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.
The following measures are used to test the effectiveness of the B&B method:

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

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). ε_{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.

	o-p				r-m				r-p				
	ε_{LB}	ε_{UB}	FiCall	T_{TOT}	ε_{LB}	ε_{UB}	FiCall	T_{TOT}	ε_{LB}	ε_{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

Table 1: Location-Routing results

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 $[1500, 2000]$
- 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 $[500, 2000]$
- five 90-30-30 instances in which location and routing costs have the same order of magnitude and the demand of each affected site is in $[1500, 2000]$.

Several C++ versions of the B&B code have been designed and tested according to which and how location problem is solved, to which branching rule is used and

depending on how many times and starting from which solution the Lagrangean heuristic is executed.
The location problem is either *MKP* or *IP* (denoted by 'p' and 'm' respectively) and it can be solved either to optimality ('o') via the CPLEX MIP solver [15] or via a surrogate relaxation ('r') by mean of Pisinger's code [21]. This gives rise to 4 different combinations, three of which are analyzed, i.e. combinations `o-p`, `r-p` and `r-m`; 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.
Table 1 reports on results relative to the quality of the solution obtained by using a B&B truncated at the root node. For each of the three combinations described above, 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. All the times are expressed in CPU seconds; the machine used to test the code is a Pentium II 400 MHertz with 128 MB of main memory.
Observe that CPLEX is able to solve 60-15-15 and 50-15-15 problems in less than 1 second. 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. 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).

In Table 2 computational results relative to several versions of `bb` on 60-15-15 problems are reported: when `bb` is used, at each iteration of the bundle method 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 (9) is satisfied. Clearly the feasibility of y has to be controlled only if *MKP* is not solved to optimality.
The `bb` version used is identified via the triple `how-which-BranchingRule` reported in the first column of Table 2. 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}) and finally the time spent while solving the rout-

Alg	ε_{LB}	ε_{FS}	Nodes	BestNode	T_{TOT}	T_{LOC}	T_{ROUT}	T_{HEUR}
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

Table 2: `bb` results on 60-15-15

ing problem (T_{ROUT}).

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` dominates Rule `a` for the combination `r-p` and is dominated when combination `r-m` is used.

The number of nodes explored is quite low in all the versions of the B&B on these instances and the optimal solution is found quickly as the column BestNode clearly shows.

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` (such improvement is not very meaningful on this instance but computational results have shown that it gets bigger as the dimension of the problem increases). 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.
A comparison between `bb` and `bb2` in terms of BestNode is pointed out in Table 3. `bb2` 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 [7]. As well as for other classes of instances the following trend emerges: `bb2` allows to compute an optimal solution to *OFLR* earlier than `bb` and the truncated version of `bb2` can thus used to compute a good solution efficiently.

	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

Table 3: comparison between `bb` and `bb2` on 50-15-15

The behaviour of our B&B on instances bigger than the ones presented up to now is also investigated: 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 soon runs out of memory without producing good feasible solutions to *OFLR*.
A truncated version of `bb1` is run in conjunction with the combination `r-m-1` on those 120-50-50 instances which Cplex was not able to solve. When `bb1` is used the Location-Routing heuristic is run only once at every node of the enumeration tree starting either from the primal solution given by the bundle if

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
10000	15.73
15000	15.73
25000	15.73
50000	15.46

Table 4: `bb1` results on 120-50-50

integer or from a feasible solution to *MKP* obtained via the tabu search heuristic. Computational results not reported here have revealed that using strategy `bb1` with `r-m-1` is a good choice on big instances. Average computational results are given in Table 4. 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 50000. 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 minutes.

Finally in Table 5 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.
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

	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

Table 5: comparison between the two Lagrangean heuristics (aggregate results)

Routing-Location gives better solution than Location-Routing on some 50-15-15 instances as pointed out in Table 5 (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.
Computational results above presented are hence 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 good quality.

6 Future research lines

In this final section future research lines are identified: on one hand we would try to enforce the lower bound used; on the other hand we would further validate the method proposed applying it to other *MIP* problems such as for instance *OND*. *OFLR* was in fact used as a benchmark but the computational results obtained up to now allow to hope that embedding a bundle method in a B&B gives good results.
In the remainder of this section a relaxation stronger than the one proposed in Section 4.1 is presented which aims at reducing the gap between lower and upper bounds discussed above: enforcing such bound is particularly important when the routing problem prevails over the location one as the computational results have shown. This is accomplished by mean of *Lagrangean decomposition*; it is easy to show that the Lagrangean relaxation with respect to the facilities capacities constraints can be interpreted in terms of Lagrangean decomposition thus implying that few changes to our B&B are required in order to solve the

stronger relaxation. Let us see how this is possible.
For matter of evidence, let us rename the location variables y^l_j, where l stands just for location; let us then introduce the variables y^r_j (r stands for routing): each variable y^r_j is a copy of y^l_j, i.e. the following relation holds

$$y^l_j = y^r_j \quad \forall j \in N. \tag{12}$$

By introducing constraint (12), *OFLR* can be rewritten as follows:

$$
\begin{aligned}
\mathit{OFLR} \qquad & \min \sum_{j\in N} c_j y^l_j + \sum_{(i,j)\in A} \gamma_{ij} x_{ij} \\
& \text{s.t.} \sum_{j:(j,i)\in A} x_{ji} - \sum_{j:(i,j)\in A'} x_{ij} = \delta_i && \forall i \in V' && (1)\\
& \sum_{(i,j)\in A} \alpha^k_{ij} x_{ij} \le \tau_k && \forall k \in R && (2)\\
& \sum_{j\in N} a_{ij} y^l_j \le t_i && \forall i \in R && (3)\\
& x_{jp} \le u_j y^r_j && \forall j \in N && (4)\\
& y^l_j = y^r_j && \forall j \in N && (12)\\
& x_{ij} \ge 0 && \forall (i,j) \in A' && (6)\\
& y^l_j \in \{0,1\} && \forall j \in N && (7)\\
& y^r_j \in \{0,1\} && \forall j \in N && (13)
\end{aligned}
$$

The Lagrangean decomposition consists of dualizing constraints (12); for a given vector λ of Lagrangean multipliers not constrained in sign it accounts for the evaluation of the function $\varphi()$ in point λ, i.e.

$$
\begin{aligned}
\varphi(\lambda) \quad = \min \quad & \{\sum_{j\in N} c_j y^l_j + \sum_{(i,j)\in A} \gamma_{ij} x_{ij} + \sum_{j\in N} \lambda_j (y^r_j - y^l_j) \\
& \text{s.t.} (1), (2), (3), (4), (6), (7), (13)\}.
\end{aligned}
$$

The function $\varphi()$ is separable as follows:

$$
\begin{aligned}
\varphi(\lambda) = \quad & \min \quad \{\sum_{j\in N} (c_j - \lambda_j) y^l_j \text{ s.t. } (3), (7)\} + \\
& \min \quad \{ \sum_{(i,j)\in A} \gamma_{ij} x_{ij} + \sum_{j\in N} \lambda_j y^r_j \text{ s.t.} (1), (2), (4), (6), (13)\}
\end{aligned}
$$

and it is equivalent to solve a *0-1 Multidimensional Knapsack* problem and an *Uncapacitated Network Design* problem.
Observe that if we further relax the integrality constraints on the variables y^r_j we

obtain just the Lagrangean relaxation described in Section 4.1. In fact, in such a case y_j^r is equal to x_{jp}/u_j, the utilization factor of facility j. The Lagrangean relaxation with respect to the facilities capacity constraints can thus be viewed as a particular case of the Lagrangean decomposition where the integrality constraints on the routing variables y_j's have been relaxed.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows*, chapter 16. Prentice Hall, 1993.

[2] M. Guignard ans S. Kim. Lagrangean Decomposition: a Model Yielding Stronger Lagrangean Bounds. *Mathematical Programming*, 39:215–228, 1987.

[3] C. Barnhart. Dual Ascent Methods for Large-Scale Multicommodity Flow Problems. *Naval Research Logistics*, 40:305–324, 1993.

[4] P. Belotti, M. Labbé, F. Maffioli, and M. Ndiaye. The Discrete Obnoxious p-Median Problem: a Branch and Cut Approach. Working paper, 2000.

[5] T.B. Boffey and J. Karkazis. Models and Methods for Location and Routing Decisions Relating to Hazardous Materials. *Studies in Locational Analysis*, 4:149–166, 1993.

[6] P. Cappanera. *Discrete Facility Location and Routing of Obnoxious Activities.* PhD thesis, University of Milano, 1999.

[7] P. Cappanera and M. Trubian. A Local Search Heuristic for Hard Instances of the Demand Constrained Multiknapsack Problem. Working paper, 2000.

[8] P. Carraresi, A. Frangioni, and 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.

[9] E. Carrizosa and F. Plastria. Location of Semi-Obnoxious Facilities. *Studies in Locational Analysis*, 12:1–27, 1999.

[10] C. Caruso, A. Colorni, and M. Paruccini. The Regional Urban Solid Waste Management System: a Modelling Approach. *European Journal of Operational Research*, 70:16–30, 1993.

[11] H.A. Eiselt and G. Laporte. Objectives in Location Problems. In Z. Drezner, editor, *Facility Location, a Survey of Applications and Methods*, Springer Series in Operations Research, chapter 8. Springer, 1995.

[12] E. Erkut and S. Neuman. Analytical Models for Locating Undesirable Facilities. *European Journal of Operational Research*, 40:275–291, 1989.

[13] A. Freville and G. Plateau. An Efficient Preprocessing Procedure for the Multidimensional 0-1 Knapsack Problem. *Discrete Applied Mathematics*, 49:189–212, 1994.

[14] G. Galbiati. On the Approximation of the Minimum Disturbance p-Facility Location Problem. Working paper, 1999 - to appear on *Discrete Applied Mathematics*.

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

[16] A. Jennings and R. Sholar. Hazardous Waste Disposal Network Analysis. *Journal of the Environmental Engineering ASCE*, 110:325–342, 1984.

[17] C. Lemaréchal. Bundle Methods in Nonsmooth Optimization. In C. Lemaréchal and R. Mifflin, editors, *Nonsmooth Optimization*, volume 3 of *IIASA Proceedings Series*. Pergamon Press, 1977.

[18] G.F. List and P.B. Mirchandani. An Integrated Network/Planar Multiobjective Model for Routing and Siting for Hazardous Materials and Wastes. *Transportation Science*, 25:146–156, 1991.

[19] G.F. List, P.B. Mirchandani, M.A. Turnquist, and K.G. Zografos. Modeling and Analysis for Hazardous Materials Transportation: Risk Analysis, Routing/Scheduling and Facility Location. *Transportation Science*, 25:100–114, 1991.

[20] J. Mahoney. *A Location-Allocation Perspective on the Hazardous Waste Management Problem: an Application to the Electroplating Industry in New England*. PhD thesis, Department of Geography, Boston University, 1991.

[21] D. Pisinger. A Minimal Algorithm for the 0-1 knapsack Problem. *Operations Research*, 45:758–767, 1997.

[22] F. Plastria. Optimal Location of Undesirable Facilities: an Overview. *Belgian Journal of Operations Research, Statistics and Computer Science*, 36:109–127, 1996.

[23] S.I. Schwartz, R.A. McBride, and R.L. Powell. Models for Aiding Hazardous Waste Facility Siting Decisions. *Journal of Environmental Systems*, 18:97–122, 1989.

[24] P.J. Shea. *Lagrangian Relaxation Based Methods for Data Association Problems in Tracking*. PhD thesis, Colorado State University, 1998.

[25] D. Shobrys. *A Model for the Selection of Shipping Routes and Storage Locations for a Hazardous Substance*. PhD thesis, Johns Hopkins University, Baltimore, 1981.

[26] K.G. Zografos. A Combined Location-Routing Model for Hazardous Waste Transportation and Disposal. In *TIMS/ORSA Joint National Meeting*, New Orleans, May 1987.

[27] K.G. Zografos and S. Samara. A Combined Location-Routing Model for Hazardous Waste Transportation and Disposal. *Transportation Research Record*, 1245:52–59, 1989.

[28] K.G. Zografos and S.S. Samara. A Multiobjective Location-Routing Model for Improving the Safety of Hazardous Waste Transportation and Disposal. Manchester England, August 1988. VIII International Multicriteria Decision Making Conference.