Reformulation of capacitated facility location problems: How redundant information can help

Karen Aardal*
Department of Computer Science
Utrecht University
P.O. Box 80089
3508 TB Utrecht, The Netherlands
aardal@cs.ruu.nl

In memory of Åsa

Abstract

Most facility location problems are computationally hard to solve. The standard technique for solving these problems is branch-and-bound. To keep the size of the branch-and-bound tree as small as possible it is important to obtain a good lower bound on the optimal solution by deriving strong linear relaxations. One way of strengthening the linear relaxation is by adding inequalities that define facets of the convex hull of feasible solutions. Here we describe some simple, but computationally very useful classes of inequalities that were originally developed for relaxations of the facility location problems. Algorithms for generating violated inequalities belonging to the described classes have been implemented as system features in various branch-and-bound software packages, so as long as the software can recognize the relaxations for which the inequalities are developed, the inequalities will be generated "automatically". Here we explicitly add the variables and constraints that are necessary to describe the relaxations which means that we actually add information that is redundant both with respect to to the integer formulation and the linear relaxation. We present computational results indicating that the reformulations are useful as the generated inequalities close a substantial part of the duality gap. Due to the smaller branch-and-bound tree, the time needed by the branch-and-bound algorithm to solve the problem to optimality is reduced by up to ninety percent.

*Research partially supported by by ESPRIT Long Term Research Project No. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology*), and by NSF grant CCR-9307391 through David B. Shmoys, Cornell University.

1 Introduction.

Facility location problems are central in combinatorial optimization since they can be associated with a large variety of applications, and since they form relaxations of many involved distribution type problems. Most facility location problems are NP-hard, and therefore have to be solved by enumerative methods. In order to keep the enumeration tree as small as possible it is crucial to find good lower bounds on the optimal solution. Here we will discuss simple ways of reformulating capacitated facility location problems such that the linear relaxations associated with the various models yield good lower bounds on the optimal value. Structural properties of the location problems treated here have been studied by e.g. Leung and Magnanti (1989), Cornuéjols, Sridharan and Thizy (1991), Aardal (1992), and Aardal, Pochet and Wolsey (1995).

It is well-known that adding inequalities that define high-dimensional faces of the convex hull of feasible solutions to the initial formulation strengthens the lower bound obtained from the linear relaxation. This so-called *cutting plane technique* has made it possible to solve large-scale instances of several hard combinatorial optimization problems, see Aardal and Van Hoesel (1995,1996).

When developing valid inequalities for a certain problem type one typically starts by investigating the facial structure of various relaxations of the problem. The advantage is that the inequalities that are valid for a relaxation RP of a polyhedron P is also valid for P, and in several cases we have that inequalities that define high-dimensional faces of the convex hull of RP ($\text{conv}(RP)$) also define high-dimensional faces of $\text{conv}(P)$. Moreover, it may be possible to adapt the inequalities developed for RP to better reflect the structure of P. An example in this spirit is the family of *two-matching inequalities*, see Edmonds (1965), that were developed for the two-matching polytope. The constraints defining the two-matching polytope form a subset of the constraints defining the traveling salesman polytope, so the two-matching constraints are also valid for the traveling salesman polytope. The two-matching constraints were generalized by Chvátal (1975) to the class of *comb inequalities* that is valid for the traveling salesman polytope.

Here we shall discuss various classes of inequalities that have proved computationally efficient for capacitated facility location problems. These inequalities define facets of relaxations of the location problems, precisely as in the case of the two-matching inequalities with respect to the two-matching polytope, but here the relaxations are slightly less obvious, as they are not formed by simply taking a subset of the location constraints. An aspect that make these classes particularly useful is that their associated *separation problem*, i.e., the problems of identifying violated inequalities belonging to the classes, can be solved approximately by efficient heuristics. Moreover, these separation algorithms have been implemented in standard software packages such as MPSARX (Van Roy and Wolsey (1987)), and MINTO (Savelsbergh et al. (1994)). To make it possible for such software to "recognize" the relaxations in order to generate the inequalities, we have to state the relaxations explicitly, which in our case means that we have to add redundant information to the models. As we shall demonstrate, the increase in the size of the problem is fairly limited, whereas the improvement of the lower bound is quite large. Some of the inequalities treated here have also served as a starting point for developing new classes of inequalities designed to reflect the structure of the location problems more closely. These more specialized classes of inequalities are not treated here, but can be found in the articles of Leung and Magnanti, and of Aardal, Pochet and Wolsey. The computational usefulness of the inequalities developed for the relaxations discussed here supports the study of the facial structure of polytopes that occur as relaxations of several more involved polytopes.

We will consider two different capacitated facility location problems; the single-level capacitated facility location problem, and the two-level capacitated facility location problem. For both problems we will derive simple relaxations, and give examples of strong valid inequalities that are valid for the relaxations, and therefore also for the location problems. We finish by presenting our computational experience. The results demonstrate that the inequalities are very useful in reducing the duality gap and the size of the branch-and-bound tree.

2 The single-level capacitated facility location problem.

The problem is defined as follows. We are given a set $N = \{1, \ldots, n\}$ of clients. Client $k \in N$ has demand d_k of a certain type of goods. The goods are delivered from facilities, and we are given a set $M = \{1, \ldots, m\}$ of possible sites where facilities can be located. A facility located at site j has capacity m_j. The fixed cost of setting up a facility at site j is $f_j > 0$, and the cost of transporting one unit from facility j to client k is $c_{jk} \geq 0$.

To model the problem we use variables y_j, which take value one if a facility is opened at site j and value zero otherwise, and variables v_{jk} that denote the flow from facility j to client k. The mathematical programming formulation is as follows.

$$\min\{\sum_{j\in M}\sum_{k\in N} c_{jk}v_{jk} + \sum_{j\in M} f_j y_j : (v,y) \in X^{\mathrm{CFL}}\}$$

where

$$\begin{aligned} X^{\mathrm{CFL}} = \{(v,y) \in \mathbb{R}_+^{m\times n} \times \mathbb{Z}_+^m : \sum_{j\in M} v_{jk} = {}& d_k, \quad k \in N, & (1)\\ \sum_{k\in N} v_{jk} \leq {}& m_j y_j, \quad j \in M, & (2)\\ 0 \leq v_{jk} \leq {}& d_k y_j, \quad j \in M,\ k \in N, & (3)\\ y_j \leq {}& 1, \quad j \in M\}. & (4) \end{aligned}$$

Throughout this section we will assume that $\sum_{j\in M} m_j - m_r \geq \sum_{k\in N} d_k$, for all $r \in M$, which ensures that X^{CFL} is full-dimensional.

The location problem can be illustrated by a bipartite graph $G = ((U,V),A)$, where the vertices in U and V represent the facility sites and clients respectively. An arc (j,k) is introduced between the vertices $j \in U$ and $k \in V$ if $c_{jk} < \infty$, see Figure 1.

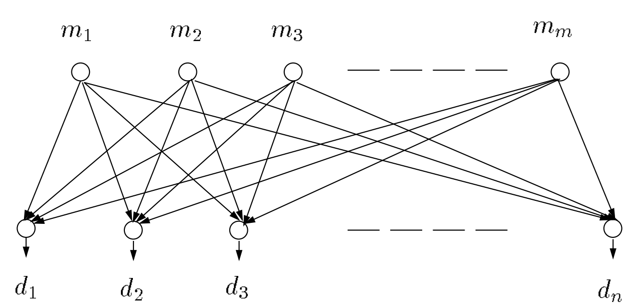


Figure 1: Representing the facility location problem by a bipartite graph.

Now, assume that we aggregate the clients to one "super client", which implies that we also aggregate all flow from a certain facility. Let

$$v_j - \sum_{k\in N} v_{jk} = 0, \tag{5}$$

and let $d(S) = \sum_{k\in S} d_k$. By using the aggregate flow v_j in the capacity constraint (2) we obtain

$$v_j \leq m_j y_j, \tag{6}$$

and by adding all demand constraints together we obtain

$$\sum_{j\in M} v_j = d(N). \tag{7}$$

Using (6) in (7) gives

$$\sum_{j \in M} m_j y_j \geq d(N). \tag{8}$$

We define the *surrogate knapsack polytope* X^{K} as:

$$X^{\mathrm{K}} = \{y \in \mathbb{Z}_+^m : \sum_{j \in M} m_j y_j \geq d(N),\ y_j \leq 1,\ j \in M\}.$$

Proposition 1 (Aardal, Pochet, Wolsey (1995).) *The polytope X^{K} is a relaxation of X^{CFL}, i.e., $X^{\mathrm{K}} \supseteq X^{\mathrm{CFL}}$.*

Consider a subset $J \subseteq M$ such that $\sum_{j \in J} m_j > \sum_{j \in M} m_j - d(N)$. We call such a set J a *cover* with respect to M and N, and if $\sum_{j \in S} m_j \leq \sum_{j \in M} m_j - d(N)$ for all $S \subset J$, then we say that J is a *minimal cover*. Since the capacity of the set $M \setminus J$ is too small to satisfy $d(N)$ we can conclude that the *cover inequality*

$$\sum_{j \in J} y_j \geq 1 \tag{9}$$

is valid for X^{K}, and since X^{K} is a relaxation of X^{CFL} we also know that (9) is valid for X^{CFL}. Wolsey (1975) gave sufficient conditions for the cover inequalities to be facet defining for $\mathrm{conv}(X^{\mathrm{K}})$. For the facility location polytope the following holds:

Theorem 2 (Aardal, Pochet, Wolsey (1995).) *Let $m_{\min} = \min_{j \in J} m_j$. If $J \subset M$ is a minimal cover with respect to M and N, and if $\sum_{j \in M \setminus J} m_j + m_{\min} > d(N)$, then the cover inequality (9) defines a facet of* $\mathrm{conv}(X^{\mathrm{CFL}}) \cap \{y \in \{0,1\}^m : y_j = 1 \text{ for } j \in M \setminus J\}$.

The most general form of a cover inequality is obtained by choosing a subset $M' \subseteq M$ and by initially setting $y_j = 0$ for all $j \in M \setminus M'$. We can now obtain a facet as in Theorem 2 for a minimal cover $J' \subset M'$ with respect to M' and N. By applying lifting to the variables $y_j,\ j \in M \setminus M'$ that are currently set to zero, and $y_j,\ j \in M' \setminus J'$ that are currently set to one, we obtain an inequality of the following form:

$$\sum_{j \in M \setminus M'} \alpha_j y_j + \sum_{j \in M' \setminus J'} \beta_j y_j + \sum_{j \in J'} y_j \geq 1 + \sum_{j \in M' \setminus J'} \beta_j. \tag{10}$$

The coefficients $\alpha_j,\ j \in M \setminus M'$ and $\beta_j,\ j \in M' \setminus J'$ are nonnegative and have to be chosen such that the inequality (10) remains valid, see e.g. Nemhauser and Wolsey (1988).

Crowder, Johnson and Padberg (1983) developed an algorithm for finding violated lifted cover inequalities (10), and such an algorithm has been implemented in for instance the software package MINTO. As long as MINTO "recognizes" the surrogate knapsack structure it will automatically generate violated cover inequalities (10). This suggests us to modify the formulation of the facility location problem by adding the constraints defining the aggregate flow (5), and the aggregate demand constraint (7), and replacing the capacity constraints (2) by the aggregate capacity constraints (6). It total we introduce m new variables and $m+1$ new constraints.

Our next relaxation takes the flow into consideration as well, but again uses aggregate information only. Consider the following *single-node flow polytope*, X^{SNF}.

$$X^{\mathrm{SNF}} = \{(v, y) \in \mathbb{R}_+^m \times \mathbb{Z}_+^m : \sum_{j \in M} v_j = d(N),\ v_j \leq m_j y_j,\ y_j \leq 1,\ j \in M\}.$$

The single-node flow model is illustrated in Figure 2.

Proposition 3 (Aardal, Pochet, Wolsey (1995).) $X^{\mathrm{SNF}} \supseteq X^{\mathrm{CFL}}$.

Definition 1 *A subset $J \subseteq M$ is called a* flow cover *with respect to N if $\sum_{j \in J} m_j = d(N) + \lambda$, where $\lambda > 0$.*

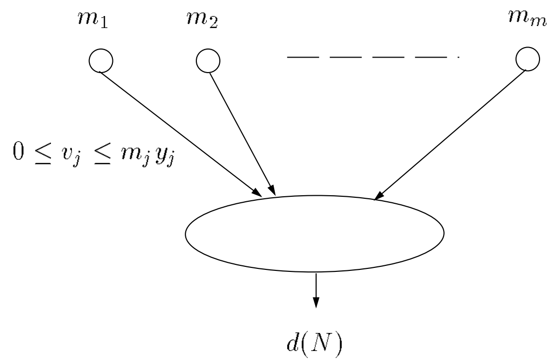


Figure 2: The single-node flow relaxation of the location problem.

If we open all arcs $j \in J$, then the maximum flow along these arcs is equal to $d(N)$, since the set J has excess capacity $\lambda > 0$. If we close arc $k \in J$ and leave all arcs in $J \backslash \{k\}$ open, then the maximum flow along the open arcs is equal to $\min\{d(N), \sum_{j \in J} m_j - m_k\} = \min\{d(N), d(N) - (m_k - \lambda)\} = d(N) - (m_k - \lambda)^+$. This is the intuition behind the *flow cover inequalities* (11) that have been developed for X^{SNF}, see Padberg, Van Roy and Wolsey (1985).

$$\sum_{j \in J} v_j \leq d(N) - \sum_{j \in J} (m_j - \lambda)^+ (1 - y_j) \tag{11}$$

As long as there exists at least one $j \in J$ such that $m_j > \lambda$, the flow cover inequalities (11) define facets of $\mathrm{conv}(X^{\mathrm{SNF}})$. For $\mathrm{conv}(X^{\mathrm{CFL}})$ we obtain a similar result.

Theorem 4 (Aardal, Pochet, Wolsey (1995).) *Let $J \subseteq M$ be a flow cover with respect to N. If $\max_{j \in J} m_j > \lambda$, then the flow cover inequality (11) defines a facet for* $\mathrm{conv}(X^{\mathrm{CFL}})$.

An algorithm for solving the separation problem based on the flow cover inequalities approximately, was presented by Van Roy and Wolsey (1987). Again, since such an algorithm belongs to the systems features of MINTO, flow cover inequalities will be generated automatically given the formulation (1), (3)–(7). It is worth noting that we can generate the cover inequalities (10), and the flow cover inequalities (11) for any subset $K \subseteq N$.

Aardal, Pochet, and Wolsey (1995) have studied the *effective capacity inequalities* and the *submodular inequalities* for X^{CFL}, which can both be viewed as generalizations of the flow cover inequalities.

We conclude this section by giving an example of a knapsack cover inequality and a flow cover inequality.

Example 1 Consider the following instance of the capacitated facility location problem, see Figure 3. First, we consider a knapsack cover inequality. Let $M' = \{2, 3, 4, 5, 6\}$. The set $J' = \{2, 4\}$ defines a cover with respect to M' and N. The inequality

$$y_2 + y_4 \geq 1$$

defines a facet of $\mathrm{conv}(X^{\mathrm{CFL}}) \cap \{y \in \{0,1\}^6 : y_3 = y_5 = y_6 = 1,\ y_1 = 0\}$. If we lift variable y_3 we need to find a value of the lifting coefficient $\beta \geq 0$ such that the inequality $\beta y_3 + y_2 + y_4 \geq 1 + \beta$ is valid. By choosing the maximum value of β we increase the dimension of the face induced by the lifted inequality by one. The maximum value of β is found by solving the following problem:

$$\beta = \min[y_2 + y_4 : \{(y_2 + y_4) \in \{0,1\}^2 : 25y_2 + 20y_4 \geq 55 - 15y_5 - 10y_6,\ y_5 = y_6 = 1\}] - 1 = 1,$$

which yields the inequality

$$y_2 + y_3 + y_4 \geq 2.$$

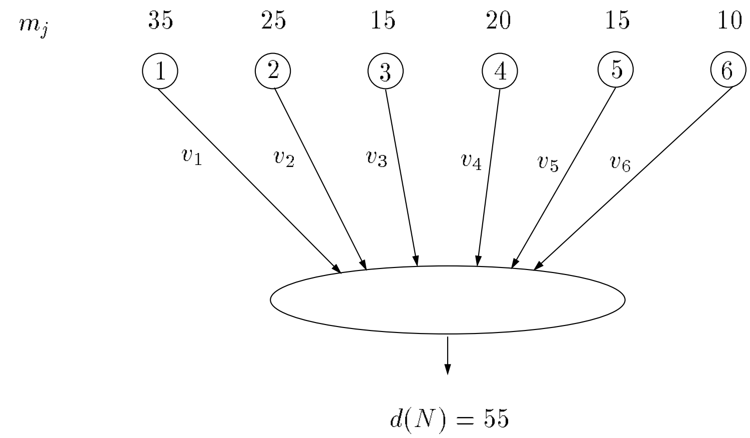


Figure 3: Aggregated facility location instance.

The lifting coefficient of variables $y_5,\ y_6$ are both equal to zero. Next, we lift the variable y_1 that is currently set equal to zero. Here we want to find the minimum value of $\alpha \geq 0$ such that the inequality $\alpha y_1 + y_2 + y_3 + y_4 \geq 2$ is valid, wich is done by solving the problem

$$\alpha = 2 - \min[y_2 + y_3 + y_4 : \{(y_2 + y_3 + y_4 \in \{0,1\}^3 : 25y_2 + 15y_3 + 20y_4 + 15y_5 + 10y_6 \geq 55 - 35\}] = 2.$$

We have now obtained the inequality

$$2y_1 + y_2 + y_3 + y_4 \geq 2,$$

which defines a facet of $\text{conv}(X^{\text{CFL}})$.

Next, we will give an example of a flow cover inequality. The set $J = \{2, 4, 5, 6\}$ defines a flow cover with respect to N. The excess capacity $\lambda = 70 - 55 = 15$, which gives rise to the flow cover inequality $v_2 + v_4 + v_5 + v_6 \leq 55 - 10(1 - y_1) - 5(1 - y_2)$. If we express this inequality in the flow variables v_{jk} we obtain:

$$\sum_{k \in N} v_{2k} + \sum_{k \in N} v_{4k} + \sum_{k \in N} v_{5k} + \sum_{k \in N} v_{6k} \leq 55 - 10(1 - y_1) - 5(1 - y_2).$$

The above inequality defines a facet of $\text{conv}(X^{\text{CFL}})$. ∎

3 The two-level capacitated facility location problem.

A typical application of the two-level capacitated facility location problem is as follows. We are given a set of clients, as in the single level case, and two sets of different facilities. The "higher-level", or major, facilities could for instance be regional distribution centers, and the "lower-level", or minor, facilities could be local distribution centers. The goods are transported form the regional distribution centers by large trucks to the local ones, and from there the goods are distributed further to the clients.

Modeling the two-level problem is slightly less straightforward than the one-level problem. Two obvious ways of formulating the problem is the "multicommodity formulation" and the "flow formulation". In the multicommodity formulation we consider the flow on the path (i, j, k), where i is a major facility, j is a minor facility, and k is a client. In the flow formulation we consider the flow at each level, and require conservation of flow at the minor depot level. We can prove, see Aardal (1992), that the linear relaxation of the multicommodity formulation is at least as strong as the linear relaxation of the flow formulation, and for many instances the difference can be quite large. A drawback with the multicommodity formulation is, however, that it grows rapidly as the size of the problem instance grows. For realistic problem sizes this means that it is impossible to keep all variables in the formulation. An alternative would then be to consider column generation rather than constraint generation. We however wanted to investigate whether it was possible to strengthen the flow formulation in an efficient way to obtain a lower bound of the same quality as the lower bound produced by the multicommodity formulation. For a set of relatively small instances we could conclude that the time needed to improve the linear relaxation of the flow formulation to the level produced by the linear relaxation of the multicommodity formulation by letting MPSARX generate various types of valid inequalities, was shorter than the time it took to solve the linear relaxation of the multicommodity formulation, see further Section 4.

Let v_{jk} and y_j be as in the formulation of the single-level problem, and let $L = \{1, \ldots, l\}$ be the set of major facilities. Let x_i be one if a major facility is opened at site $i \in L$, and zero otherwise, and let w_{ij} denote the flow between major facility i and minor facility j. The flow formulation of the two-level problem is given below.

$$\min\{\sum_{i \in L} \sum_{j \in M} d_{ij} w_{ij} + \sum_{j \in M} \sum_{k \in N} c_{jk} v_{jk} + \sum_{i \in L} g_i x_i + \sum_{j \in M} f_j y_j : (v, y) \in X^{\mathrm{TCFL}}\}$$

where

$$\begin{aligned}
X^{\mathrm{CFL}} = \{(w, v, x, y) \in \mathbb{R}_+^{l \times m} \times \mathbb{R}_+^{m \times n} \times \mathbb{Z}_+^l \times \mathbb{Z}_+^m :&& \\
\sum_{j \in M} w_{ij} \leq&\ u_i x_i, \quad i \in L, && (12)\\
\sum_{i \in L} w_{ij} \leq&\ m_j y_j, \quad j \in M, && (13)\\
\sum_{i \in L} w_{ij} - \sum_{k \in N} v_{jk} =&\ 0, \quad j \in M, && (14)\\
\sum_{j \in M} v_{jk} =&\ d_k, \quad k \in N, && (15)\\
\sum_{k \in N} v_{jk} \leq&\ m_j y_j, \quad j \in M, && (16)\\
v_{jk} \leq&\ d_k y_j, \quad j \in M,\ k \in N, && (17)\\
x_i \leq&\ 1 \quad i \in L, && (18)\\
y_j \leq&\ 1, \quad j \in M\}. && (19)
\end{aligned}$$

Again we will assume that X^{TCFL} is full-dimensional, which is the case if $\sum_{j \in M} m_j - m_r \geq d(N)$, for all $r \in M$, and if $\sum_{i \in L} u_i - u_s \geq d(N)$, for all $s \in L$. The two-level problem can be illustrated by the network shown in Figure 4.

Since the one-level problem is a relaxation of the two-level problem, we can use the inequalities mentioned in Section 2 for the two-level problem as well.

Theorem 5 *Let $u_{\min} = \min_{i \in I} u_i$. If $I \subset L$ is a minimal cover with respect to L and N, and if $\sum_{i \in L \setminus I} u_i + u_{\min} > d(N)$, then the cover inequality*

$$\sum_{i \in I} x_i \geq 1 \tag{20}$$

defines a facet of $\mathrm{conv}(X^{\mathrm{TCFL}}) \cap \{x \in \{0,1\}^l : x_i = 1,\ l \in L \setminus I\}$.

The cover inequality $\sum_{j \in J} y_j \geq 1$, (9), defines a facet of $\mathrm{conv}(X^{\mathrm{TCFL}}) \cap \{y \in \{0,1\}^m : y_j = 1,\ j \in M \setminus J\}$ under the same condition as given in Theorem 2, and by applying lifting to inequalities (9) and (20), as described in Section 2, we can obtain cover inequalities containing variables whose indices do not belong to the cover, c.f. inequality (10).

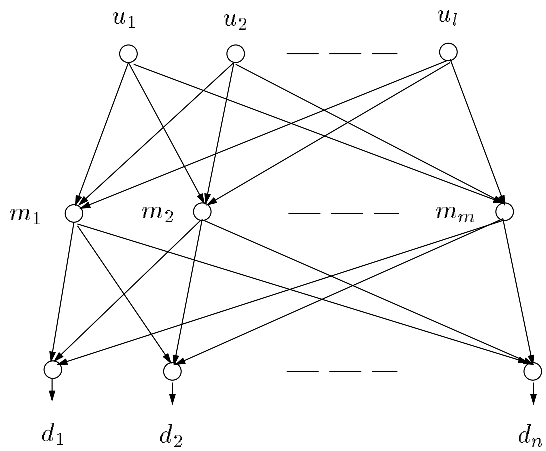


Figure 4: Network representation of the two-level problem.

Similarly, we can generate flow cover inequalities for both levels. The flow cover inequalities (11) define facets of conv(X^{TCFL}) usder the same condition as given in Theorem 4, and for the major level we obtain a similar result. Let

$$w_i - \sum_{j \in M} w_{ij} = 0. \tag{21}$$

Theorem 6 *Let $I \subseteq L$ be a flow cover with respect to N, and let $\sum_{i\in I} u_i = d(N) + \lambda$ where $\lambda > 0$. If $\max_{i\in I} u_i > \lambda$, then the flow cover inequality*

$$\sum_{i\in I} w_i \leq d(N) - \sum_{i\in I} (u_i - \lambda)^+ (1 - x_i) \tag{22}$$

defines a facet of conv(X^{TCFL}).

In order for MPSARX or MINTO to generate the knapsack and flow cover inequalities automatically we need to reformulate the two-level location problem as follows. Add the defining constraints (5) and (21), and the aggregate demand constraints (7) and

$$\sum_{i\in L} w_i = d(N). \tag{23}$$

Moreover, substitute the current capacity constraints (12), and (16) by the aggregate capacity constraints (6) and

$$w_i \leq u_i x_i. \tag{24}$$

In total we have added $l + m$ variables and $l + m + 2$ constraints.

Next, we will consider a relaxation of the two-level problem that involves both facility levels. Consider the path (i, j, k) given in Figure 5. Let $\bar{m}_{ij} = \min\{u_i, m_j\}$ for $i \in L,\ j \in M$, and let $\bar{d}_{jk} = \min\{m_j, d_k\}$ for $j \in M,\ k \in N$. The structure illustrated in Figure 5 can be described by the following constraints.

$$w_i - \sum_{p\in M\setminus\{j\}} w_{ip} - w_{ij} = \quad 0, \tag{25}$$

$$w_{ij} + \sum_{l \in I \subset L\setminus\{i\}} w_{lj} - \sum_{q\in N\setminus\{k\}} v_{jq} - v_{jk} = \quad 0, \tag{26}$$

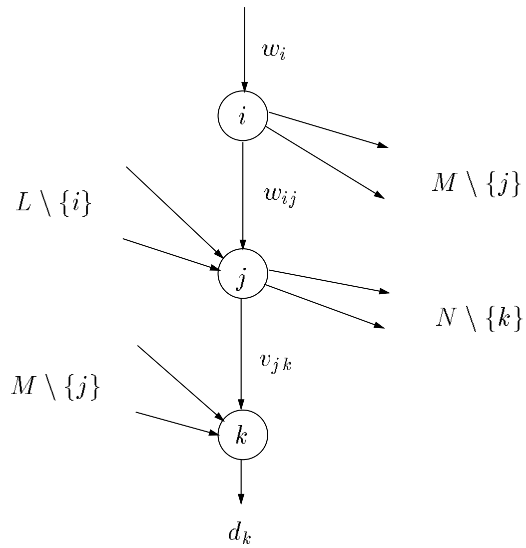


Figure 5: The path relaxation of the two-level problem.

$$\begin{aligned} v_{jk} + \sum_{p \in M \setminus \{j\}} v_{pk} = &\quad d_k, && (27)\\ w_i \leq &\quad u_i x_i, \quad i \in L, && (28)\\ w_{ij} \leq &\quad \bar{m}_{ij} x_i, \quad i \in L,\ j \in M, && (29)\\ v_{jk} \leq &\quad \bar{d}_{jk} y_j, \quad j \in M,\ k \in N. && (30)\end{aligned}$$

Proposition 7 *The polytope X^{P}, defined as all vectors satisfying constraints (25)–(30), forms a relaxation of X^{TCFL}.*

In the *fixed charge path inequalities* (Van Roy and Wolsey (1987)) we want to express an upper bound on the flow on a subset of the inflow arcs in terms of possible outflow. If we for instance consider the inflow to node i, we note that the flow can either continue along the path (i, j, k), or it can exit along a subset of outflow arcs $(M \setminus \{j\}) \cup (N \setminus \{k\})$. For a path of length three we obtain the following family of inequalities.

Proposition 8 *Let $J^+ \subset M \setminus \{j\}$ be a subset of the inflow arcs to node k, and let $J^- \subset M \setminus \{j\}$ be a subset of the outflow arcs from node i. Moreover, let $I \subset L \setminus \{i\}$, and $K \subseteq N \setminus \{k\}$. The following family of fixed-charge path inequalities is valid for X^{TCFL}:*

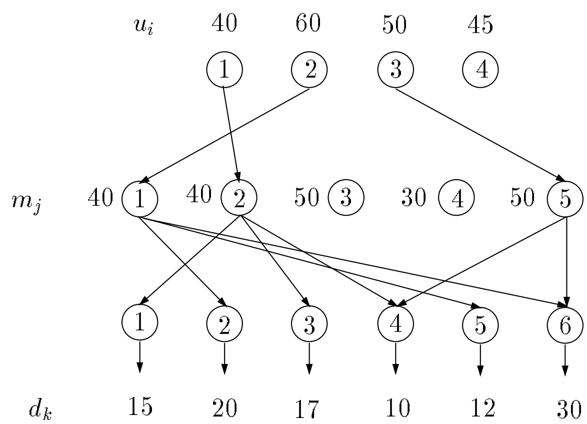
$$\begin{aligned} w_i + \textstyle\sum_{l \in I} w_{lj} + \sum_{p \in J^+} v_{pk} \leq (\sum_{p \in J^-} \bar{m}_{ip} + \sum_{q \in K} \bar{d}_{jq} + d_k) x_i + \\ \sum_{l \in I} (\sum_{q \in K} \bar{d}_{jq} + d_k) x_l + \sum_{p \in J^+} d_k y_p + \sum_{p \in (M \setminus \{j\}) \setminus J^-} w_{ip} + \sum_{q \in (N \setminus \{k\}) \setminus K} v_{jq}. \end{aligned} \quad (31)$$

An example of a path inequality is given below.

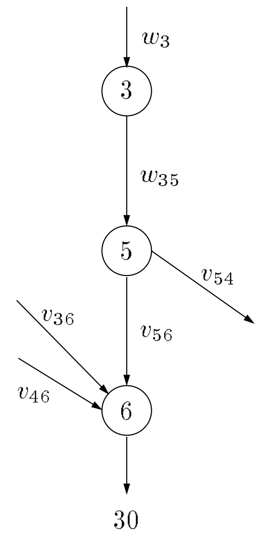
Example 2 Consider the instance of the two-level capacitated facility location problem illustrated in Figure 6 (a). Next, consider the path $(i, j, k) = (3, 5, 6)$ and the subsets of in- and outflow arcs as shown in Figure 6 (b). In this example we have $I = \emptyset$, $J^+ = \{3, 4\}$, $J^- = \emptyset$, and $K = \{4\}$. The inequality

$$w_3 + v_{36} + v_{46} \leq (30 + 10) x_3 + 30 y_3 + 30 y_4 + w_{31} + w_{32} + w_{33} + w_{34} + v_{51} + v_{52} + v_{53} + v_{55}$$

is valid for this particular instance. ∎



(a) The structure of a fractional solution.



(b) The path $(3, 5, 6)$.

Figure 6:

For the software to recognize the path structure it is not necessary to add more redundant information to the model than we already have, but it is important to present the constraints in the right form, as the sign of a variable indicates whether or not it represents inflow or outflow. Therefore it is important for instance to write constraints (21) as $w_i - \sum_{j \in M} w_{ij} = 0$, and not $\sum_{j \in M} w_{ij} - w_i = 0$, (c.f. the node balance constraints (14)).

4 Computational Experience

Here we consider the computational effect of adding knapsack cover inequalities to the linear relaxation of the single-level formulation, and knapsack cover, flow cover and fixed charge path inequalities to the two-level formulation. The interested reader is referred to Aardal (1995) for more computational results regarding the single-level problem, and to Aardal (1992) for results regarding the two-level problem.

We have considered single-level instances of 5 different sizes $a \times b$, where a denotes the number of clients and b the number of facilities. The problem characteristics of the instances are given in Table 1. For each size we have 5 instances. All instances except the ones of size 100×75 were provided by J.-M. Thizy, and were also generated, and used as test instances by Cornuéjols, Sridharan and Thizy (1991). The 100×75-instances were generated by the author according to same principles as the smaller ones. For more details about the instances we refer to Cornuéjols et al.

In Table 2 we use the following notation. Each instance is labeled *aaabbc*, where *aaa* is the number of clients, *bb* the number of facilities, and *c* the number of the instance in the group of instances having the same size. The duality gap is defined as the relative difference between optimum value, z^* and the value of the linear relaxation, z^{LP}, i.e., $(z^* - z^{\text{LP}})/z^*$. The computing time is given in seconds. The percentage duality gap closed is calculated as $(z^{\text{CUT}} - z^{\text{LP}})/(z^* - z^{\text{LP}})$, where z^{CUT} denotes the value of the linear relaxation after all the violated inequalities that have been identified in the root node has been added. We do add violated cover inequalities in all nodes of the branch-and-bound tree.

problem type	# variables	# constraints	# nonzeros
25 × 8	216	242	832
50 × 16	832	883	3,264
50 × 33	1,716	1,767	6,732
50 × 50	2,600	2,651	10,200
100× 75	7,650	7,751	30,300

Table 1: Problem characteristics for the single-level instances.

In the first part of Table 2 we show the duality gap of the instances, the number of branch-and-bound nodes, and the computing time needed to solve the instances to optimality using branch-and-bound only. In the second part of the table we show the effect of letting MINTO automatically add violated knapsack cover inequalities (10). In the last column of Table 2 we present the average decrease in the computing time over the instances having the same size. We use MINTO Version 1.6a, in combination with CPLEX 2.1 Callable Library (CPLEX Optimization, Inc. (1989)), on a SUN Sparc ELC computer.

problem	duality gap (%)	# B&B nodes	time	# cover inequalities	% gap closed	# B&B nodes	time	% time reduction	ave. % time reduction
025081	5.7	31	5	5	94.0	3	4	20.0	
025082	6.4	70	17	10	78.1	9	5	70.6	
025083	5.9	21	6	5	69.0	5	6	0.0	28.9
025084	1.3	9	4	3	100.0	1	4	0.0	
025085	5.5	71	13	4	82.8	7	6	53.8	
050161	2.2	61	59	31	68.1	17	51	13.6	
050162	0.4	19	17	2	48.0	13	20	−17.6	
050163	2.3	143	129	16	82.0	13	48	62.8	22.9
050164	2.1	141	173	33	33.7	35	109	37.0	
050165	0.9	43	43	24	52.3	15	35	18.6	
050331	1.5	1,361	1,813	34	81.8	41	173	90.5	
050332	1.2	797	860	118	65.7	129	476	44.7	
050333	0.2	31	66	0	0.0	31	66	0.0	57.1
050334	1.0	1,349	2,606	130	10.8	175	975	62.6	
050335	1.6	1,617	2,050	46	78.4	49	255	87.6	
050501	0.3	143	278	3	100.0	1	54	80.6	
050502	0.1	67	116	9	37.2	7	93	19.8	
050503	0.4	361	681	10	32.4	7	107	84.3	39.7
050504	0.2	123	283	43	38.9	47	356	−25.8	
050505	0.0	1	75	LP optimal	–	–	–	–	
100751	0.7	4,077	22,977	295	40.9	611	10,560	54.0	
100752	0.6	15,419	74,351	648	55.4	1,423	20,055	73.0	
100753	0.1	183	761	48	12.5	59	844	−10.9	44.7
100754	0.3	6,687	40,604	228	9.1	537	11,076	72.7	
100755	0.1	117	621	9	34.6	23	406	34.6	

Table 2: Results from adding knapsack cover inequalities using MINTO.

The number of branch-and-bound nodes needed to solve the instances to optimality decreased, after adding cover inequalities, for all instances except one (050333), where it remained the same. The computing time also decreased for all instances except 6; for three of these instances the computing time remained the same. For eleven of the twenty-five instances the relative reduction in computing time is larger than 50 %.

Aardal (1995) present computational results from adding flow cover inequalities and generalizations of the flow cover inequalities to the linear relaxation. The average percentage time reduction after adding all these inequalities, including the knapsack inequalities was 30.3 %, 47.7 %, 81.9 %, 72.4 %, and 52.7 % for the different problem sizes.

Next we will present our computational experience with adding knapsack cover, flow cover, and fixed-charge path inequalities to the linear relaxation of the two-level problem. Here we use formulation (13)-(15), (17)-(19), (5)-(7), (21), (23), (24). For convenience of comparison, we give the formulation below.

$$\sum_{i\in L} w_{ij} \leq m_j y_j, \qquad j \in M, \tag{13}$$

$$\sum_{i\in L} w_{ij} - \sum_{k\in N} v_{jk} = 0, \qquad j \in M, \tag{14}$$

$$\textstyle\sum_{j\in M} v_{jk} = d_k, \qquad k \in N, \tag{15}$$

$$v_{jk} \leq d_k y_j, \qquad j \in M,\ k \in N, \tag{17}$$

$$x_i \leq 1, \qquad i \in L, \tag{18}$$

$$y_j \leq 1, \qquad j \in M, \tag{19}$$

$$v_j - \sum_{k\in N} v_{jk} = 0, \qquad j \in M, \tag{5}$$

$$v_j \leq m_j y_j, \qquad j \in M, \tag{6}$$

$$\sum_{j\in M} v_j = d(N), \tag{7}$$

$$w_i - \sum_{j\in M} w_{ij} = 0, \qquad i \in L, \tag{21}$$

$$w_i \leq u_i x_i, \qquad i \in L, \tag{24}$$

$$\sum_{i\in L} w_i = d(N). \tag{23}$$

The instances are generated according to the same principles as the single-level instances. In Table 3 we present the problem characteristics for the two-level instances. We considered small and medium size problems $a \times b \times c$ where a denotes the number of clients, b the number of minor facilities, and c the number of major facilities.

problem type	# variables	# constraints	# nonzeros
25 $\times$ 8$\times$ 4	256	280	1,194
25 $\times$ 16$\times$ 6	540	526	2,430
50 $\times$ 16 $\times$ 6	940	951	4,480

Table 3: Problem characteristics for the two-level instances.

In Table 4 we show the results from solving the two-level instances by pure branch-and-bound. In the branch-and-bound phase we give the x_i-variables higher priority than the y_j-variables, i.e., as long as there are fractional x_i-variables we branch on such a variable before selecting a y_j-variable. We use the notation *aabbcd* for the instances, where *aa* denotes the number of clients, *bb* the number of minor facilities, *c* the number of major facilities, and *d* the number of the instance in the set having the same size. For the computation we use MPSARX (Van Roy and Wolsey (1987)) implemented on a Data General MV 15000 computer. This computer is slow compared to the SUN Sparc ELC, but what we want to demonstrate is the *decrease* in computing time if we add inequalities, and not the absolute time needed to solve the instances.

problem	duality gap (%)	# B&B nodes	time
250841	7.4	87	122.9
250842	8.1	85	93.4
250843	10.8	63	156.7
251661	6.3	353	459.7
251662	6.4	549	883.3
251663	17.2	203	916.5
501661	3.5	301	1,088.8
501662	7.1	75	1,295.6
501663	11.0	27	712.2

Table 4: Results from solving two-level instances by branch-and-bound only.

In Table 5 we present the results after letting MPSARX generate cover, flow cover and fixed-charge path inequalities in the root node of the branch-and-bound tree. These inequalities closed a large part of the duality gap, 70.0–99.3 %. The number of branch-and-bound nodes, and the computing time thereby decreased significantly.

problem	# cover ineq.	# flow cover ineq.	# path ineq.	% gap closed	# B& B nodes	time	% time reduction	ave. % time reduction
250841	6	17	0	93.1	11	60.0	51.2	
250842	6	21	1	87.5	5	59.5	36.3	42.0
250843	4	12	1	99.3	7	96.2	38.6	
251661	5	14	1	96.6	35	118.9	74.1	
251662	4	14	2	92.6	35	242.1	72.6	53.3
251663	3	15	27	83.9	143	796.8	13.1	
501661	4	11	0	70.0	207	867.1	20.4	
501662	3	17	18	94.7	11	516.2	60.2	35.5
501663	2	16	5	80.2	13	528.5	25.8	

Table 5: Results after adding cover, flow cover and fixed-charge path inequalities.

We also compared the multicommodity and flow formulations in terms of quality of the linear relaxations and computing times. Here we used instances having 20 clients, 10 minor facilities, and 5 major facilities. We again used MPSARX implemented on a Data General MV 15000 machine. In Table 6, the first four instances, F-1–F-4, are formulated according to the flow formulation, i.e., formulation (13)-(15), (17)-(19), (5)-(7), (21), (23), (24). The second group of instances, MC-1–MC-4, have been formulated using variables v_{ijk} denoting the flow on the path (i, j, k), and variables x_i, y_j, w_i, v_j as in the flow formulation. "LP-time" gives the time in seconds of solving the initial linear programming relaxation. The MPSARX cuts that have been generated are the same as in Table 5, i.e., cover, flow cover, and fixed-charge path inequalities. The cuts are generated in the root node only, and "time cuts" gives the time in seconds spent on constraint generation. The results indicate that the initial linear relaxation of the flow formulation is weaker than the multicommodity relaxation, which indeed can be mathematically proved, whereas the quality of the lower bound after adding the cuts to the flow formulation is of the same quality as the bound obtained from the multicommodity formulation after adding cuts. The flow formulation bound z^{CUT} is also obtained in shorter time. It is clear that if one would like to use the multicommodity formulation, then one should avoid to work with all variables present in the formulation. A column generation approach, possibly in combination with row generation, could then be an interesting alternative.

model	# vars.	# constr.	LP time	z^{LP}	# MPSARX cuts	time cuts	z^{CUT}	% gap closed	# B&B nodes	B&B time	z^*	total time
F-1	280	294	12.1	1,627.4	20	11.5	1,685.2	80.8	53	37.7	1,699	61.3
F-2	280	294	8.6	1,797.4	21	21.6	1,866.5	84.7	9	9.0	1,879	39.2
F-3	280	294	11.3	1,921.3	22	16.6	2,003.2	62.6	39	43.8	2,052	71.7
F-4	280	294	9.3	2,188.7	23	24.8	2,280.8	83.3	19	19.7	2,302	53.8
MC-1	1,030	354	61.5	1,657.2	7	135.6	1,670.5	31.9	9	24.0	1,699	221.1
MC-2	1,030	354	100.8	1,800.9	12	129.4	1,866.4	84.0	11	22.4	1,879	252.6
MC-3	1,030	354	73.0	1,970.1	14	114.1	2,009.8	48.5	35	118.6	2,052	305.7
MC-4	1,030	354	91.6	2,193.0	12	120.8	2,264.0	65.2	25	53.6	2,302	266.0

Table 6: Comparing the flow and multicommodity formulations.

The computational study indicates that the inequalities that have been developed for the rather drastic knapsack, single-node flow, and path relaxations of the facility location problems are surprisingly effective, and can be generated fast. These inequalities are therefore computationally useful when solving the location problems. When solving large-scale instances we need to close the duality gap even more to limit the size of the branch-and-bound tree, and for this purpose we need inequalities that are designed specifically for these problems. Such inequalities have been developed by e.g. Aardal, Pochet and Wolsey (1995) for the single-level problem, and to some extent by Aardal (1992) for the two-level problem.

References

K. Aardal (1992). On the solution of one and two-level capacitated facility location problems by the cutting plane approach. Ph.D. Thesis Université Catholique de Louvain, Lovain-la-Neuve, Belgium.

K. Aardal (1995). Capacitated facility location: Separation algorithms and computational experience. Report UU-CS-1995-35, Department of Computer Science, Utrecht University. To appear in *Mathematical Programming*.

K. Aardal and C.P.M. Van Hoesel (1995). Polyhedral techniques in combinatorial optimization II: Computations. Report UU-CS-1995-42, Department of Computer Science, Utrecht University (submitted).

K. Aardal and C.P.M. Van Hoesel (1996). Polyhedral techniques in combinatorial optimization I: Theory. *Statistica Neerlandica* 50, 3–26.

K. Aardal, Y. Pochet and L.A. Wolsey (1995). Capacitated facility location: Valid inequalities and facets. *Mathematics of Operations Research* 20, 562–582.

V. Chvátal (1975). On certain polytopes associated with graphs. *Journal of Combinatorial Theory B* 18, 138–154.

G. Cornuéjols, R. Sridharan and J.-M. Thizy (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research* 50, 280–297.

CPLEX Optimization, Inc. (1989). Using the CPLEX Callable Library.

Crowder, Johnson and Padberg (1983). Solving large-scale zero-one linear programming problems. *Operations Research* 5, 803–834.

J. Edmonds (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards (B)* 69, 67–72.

J.M.Y. Leung and T.L. Magnanti (1989). Valid inequalities and facets of the capacitated plant location problem. *Mathematical Programming* 44, 271–291.

M.W.P. Savelsbergh, G.C. Sigismondi and G.L. Nemhauser (1995). "Functional description of MINTO, a Mixed INTeger Optimizer". *Operations Research Letters* 15, 47–58.

G.L. Nemhauser and L.A. Wolsey (1988). *Integer and Combinatorial Optimization*, Wiley & Sons, Inc., New York.

M.W. Padberg, T.J. Van Roy, and L.A. Wolsey (1985). Valid inequalities for fixed charge problems. *Operations Research* 33, 842–861.

T.J. Van Roy and L.A. Wolsey (1987). Solving mixed integer programming using automatic reformulation. *Operations Research* 35, 45–57.