Mathematical Methods of Operations Research (1996) 44:267–290

Accelerating Benders' Decomposition for the Capacitated Facility Location Problem[1]

PAUL WENTGES

Universität Ulm, Abt. Betriebswirtschaft, 89069 Ulm, Germany

Abstract: Discrete facility location problems are attractive candidates for decomposition procedures since two types of decisions have to be performed: on the one hand the yes/no-decision where to locate the facilities, on the other hand the decision how to allocate the demand to the selected facilities. Nevertheless, Benders' decomposition seems to have a rather slow convergence behaviour when applied for solving location problems. In the following, a procedure will be presented for strengthening the Benders' cuts for the capacitated facility location problem. Computational results show the efficiency of the modified Benders' decomposition algorithm. Furthermore, the pareto-optimality of the strengthened Benders' cuts in the sense of [Magnanti and Wong 1990] is shown under a weak assumption.

Key Words: Facility location, Mixed integer programming, Benders' decomposition and Pareto-optimality.

1 Introduction

The capacitated facility location problem is – besides simple plant location and P-median problems – one of the most important discrete location problems and serves as a basic component of many other location models. The capacitated facility location problem can be formulated as the following mixed-integer programming problem (*CFLP*):

[1] This paper was written when the author was at the Institute for Operations Research, University of St. Gallen, Switzerland, and partly supported by "Schweizerischer Nationalfond zur Förderung der wissenschaftlichen Forschung (Grant 12-30140.90)".

0340–9422/96/44:2/267–290 $2.50

$$
(CFLP)\quad
\begin{cases}
\min \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \ , & \\
\sum_{j=1}^{n} x_{ij} = d_i \quad \forall i = 1, \dots, m \ , & (D) \\
x_{ij} \le d_i y_j \quad \forall i = 1, \dots, m, j = 1, \dots, n \ , & (B) \\
\sum_{i=1}^{m} x_{ij} \le s_j y_j \quad \forall j = 1, \dots, n \ , & (C) \\
x_{ij} \ge 0, y_j \in \{0, 1\} \quad \forall i = 1, \dots, m, j = 1, \dots, n \ , & (G)
\end{cases}
$$

where m represents the number of customers, n the number of potential facility locations, d_i the demand of customer i for a certain commodity, and s_j the capacity of facility j to supply the customers with the commodity considered. c_{ij} are the costs of transporting one unit of the commodity from facility j to customer i, f_j are the fixed costs of maintaining/building facility j. The variable y_j indicates whether facility j is open ($y_j = 1$) or not, and x_{ij} represents the amount shipped from facility j to customer i. The aim is to select facility sites in such a way that the total distribution costs – consisting of the fixed costs of maintaining/building the facilities and the variable transportation costs – are minimized while the demand of all customers is met (constraints (D)) and the capacity restrictions on all facilities are satisfied (constraints (C)).

Other applications of $(CFLP)$ include, for example, the optimal design of computer and telecommunication networks (see [Kochmann and McCallum 1981], [Mirzaian 1985] or [Boffey 1989]) or problems in production planning (see [Pochey and Wolsey 1988]).

In the following, it will be assumed without loss of generality that $d_i \le s_j \ \forall i, j$ (otherwise replace constraints (B) by $x_{ij} \le \min\{d_i, s_j\} \cdot y_j$). Sometimes $(CFLP)$ is formulated without the restrictions (B) because the restrictions (B) result from (C) and (G). In this case, one speaks of the *weak* formulation of $(CFLP)$ since the LP-relaxation of $(CFLP)$ then yields inferior lower bounds. The chosen *strong* formulation proves to be very helpful for the construction of strong Benders' cuts.

$(CFLP)$ is strongly NP-hard (see [Cornuejols et al. 1991, p. 289]). The cross decomposition algorithm of [Van Roy 1986] is often viewed to be the most efficient algorithm for solving $(CFLP)$ (at least for smaller instances of $(CFLP)$, e.g. see [Beasley 1988]). The basic idea of the cross decomposition algorithm is to exploit simultaneously both the primal structure (temporarily fixing the variables y yields a normal transportation problem) and the dual structure (with relaxing the capacity constraints (C) in Lagrangean fashion) of $(CFLP)$, i.e. to combine Benders' decomposition and Dantzig-Wolfe decomposition in one single framework. Nevertheless, [Cornuejols et al. 1991] report that the cross decomposition algorithm often yields bad results for harder problem instances (the same observation was made in [Wentges 1994]). Therefore, Cornuejols proposes a Lagrangean heuristic based on relaxing the customer constraints (D).

The first branch and bound procedures for solving $(CFLP)$ use LP-relaxations of both the weak and the strong formulation. More efficient branch and bound algorithms from [Nauss 1978], [Christofides and Beasley 1983], [Beasley 1988], and [Ryu 1993] all rely on (different) Lagrange relaxations of $(CFLP)$. In [Jacobsen 1983], a number of primal heuristics is presented. A more detailed survey of procedures for solving $(CFLP)$ can be found in [Van Roy 1986], [Cornuejols et al. 1991], [Ryu 1993] or [Wentges 1994].

In this paper, procedures to construct strong Benders' cuts in order to accelerate Benders' decomposition for $(CFLP)$ are presented. In the next section, the Benders' decomposition algorithm is introduced, and various methods to accelerate this procedure are briefly reviewed. The algorithms to strengthen the Benders' cuts are developed in section 3. In section 4, the pareto-optimality of some of these modified Benders' cuts is shown under a rather weak assumption on the actual set of open facilities. Last but not least, section 5 contains computational results. The modified Benders' decomposition algorithm was tested on the problem sets described in [Beasley 1988] and [Beasley 1990].

Throughout the paper, the following notation is used: $I := \{1, \ldots, m\} \mathrel{\hat{=}}$ set of customers, and $J := \{1, \ldots, n\} \mathrel{\hat{=}}$ set of potential locations. For fixed $\tilde{y}$ let O be the set of the open facilities ($O = O(\tilde{y}) = \{j: \tilde{y}_j = 1\}$) and C the set of the closed facilities ($C = C(\tilde{y}) = \{j: \tilde{y}_j = 0\}$). The feasible set of an optimization problem (P) is denoted $F(P)$, its optimal value $v(P)$. Furthermore, conv(S) is the convex hull of the set S.

2 Benders' Decomposition

The Benders' (or primal) decomposition procedure for solving linear mixed-integer programming problems, introduced in [Benders 1962], is based on the fact that every linear mixed-integer program can be divided into two classes of partial problems separating the continuous from the integer variables: the so-called Benders' subproblems with only continuous variables, and the Benders' master problems with the complicating integer variables (and one additional continuous variable). Exploiting this property by solving successively and repeatedly one problem of each class can lead – dependent on the structure of the problem – to efficient solution procedures.

Applications of Benders' decomposition for solving location problems can be found e.g. in [Balas 1965] for the uncapacitated facility location problem or in [Magnanti and Wong 1981] and [Magnanti and Wong 1990] for P-median problems. [Geoffrion and Graves 1974] describe the efficient solution of a multicommodity distribution problem with Benders' decomposition. More applications are discussed in [Magnanti and Wong 1990, p. 248 ff.].

The capacitated facility location problem is an attractive candidate for Benders' decomposition since it inhabits an efficiently exploitable primal structure. Fixing the binary location variables $y = (y_j)_{j=1}^n$ to $\tilde{y} = (\tilde{y}_j)_{j=1}^n$ produces the following Benders' subproblems $(SP_{\tilde{y}})$

$$
(SP_{\tilde{y}}) \quad \begin{cases} \min \sum_{j=1}^{n} f_j \tilde{y}_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \ , & \\ \sum_{j=1}^{n} x_{ij} = d_i \quad \forall i = 1, \ldots, m \ , & (D) \\ x_{ij} \le d_i \tilde{y}_j \quad \forall i = 1, \ldots, m, j = 1, \ldots, n \ , & (B) \\ \sum_{i=1}^{m} x_{ij} \le s_j \tilde{y}_j, \quad \forall j = 1, \ldots, n \ , & (C) \\ x_{ij} \ge 0, \quad \forall i = 1, \ldots, m, j = 1, \ldots, n \ . & \end{cases}
$$

The dual $(DSP_{\tilde{y}})$ of $(SP_{\tilde{y}})$ can be written as

$$
(DSP_{\tilde{y}}) \quad \begin{cases} \max \sum_{i=1}^{m} d_i \lambda_i + \sum_{j=1}^{n} \left(f_j - \sum_{i=1}^{m} d_i v_{ij} - s_j \mu_j \right) \cdot \tilde{y}_j \ , \\ \lambda_i - v_{ij} - \mu_j \le c_{ij} \quad \forall i = 1, \ldots, m, j = 1, \ldots, n \ , \\ \lambda_i \in \mathbb{R}, v_{ij} \ge 0, \mu_j \ge 0 \quad \forall i = 1, \ldots, m, j = 1, \ldots, n \ . \end{cases}
$$

Thus, the subproblems $(SP_{\tilde{y}})$ are relative easy to solve transportation problems. Adding the redundant constraint $(T) \sum_{j=1}^n s_j y_j \ge \sum_{i=1}^m d_i$ to $(CFLP)$ ((T) follows from (D) and (C)), one obtains the following class of master problems (MP_T)

$$
(MP_T) \quad \begin{cases} \min \rho \ , & \\ \sum_{i=1}^{m} d_i \lambda_i^t + \sum_{j=1}^{n} \left(f_j - \sum_{i=1}^{m} d_i v_{ij}^t - s_j \mu_j^t \right) \cdot y_j \le \rho \quad \forall t \in T \ , & (+) \\ \sum_{j=1}^{n} s_j y_j \ge \sum_{i=1}^{m} d_i \ , & \\ \rho \in \mathbb{R}, y_j \in \{0, 1\} \quad \forall j = 1, \ldots, n \ . & \end{cases}
$$

Here $\{(\lambda^t, v^t, \mu^t) : t \in T_{PA}\}$ denotes the set of all extreme points of $F(DSP)$ and $T \subset T_{PA}$ a suitable index set. The constraints $(+)$ are called *Benders' cuts* (belonging to μ^t). In the following let

$$P := \left\{ y = (y_j)_{j=1}^n : \sum_{j=1}^n s_j y_j \geq \sum_{i=1}^m d_i,\ y_j \in \{0, 1\} \quad \forall j = 1, \ldots, n \right\} .$$

It can be shown that $v(CFLP) = v(MP_{T_{PA}})$ (see [Lasdon 1970, section 7.3] or [Minoux 1986, section 8.4]; since $F(SP_{\tilde{y}})$ is non-empty and bounded for all $\tilde{y} \in P$, one does not have to consider the extreme rays of $F(DSP)$). The problem is that the set $F(DSP)$ contains many extreme points which are normally not known at the beginning. Since the exact determination of all extreme points is too time consuming and not all constraints are active in the optimal solution of $(MP_{T_{PA}})$, the following iterative procedure seems reasonable: One chooses a starting solution y^1 and index set $T_P^0 \subset T_{PA}$, solves the primal subproblem (SP_{y^1}), determines the Benders' cut belonging to the optimal dual solution of (SP_{y^1}) and adds the newly constructed Benders' cut to the relaxed Benders' master problem $(MP_{T_P^0})$. Afterwards one solves the relaxed master problem (let the optimal solution be (ρ^1, y^2)) and then subsequently (SP_{y^2}). One continues in this fashion until $v(MP_{T_P^k}) = v(SP_{y^k})$ is valid. Formally, Benders' decomposition applied to $(CFLP)$ can be described as follows:

Algorithm 2.1: Benders' decomposition for solving $(CFLP)$

Notation: Let v_P be the value of the best upper bound found so far and $(x^{\text{best}}, y^{\text{best}})$ the corresponding optimal variables. Let $\varepsilon \geq 0$ be the tolerance level for the solution of $(CFLP)$. T_P^k denotes the index set of the Benders' cuts after k iterations.

Step 1: Initialisation:
Set $T_P^0 := \emptyset$, $v_P := \infty$ and $k := 1$. Determine an arbitrary feasible location vector $y^1 \in P$.

Step 2: Solving the Benders' subproblem:

(i) Solve the transportation problem (SP_{y^k}): Let x^k be an optimal solution.
(ii) Determine an optimal dual solution (λ^k, v^k, μ^k) of (DSP_{y^k}) and add the Benders' cut belonging to (λ^k, v^k, μ^k) to the master problem $(MP_{T_P^k})$ $(T_P^k := T_P^{k-1} \cup \{k\})$.
(iii) If $v(SP_{y^k}) \geq v_P$, go to step 3, otherwise set $v_P := v(SP_{y^k})$ and $(x^{\text{best}}, y^{\text{best}}) := (x^k, y^k)$, respectively. Perform an optimality test: If $v_P \leq v(MP_{T_P^{k-1}}) + \varepsilon$, stop the procedure.

Step 3: Solving the Benders' master problem:

(i) Solve the relaxed master problem $(MP_{T_P^k})$: Let (ρ^k, y^{k+1}) be an optimal solution.
(ii) Optimality test: If $v(MP_{T_P^k}) \geq v_P - \varepsilon$, stop the procedure, otherwise set $k := k + 1$ and return to step 2.

The feasible set $F(DSP)$ (independent of y) of the linear programs (DSP_y) has only a finite number of extreme points. Therefore the Benders' decomposition procedure terminates after a finite number of steps (see [Lasdon 1970, p. 379] or [Minoux 1986, section 8.4.4 and 8.4.5]):

Proposition 2.1:

a) *In every iteration of the Benders' decomposition algorithm 2.1 the following holds:*

$$v(MP_{T_P^{k-1}}) \leq v(MP_{T_P^k}) \leq v(CFLP) \leq v_P \ .$$

(b) *The Benders' decomposition procedure terminates for all $\varepsilon \geq 0$ after a finite number of steps and yields an ε-optimal solution of $(CFLP)$.*

In every iteration of algorithm 2.1, a feasible solution of $(CFLP)$ is constructed. Therefore, Benders' decomposition can be used as a heuristic by stopping algorithm 2.1 after a predetermined maximum number of iterations.

Another advantage of Benders' decomposition is the fact that additional constraints (for example lower or upper bounds on the number of open facilities, specifications of permitted location combinations, precedence relations like "$y_k \leq y_l$") in the location variables y can be handled relatively easy, since the subproblems – unaffected – remain easy to solve transportation problems, and the additional constraints make the structureless master problems only insignificantly more difficult (dependent on the number of additional constraints). Furthermore, Benders' decomposition is an integral part of the cross decomposition algorithm of [Van Roy 1986].

Computational results, however, show that the Benders' decomposition procedure for solving $(CFLP)$ has a very bad convergence behaviour when it is performed as described in algorithm 2.1 without any modifications. In the following, some techniques to improve the convergence behaviour of Benders' decomposition are summarized.

The most important feature in speeding up Benders' decomposition is the development of Benders' cuts to be as "strong" as possible. In the next section, the formal concepts of *dominance* and *pareto-optimality* introduced by [Magnanti and Wong 1981] to measure the quality of Benders' cuts and algorithms for constructing strong cuts for $(CFLP)$ are presented.

Tightly connected with the determination of strong Benders' cuts is an appropriate model formulation. The strong formulation of (*CFLP*) with the constraints (*B*) proves to be clearly superior to the weak formulation, since the dual variables v_{ij} corresponding to the constraints (*B*) play an important role for constructing strong cuts. [Magnanti and Wong 1981] used the concept of *cut richness* to compare different model formulations. They showed that the tighter the LP-relaxation of a model formulation, the stronger the corresponding Benders' cuts.

Very time consuming is the exact solution of the structureless and integer-valued (with one continuous variable) Benders' master problems within every iteration step. Therefore several strategies have been proposed to handle this difficulty. [McDaniel and Devine 1977] suggested the solution of the LP-relaxation of the master problems in the first iterations. Another possibility is solving the master problems only heuristically, for example with the – slightly modified – Balas-Martin heuristic (see [Balas and Martin 1980]). Proceeding in this way has the advantage that, besides saving time, a feasible solution of (*CFLP*) is obtained in every iteration. Furthermore, [Geoffrion and Graves 1974] emphasize that the exact solution of the master problems is not necessary because they contain too little information at the beginning of the decomposition algorithm. Therefore, [Geoffrion and Graves 1974] proposed the so-called ε-method. Here the continuous variable ρ is fixed in every iteration step at a certain value dependent on the value of the best solution of (*CFLP*) found so far, and only a feasible solution of the resulting binary problem is sought. The cross-decomposition algorithm of [Van Roy 1986] is another way to avoid solving the master problems in every iteration step.

In addition, [Magnanti and Wong 1981] point out the importance of good initial cuts. In our computations, good results were achieved using the Benders' cuts corresponding to the optimal solutions of the AddHi- and DropHi-heuristic (see [Jacobsen 1983]) as initial cuts.

3 Deriving Strong Cuts

[Magnanti and Wong 1981] introduced the concept of *dominance* and *pareto-optimality* for general problems (*P*) of the type

$$(P) \quad \begin{cases} \min \rho \ , \\ f(u) + g(u)' \cdot y \leq \rho \quad \forall u \in U \ , \\ \rho \in \mathbb{R}, y \in Y \end{cases}$$

to state the "strongness" of a Benders' cut more precisely:

Definition 3.1:

a) The cut

$$f(u^1) + g(u^1)' \cdot y \leq \rho$$

dominates the cut (or: is *stronger* than the cut)

$$f(u^2) + g(u^2)' \cdot y \leq \rho \ ,$$

if

$$f(u^1) + g(u^1)' \cdot y \geq f(u^2) + g(u^2)' \cdot y$$

holds for all $y \in Y$ – with strictly "$>$" for at least one $y_0 \in Y$.

b) A cut is called *pareto-optimal*, if it is not dominated by any other cut.

Since the Benders' subproblems (SP_y) are highly degenerate, the dual linear programs (DSP_y) have many optimal solutions. Therefore, the right choice of an optimal solution (λ, v, μ) and the corresponding Benders' cut is of crucial importance for the efficiency of the Benders' decomposition algorithm.

Solving the Benders' subproblem $(SP_{\tilde{y}})$ with a normal transportation algorithm, one has to eliminate restrictions (B) and the closed facilities $j \in C$. Furthermore, a fictitious customer "$m + 1$" with the fictitious demand $\sum_{j=1}^{n} s_j \tilde{y}_j - \sum_{i=1}^{m} d_i$ has to be added. The application of the transportation algorithm yields an optimal solution of $(SP_{\tilde{y}})$ and at the same time an optimal solution $(\tilde{\lambda}, \tilde{\mu})$ of the corresponding dual $(DSP_{\tilde{y}})$. Usually one chooses the optimal dual solution with $\tilde{\lambda}_{m+1} = 0$. A first "natural" Benders' cut corresponding to $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ can now be completed as follows:

$$\text{Let } \tilde{v} :\equiv 0 \qquad \text{and} \quad \tilde{\mu}_j := \max_{i=1,\ldots,m} \{\tilde{\lambda}_i - c_{ij}, 0\} \quad \forall j \in C \ .$$

Applying the Benders' decomposition algorithm with these cuts to $(CFLP)$ yields unacceptable computational results. A method for strengthening the natural Benders' cuts was first introduced by [Van Roy 1986]. Van Roy proposes increasing the "Benders' values" $(f_j - \sum_{i=1}^{m} d_i \tilde{v}_{ij} - s_j \tilde{\mu}_j)$ of the closed facilities $j \in C$ with the help of the dual variables v_{ij} since the optimal value $v(DSP_{\tilde{y}})$ of $(DSP_{\tilde{y}})$

$$v(DSP_{\tilde{y}}) = \sum_{i=1}^{m} d_i \tilde{\lambda}_i + \sum_{j=1}^{n} \left(f_j - \sum_{i=1}^{m} d_i \tilde{v}_{ij} - s_j \tilde{\mu}_j \right) \cdot \tilde{y}_j$$

remains unchanged if the variables $\tilde{v}_{ij}$ and $\tilde{\mu}_j$, $j \in C$ are modified (since $\tilde{y}_j = 0$), and the Benders' cuts become stronger as the Benders' values increase. Therefore, the solution of the linear programs

$$\begin{cases} \max f_j - \sum_{i=1}^{m} d_i v_{ij} - s_j \mu_j \ , & (1) \\ \tilde{\lambda}_i - v_{ij} - \mu_j \le c_{ij} \quad \forall i = 1, \ldots, m \ , & (2) \\ v_{ij} \ge 0, \mu_j \ge 0 \quad \forall i = 1, \ldots, m & (3) \end{cases}$$

for all $j \in C$ yields a strong Benders' cut $(\tilde{\lambda}, \bar{v}, \bar{\mu})$ (with, for all $j \in O$, unchanged $\bar{\mu}_j = \tilde{\mu}_j$ and $\bar{v}_{ij} = \tilde{v}_{ij}\ \forall i \in I$). The constraints (2) guarantee that $(\tilde{\lambda}, \bar{v}, \bar{\mu})$ is a feasible solution of $(DSP_{\tilde{y}})$. Since the linear programs (1)–(3) are equivalent to

$$\begin{cases} \max f_j - \sum_{i=1}^{m} (d_i \cdot \max\{0, \tilde{\lambda}_i - c_{ij} - \mu_j\}) - s_j \mu_j \ , \\ \mu_j \ge 0 \end{cases}$$

for all $j \in C$, one obtains the following algorithm for constructing a strong Benders' cut, which dominates the original natural cut:

Algorithm 3.1: Algorithm of Van Roy for strengthening the Benders' cuts

Notation: Let $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ be an optimal dual solution of the transportation problem $(SP_{\tilde{y}})$ with $\tilde{v} \equiv 0$ and $\tilde{\lambda}_{m+1} = 0$.
For all $j \in C$ $(\tilde{y}_j = 0)$ perform the following three steps:

Step 1: Sort the set $I := \{1, \ldots, m\}$ in descending order of the values $\tilde{\lambda}_i - c_{ij}$. Set the index set $I^* := \emptyset$.
Step 2: (i) Determine the customer $\bar{i} \in I$ with the highest value $\tilde{\lambda}_i - c_{ij}$, $i \in I$.
(ii) If $\tilde{\lambda}_{\bar{i}} - c_{\bar{i}j} \le 0$ or $\sum_{i \in \{I^* \cup \bar{i}\}} d_i \ge s_j$, go to step 3. Otherwise set

$$I^* := I^* \cup \{\bar{i}\} \ , \qquad I := I - \{\bar{i}\} \text{ and go to (i) } .$$

Step 3: If $I = \emptyset$, set $\bar{\mu}_j := 0$, otherwise define $\bar{\mu}_j := \max\{0, \tilde{\lambda}_{\bar{i}} - c_{\bar{i}j}\}$. Furthermore set $\bar{v}_{ij} := \tilde{\lambda}_i - c_{ij} - \bar{\mu}_j$ for all $i \in I^*$ and $\bar{v}_{ij} := 0$ for all $i \in I$.

Modifying the natural Benders' cuts in this manner already yields substantial savings in computation time.

Another significant improvement of the convergence behaviour of the Benders' decomposition procedure could be obtained by strengthening the Benders' cuts as described in algorithms 3.2 and 3.3. The central idea of these two algorithms – altering the Benders' values also for the open facilities – can be illustrated as follows:

The Benders' value $f_j - \sum_{i=1}^{m} d_i v_{ij} - s_j\mu_j$ can be interpreted as a measure of the goodness of location j. For the natural Benders' cut $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ with $\tilde{v} \equiv 0$ and $\tilde{\lambda}_{m+1} = 0$ one gets the Benders' values $f_j - s_j\tilde{\mu}_j$. The fixed cost reduction $s_j\tilde{\mu}_j$ is a marginal fixed cost reduction, because the value $\tilde{\mu}_j$ is always dependent on the most disadvantageously (with the "last available" capacity unity) served customer. If x_{ij} is in the basis of the optimal solution of $(SP_{\bar{y}})$, then $\tilde{\lambda}_i - \tilde{\mu}_j = c_{ij}$. With $\tilde{\lambda}_{m+1} = c_{m+1,j} = 0$ it follows that $\tilde{\mu}_j = 0$ if the fictitious customer $m+1$ is served by the open facility j. But then the Benders' value $f_j - s_j\tilde{\mu}_j$ no longer constitutes a good measure for the goodness of this location j if, for example, a certain group of customers can be served efficiently only by facility j.

One solution is the simultaneous increase of both the serving costs $\tilde{\lambda}_i$ of customer i and the supplementary charge $\tilde{v}_{ij(i)}$, which considers the special customer-location relationship (let $j(i)$ be the facility to which customer i is assigned in the primal solution of the actual primal subproblem). The serving costs λ_i should be increased to $\hat{\lambda}_i$ until a second open facility k can supply this customer as good as facility $j(i)$, which means that $\hat{\lambda}_i = c_{ik} + \tilde{\mu}_k$ is valid. Thus the customer i has to pay a fair price for the advantage that he is being served by the location $j(i)$ which is most convenient for him, and this contribution of customer i directly benefits facility $j(i)$ (proportionally to the demand d_i of customer i).

Altogether, all these open facilities are "rewarded" with a lower Benders' value $f_j - \sum_{i=1}^{m} d_i\hat{v}_{ij} - s_j\tilde{\mu}_j$, which can serve certain customers strictly better than all other facilities. Increasing $\tilde{\lambda}_i$ to $\hat{\lambda}_i$ and $\tilde{v}_{ij}$ to $\hat{v}_{ij}$ the objective value

$$v(DSP_{\bar{y}}) = \sum_{i=1}^{m} d_i\hat{\lambda}_i + \sum_{j=1}^{n}\left(f_j - \sum_{i=1}^{m} d_i\hat{v}_{ij} - s_j\tilde{\mu}_j\right)\cdot \bar{y}_j$$

remains unchanged, and the constraints

$$\hat{\lambda}_i \leq c_{ij} + \tilde{\mu}_j + \hat{v}_{ij} \quad \forall i = 1, \ldots, m,\ j \in O$$

are satisfied.

Formally, the algorithm for strengthening the original Benders' cuts can be described as follows:

Algorithm 3.2: Algorithm for strengthening the Benders' cuts without considering the closed facilities

Notation: Let the "natural" dual solution $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ of the actual Benders' subproblem $(SP_{\tilde{y}})$ with $\tilde{\lambda}_{m+1} = 0$, $\tilde{v} \equiv 0$ and $\tilde{\mu}_j = \max_{i=1,\ldots,m}\{0, \tilde{\lambda}_i - c_{ij}\}$ for the closed facilities $j \in C$ be given.

Step 1: For all customers $i = 1, \ldots, m$ perform the following steps:

(i) Determine the smallest (k_i) and the second smallest (z_i) value of the set $\{c_{ij} + \tilde{\mu}_j : j \in O\}$. It follows: $k_i = c_{ij(i)} + \tilde{\mu}_{j(i)} = \tilde{\lambda}_i$ for some $j(i) \in O$.
(ii) Set $\hat{v}_{ij} = 0 \ \forall j \in O, \ j \neq j(i)$. If $w_i := z_i - k_i > 0$ holds, set $\hat{\lambda}_i := \tilde{\lambda}_i + w_i$ and $\hat{v}_{ij(i)} := w_i$. Otherwise set $\hat{\lambda}_i = \tilde{\lambda}_i$ and $\hat{v}_{ij(i)} := 0$.

Step 2: Apply Van Roy's algorithm 3.1 (with the newly constructed $\hat{\lambda}$ from step 1) to determine $\hat{v}_{ij}$ and $\hat{\mu}_j$ for all closed facilities $j \in C$.

The computational results show the positive effect of changing the Benders' cuts with algorithm 3.2 on the convergence behaviour of the Benders' decomposition procedure. The pareto-optimality of the strengthened Benders' cuts is shown in the next section.

Since the closed facilities were not taken into consideration for the increase of $\tilde{\lambda}$ in algorithm 3.2, the serving costs $\tilde{\lambda}_i$ were possibly raised by a large amount, although the customer i could have been served very well by a few facilities which were closed in the actual solution $\tilde{y}$. Therefore, algorithm 3.2 can be altered with even better results depending on the problem structure and the number n of potential facility locations (see section 5) if, in the first step of algorithm 3.2, the closed facilities are also taken into account. This requires, though, that first the values $\tilde{v}_{ij}$ and $\tilde{\mu}_j \ \forall j \in C$ are changed with algorithm 3.1 of Van Roy in order to make a fair comparison of the open with the closed facilities possible.

Correspondingly, one obtains the following algorithm 3.3 to construct strong Benders' cuts:

Algorithm 3.3: Algorithm for strengthening the Benders' cuts with consideration of the closed facilities

Notation: As in algorithm 3.2.

Step 1: Carry out algorithm 3.1 to adjust the dual variables $\tilde{\mu}_j$, $\tilde{v}_{ij}$ for all $j \in C$.
Step 2: For all customers $i = 1, \ldots, m$ do:

(i) Determine the smallest (k_i) and second smallest (z_i) value of the set $\{c_{ij} + \tilde{\mu}_j : j \in O\}$. It follows: $k_i = c_{ij(i)} + \tilde{\mu}_{j(i)} = \tilde{\lambda}_i$ for some $j(i) \in O$.

(ii) Determine either the smallest, second smallest, or third smallest (etc.) element e_i of the set $\{c_{ij} + \bar{\mu}_j : j \in C\}$.

(iii) Calculate $w_i := \max\{0, \min\{z_i - k_i, e_i - k_i\}\}$. Set $v_{ij}^* = 0 \; \forall j \in O, j \neq j(i)$. If $w_i > 0$ holds, set $\lambda_i^* := \tilde{\lambda}_i + w_i$, and $v_{ij(i)}^* := w_i$. Otherwise set $\lambda_i^* = \tilde{\lambda}_i$ and $v_{ij(i)}^* := 0$.

Step 3: Apply again algorithm 3.1 (with the newly constructed λ^* of step 2) to definitely calculate μ_j^*, v_{ij}^* for all closed facilities $j \in C$.

For a more detailed interpretation of the modified Benders' cuts see [Wentges 1994, section 7.1].

4 Pareto-Optimality

In the previous section, the concept of pareto-optimality from [Magnanti and Wong 1981] for a class of problems (P) was presented. [Magnanti and Wong 1981], at the same time, introduced a linear model for the construction of pareto-optimal cuts for problems of type (P). In the following, let the relative interior of a set M be the interior of M relative to the smallest affine space which contains M; for example, the relative interior of a disc in $\mathbb{R}^3$ equals the interior of the disk when it is seen as a circle in $\mathbb{R}^2$ (see [Magnanti and Wong 1990, p. 226]).

Theorem 4.1: (see [Magnanti and Wong 1981, Theorem 1])
Let $\hat{y} \in Y$ be given and $U(\hat{y})$ denote the set of optimal solutions of the optimization problem

$$\max_{u \in U} \{f(u) + g(u)' \cdot \hat{y}\} \; .$$

Furthermore, let y^0 be a any element of the relative interior of the convex hull of Y and u^0 an optimal solution of the problem

$$\max_{u \in U(\hat{y})} \{f(u) + g(u)' \cdot y^0\} \; .$$

It follows:

u^0 *(the cut belonging to u^0, respectively) is pareto-optimal* .

[Magnanti and Wong 1981, S. 472 ff.] emphasized that – applying the above method – the resulting pareto-optimal Benders' cuts are highly dependent on the choice of y^0. Furthermore, one has to decide if the "quality" of the pareto-optimal cuts really justifies the computational burden of constructing them. In [Magnanti and Wong 1990], theorem 4.1 was used to determine pareto-optimal cuts for the P-median problem.

In the following, the pareto-optimality of the cuts constructed with algorithm 3.2 will be shown with the help of theorem 4.1.

Applying theorem 4.1 to $(CFLP)$ with the actual location vector $\tilde{y}$ and the optimal solution $(\tilde{\lambda}, \tilde{\mu})$ of $(DSP_{\tilde{y}})$, one obtains the following corollary:

Corollary 4.1: Let y^0 be an element of the relative interior of the convex hull of $P = \left\{ y = (y_j)_{j=1}^n : y_j \in \{0, 1\} \text{ and } \sum_{j=1}^n s_j y_j \geq \sum_{i=1}^m d_i \right\}$. If (λ^0, v^0, μ^0) is an optimal solution of

$$(M) \begin{cases} \max \sum_{i=1}^m d_i \lambda_i + \sum_{j=1}^n \left(f_j - \sum_{i=1}^m d_i v_{ij} - s_j \mu_j \right) \cdot y_j^0 \ , & \\ \sum_{i=1}^m d_i \lambda_i + \sum_{j \in O} \left(f_j - \sum_{i=1}^m d_i v_{ij} - s_j \mu_j \right) = c \ , & (4) \\ \lambda_i - v_{ij} - \mu_j \leq c_{ij} \qquad \forall i = 1, \ldots, m \ , \quad j = 1, \ldots, n \ , & (5) \\ \lambda_i \in \mathbb{R} \ , \quad \mu_j \geq 0 \ , \quad v_{ij} \geq 0 \ , \qquad \forall i = 1, \ldots, m \ , \quad j = 1, \ldots, n & (6) \end{cases}$$

with $c := v(DSP_{\tilde{y}}) = \sum_{i=1}^m d_i \tilde{\lambda}_i + \sum_{j \in O} (f_j - s_j \tilde{\mu}_j)$, then (λ^0, v^0, μ^0) (the cut belonging to (λ^0, v^0, μ^0)) is pareto-optimal.

Proof: Application of theorem 4.1 with

- $Y := P = \left\{ y = (y_j)_{j=1}^n : y_j \in \{0, 1\} \quad \text{and} \quad \sum_{j=1}^n s_j y_j \geq \sum_{i=1}^m d_i \right\}$;
- $u := (\lambda, v, \mu)$;
- $U := F(DSP) = \{(\lambda, v, \mu)\text{: constraints (5), (6) are satisfied}\}$;
- $f(u) := f(\lambda, v, \mu) = \sum_{i=1}^m d_i \lambda_i$;
- $g(u) := g(\lambda, v, \mu) = \left(f_j - \sum_{i=1}^m d_i v_{ij} - s_j \mu_j \right)_{j=1}^n$

yields the desired result. Here, restriction (4) was added to assure that $(\lambda^0, \nu^0, \mu^0) \in U(\tilde{y})$. □

Restriction (4) now is equivalent to

$$\sum_{i=1}^{m} d_i\lambda_i - \sum_{j \in O}\left(\sum_{i=1}^{m} d_i\nu_{ij} + s_j\mu_j\right) = \sum_{i=1}^{m} d_i\tilde{\lambda}_i - \sum_{j \in O} s_j\tilde{\mu}_j \ .$$

With $c_0 := \sum_{i=1}^{m} d_i\tilde{\lambda}_i - \sum_{j \in O} s_j\tilde{\mu}_j$ and substituting $\sum_{i=1}^{m} d_i\lambda_i$ by

$$\sum_{i=1}^{m} d_i\lambda_i = \sum_{j \in O}\left(\sum_{i=1}^{m} d_i\nu_{ij} + s_j\mu_j\right) + c_0 = \sum_{j=1}^{n}\left(\sum_{i=1}^{m} d_i\nu_{ij} + s_j\mu_j\right)\cdot\tilde{y}_j + c_0 \ ,$$

the objective function of (M) can be written as

$$\max c_0 + \sum_{j=1}^{n} f_j y_j^0 + \sum_{j=1}^{n} (\tilde{y}_j - y_j^0)\cdot\left(s_j\mu_j + \sum_{i=1}^{m} d_i\nu_{ij}\right) \ .$$

Let $\delta_j := \tilde{y}_j - y_j^0$. Since $c_0 + \sum_{j=1}^{n} f_j y_j^0$ is a constant, the following linear program (M') equivalent to (M) is obtained to construct pareto-optimal cuts:

$$(M') \begin{cases} \max \sum_{j=1}^{n}\left(s_j\mu_j + \sum_{i=1}^{m} d_i\nu_{ij}\right)\cdot\delta_j \ , & (7) \\ \sum_{i=1}^{m} d_i\lambda_i - \sum_{j \in O}\left(\sum_{i=1}^{m} d_i\nu_{ij} + s_j\mu_j\right) = c_0 \ , & (8) \\ \lambda_i - \nu_{ij} - \mu_j \le c_{ij} \qquad \forall i = 1, \ldots, m \ , \quad j = 1, \ldots, n \ , & (9) \\ \lambda_i \in \mathbb{R} \ , \quad \mu_j \ge 0 \ , \quad \nu_{ij} \ge 0 \ , \qquad \forall i = 1, \ldots, m \ , \quad j = 1, \ldots, n \ . & (10) \end{cases}$$

There seems to be no efficient procedure to solve (M') for arbitrarily chosen elements y^0 of the relative interior of the convex hull of P which can exploit more than the LP-structure of (M'). However, the following holds:

Theorem 4.2: With the additional assumption of

$$(V^*) \qquad \sum_{j \in O\setminus\{h\}} s_j \ge \sum_{i=1}^{m} d_i \qquad \textit{for all } h \in O$$

on the actual location vector $\tilde{y}$ $(O = \{j\colon \tilde{y}_j = 1\})$, *it follows that the Benders' cut constructed with algorithm* 3.2 *is pareto-optimal.*

Proof: Let

$$\varepsilon_1 := 1 - |O| \cdot a \qquad \text{for} \quad \frac{|C|}{1 + |O| \cdot |C|} < a < \frac{1}{|O|} \ .$$

The assertion of theorem 4.2 follows with corollary 4.1 using the vector $y^0 = (y_j^0)_{j=1}^n$ with

$$y_j^0 := \begin{cases} 1 - a & \text{for } j \in O \ , \\ \varepsilon_1 & \text{for } j \in C \ , \end{cases}$$

because the cut corresponding to the optimal solution $(\lambda^0, \mu^0, \nu^0)$ of the linear program (M) (or (M')) formulated in corollary 4.1 equals for this vector y^0 the cut constructed with algorithm 3.2. This is shown in steps (i)–(iv):

(i) First, it will be proved that y^0 is an element of the interior of the convex hull of P.

α) *y^0 is an element of the convex hull of P*
Let $y^h = (y_j^h)_{j=1}^n$ for all $h \in O$ be defined as

$$y_j^h := \begin{cases} 1 & \text{for } j \in O \backslash \{h\} \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

and $e = (e_j)_{j=1}^n$ with $e_j := 1 \ \forall j \in \{1, \ldots, n\}$. $e \in P$, and with assumption (V^*) one gets $y^h \in P$ for all $h \in O$. It follows

$$(*) \quad y^0 = \sum_{h \in O} a \cdot y^h + \varepsilon_1 \cdot e \ ,$$

because

$$\sum_{h \in O} a \cdot y_j^h + \varepsilon_1 \cdot e_j = (|O| - 1) \cdot a + \varepsilon_1 = 1 - a = y_j^0 \qquad \forall j \in O$$

and

$$\sum_{h \in O} a \cdot y_j^h + \varepsilon_1 \cdot e_j = \varepsilon_1 = y_j^0 \qquad \forall j \in C \ .$$

Since furthermore

$$\varepsilon_1 \geq 0 \ , \qquad a \geq 0 \quad \text{and} \quad \varepsilon_1 + \sum_{j \in O} a = 1 \ ,$$

the first claim is true.

β) *y^0 is an element of the interior of the convex hull of P*
For all $h \in \{1, \ldots, n\}$ let e^h and $\tilde{e}^h$ be defined as

$$e^h := (e_j^h)_{j=1}^n \qquad \text{with } e_j^h := \begin{cases} 1 & \text{for } j = h \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

and

$$\tilde{e}^h := (\tilde{e}_j^h)_{j=1}^n \qquad \text{with } \tilde{e}_j^h := \begin{cases} 0 & \text{for } j = h \ , \\ 1 & \text{otherwise} \ . \end{cases}$$

In addition, let $\tilde{e} := (\tilde{e}_j)_{j=1}^n$ with $\tilde{e}_j = 1$ for all $j \in O$ and $\tilde{e}_j = 0$ for all $j \in C$.

For all $h \in O$ it follows: Taking vector $\tilde{e}^h \in P$ instead of e in the convex combination $(*)$, one obtains the vector $(y^0 - \varepsilon_1 \cdot e^h) \in \text{conv}(P)$; choosing $\tilde{e}$ instead of y^h in $(*)$, the vector $(y^0 + a \cdot e^h) \in \text{conv}(P)$ results.

Analogously, one shows that $y^0 - \varepsilon_1 \cdot e^h$ and $y^0 + a \cdot e^h$ are both elements of the convex hull of P for all $h \in C$.

Choosing δ small enough, for all vectors $(y^0 + \gamma)$ with $\gamma = (\gamma_j)_{j=1}^n$ and $\|\gamma\| < \delta$ one gets:

$$y^0 + n \cdot \gamma_j \cdot e^j \qquad \text{is an element of the convex hull of } P$$

and

$$y^0 + \gamma = \frac{1}{n} \sum_{j=1}^{n} (y^0 + n \cdot \gamma_j \cdot e^j) \ ,$$

thus $(y^0 + \gamma) \in \text{conv}(P)$. It follows that y^0 is an element of the interior of the convex hull of P.

(ii) Let the "natural" optimal solution $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ of $(DSP_{\tilde{y}})$ with $\tilde{\lambda}_{m+1} = 0$ for the fictitious customer $m + 1$ be given. $(\tilde{\lambda}, \tilde{v}, \tilde{\mu})$ is a feasible solution of (M').

Since the coefficient $\delta_j = \tilde{y}_j - y_j^0$ is positive for all $j \in O$ and negative for all $j \in C$, the values of the dual variables $\mu_j = \tilde{\mu}_j$ and $v \equiv \tilde{v} \equiv 0$ for $j \in O$ have to be increased in order to maximize the objective function (7) of (M'). However, increases of μ_j or $v_{ij}, j \in O$ must be compensated at the same time by increases of the dual variables λ_i to avoid the violation of constraint (8) (but without violating constraints (9)).

Since modifying the variables μ_j and v_{ij} causes the same changes in both the objective function (7) and constraint (8), a restriction to potential increases in λ_i and v_{ij} is admissible (usually, increasing μ_j for $j \in O$ is not possible at all because of constraints (8)–(10)).

First of all, one can concentrate on determining optimal values of the dual variables λ_i, v_{ij} for $j \in O$, because v_{ij}, μ_j, $j \in C$ are not represented in constraint (8). Furthermore, violations of constraint (9) induced by increases of λ_i can be compensated by increasing v_{ij} by the same amount and, due to the special choice of y_0, for any $k \in O$ the following holds:

$$\begin{aligned}\sum_{j \in C} (-\delta_j) &= |C| \cdot \varepsilon_1 = |C| \cdot (1 - |O| \cdot a) \\ &= |C| - |O| \cdot |C| \cdot a \\ &< |C| - \frac{|O| \cdot |C| \cdot |C|}{1 + |O| \cdot |C|} \\ &= \frac{|C|}{1 + |O| \cdot |C|} < a = \delta_k \ .\end{aligned}$$

Therefore, simultaneously increasing λ_i and $v_{ij}, j \in O$ is advantageous for the objective function (7) of (M') even if all $v_{ij}, j \in C$ have to be increased by the same amount.

(iii) Now, one can increase the dual variable $\lambda_i, i \in \{1, \ldots, m\}$ provided that only one of the restrictions (9) is violated and therefore only one variable $v_{ij}, j \in O$ has to be increased for compensation, because otherwise constraint (8) will be violated.

However, increasing all λ_i (and some v_{ij}, $j \in O$) as much as possible in order to maximize the objective function (7) exactly corresponds to step 1 of algorithm 3.2.

This can be shown formally as follows:

Let $j(i)$ again be the facility which can supply customer i the best of all open facilities in the actual solution $\tilde{y}$ and thus has the lowest value $c_{ij} + \tilde{\mu}_j$. Fixing μ to $\tilde{\mu}$, without considering the closed facilities $j \in C$, one obtains for all $i = 1, \ldots, m$:

α)

$$\lambda_i - v_{ij(i)} \overset{(9)}{\leq} c_{ij(i)} + \tilde{\mu}_{j(i)} = \tilde{\lambda}_i \ ,$$

so with $v_{ij} \geq 0$ for all $j \in O$ it follows that

$$\lambda_i - \sum_{j \in O} v_{ij} \leq \tilde{\lambda}_i \ .$$

β)

$$\text{Constraint (8)} \Leftrightarrow \sum_{i=1}^{m} d_i \lambda_i - \sum_{j \in O} \sum_{i=1}^{m} d_i v_{ij} = \sum_{i=1}^{m} d_i \tilde{\lambda}_i \ .$$

With α) and β) the following is true for all $i = 1, \ldots, m$:

$$\lambda_i - \sum_{j \in O} v_{ij} = \tilde{\lambda}_i \ .$$

Therefore, (M') can be decomposed into the partial problems (M_i), $i = 1, \ldots, m$

$$(M_i) \begin{cases} \max \sum_{j \in O} \delta_j \cdot d_i \cdot v_{ij} \ , & \\ \lambda_i - \sum_{j \in O} v_{ij} = \tilde{\lambda}_i \ , & (11) \\ \lambda_i - v_{ij} \leq c_{ij} + \tilde{\mu}_j \qquad \forall j \in O \ , & (12) \\ \lambda_i \in \mathbb{R} \ , \quad v_{ij} \geq 0 \qquad \forall j \in O \ , & (13) \end{cases}$$

or

$$\max \left\{ \sum_{j \in O} v_{ij} \text{: Constraints (11)–(13) are satisfied} \right\} \ .$$

Thus, one gets as optimal solution of (M') for all $i = 1, \ldots, m$:

$$\lambda_i = \min\{c_{ij} + \tilde{\mu}_j : j \in O \backslash \{j(i)\}\} \ , \qquad \text{and}$$

$$v_{ij} = \begin{cases} \lambda_i - \tilde{\lambda}_i & \text{for } j = j(i) \ , \\ 0 & \text{for all } j \in O \backslash \{j(i)\} \ , \end{cases}$$

which is the solution constructed with algorithm 3.2, step 1.

(iv) If no further increases of the variables λ_i for all $i \in \{1, \ldots, m\}$ are possible, one has then to optimize the dual variables μ_j and v_{ij} for $j \in C$. But the resulting linear programs for this optimization – without constraint (8), with fixed $\lambda = (\lambda_i)_{i=1}^m$ and after decomposition for all $j \in C$ – are equivalent to the linear programs of Van Roy for constructing strong Benders' cuts. Therefore, the optimization of μ_j, v_{ij} for $j \in C$ can be achieved by step 2 of algorithm 3.2. □

For other elements y^0 of the relative interior of the convex hull of P, the construction of pareto-optimal Benders' cuts with corollary 4.1 does not seem to be very promising, because the solution of (M') without decomposition into the partial problems (M_i) is relatively time consuming.

5 Computational Results

The different algorithms for strengthening the Benders' cuts were tested on the problem sets provided by [Beasley 1990] (see also [Beasley 1988]). In these computations, the primal subproblems (SP_y) were solved with a standard transportation algorithm (here, the modified distribution method as described in [Domschke 1989, chapter 8] was used). The Benders' master problems were solved heuristically by the Balas-Martin heuristic (see section 2) until (i) the value of the Balas-Martin heuristic for $(MP_{T_P^k})$ coincided exactly with the value v_P of the best primal solution (x^{best}, y^{best}) found so far, and additionally $v(SP_{y^k}) = v_P$ held for the heuristic solution y^k of $(MP_{T_P^k})$. (ii) the value $v(MP_{T_P^k})$ determined by the Balas-Martin heuristic was in six iterations greater than v_P (criteria (ii), however, never stopped the algorithm in these test problems), or (iii) the maximum number of iterations (70) was reached. In the beginning of the Benders' decomposition algorithm, the Benders' cuts belonging to the solutions of the AddHi- and DropHi heuristics were added to the master problem $(MP_{T_P^0})$, and y^1 was taken to be the optimal solution of $(MP_{T_P^0})$.

All computations were performed on an "Athena PC AT/486 (33 MHz)", and the algorithms were programmed in Pascal. Because of space limitations, the algorithms were tested only on the problem sets IV–XIII of [Beasley 1990] with up to 50 facilities and 50 customers (problem sets A to C have been omitted).

Table 1 gives the results for the original Benders' decomposition algorithm with the Benders' cuts modified by the algorithm of Van Roy (algorithm 3.1). Table 2 summarizes the results for Benders' decomposition with the Benders' cuts constructed by algorithms 3.2 and 3.3 (here e_i was taken to be the third smallest element, see algorithm 3.3).

In all examples, the optimality of the solutions determined by this implementation of the Benders' decomposition algorithm (with the Balas-Martin heuristic)

Table 1. Benders' decomposition with algorithm 3.1 to strengthen the Benders' cuts

Prob.	LP-value	$v(MP_{T_P})$	v_P	time	it.
IV-1	1032115	1040444	1040444	1.81	8
IV-2	1086406	1098000	1098000	1.92	6
IV-3	1139669	1153000	1153000	0.99	5
IV-4	1217525	1235500	1235500	1.38	6
V-1	1003372	1025208	1025208	17.14	25
VI-1	920486	932616	932616	24.82	28
VI-2	959342	977799	977799	33.83	34
VI-3	985484	1014062	1014062	44.65	35
VI-4	1021996	1045650	1045650	6.81	17
VII-1	922156	932616	932616	24.33	31
VII-2	963082	977799	977799	6.92	17
VII-3	990954	1010641	1010641	4.34	14
VII-4	1015592	1034977	1034977	3.19	12
VIII-1	822700	833031	838499	514.11	70
VIII-2	890137	910890	910890	368.16	64
VIII-3	956309	975890	975890	227.84	53
VIII-4	1044511	1069370	1069370	406.94	65
IX-1	778576	794453	796649	511.80	70
IX-2	826896	849066	856496	460.44	70
IX-3	863269	892627	896618	467.86	70
IX-4	902230	945806	946051	408.59	70
X-1	782486	791995	796649	303.85	70
X-2	833803	854704	854704	113.20	48
X-3	871843	893782	893782	45.98	37
X-4	906421	928942	928942	6.75	15
XI-1	811056	823673	826125	1100.54	70
XI-2	878593	897730	901377	976.58	70

Table 1. (cont.)

Prob.	LP-value	$v(MP_{T_P})$	v_P	time	it.
XI-3	943739	965833	971158	984.97	70
XI-4	1036334	1059833	1063357	1029.19	70
XII-1	773684	789578	793440	1082.36	70
XII-2	823662	845255	853288	980.53	70
XII-3	859291	887721	895303	810.15	70
XII-4	901435	942915	946052	688.98	70
XIII-1	776563	789720	793440	721.72	70
XIII-2	832920	850968	851670	594.08	70
XIII-3	871828	891874	893252	596.99	70
XIII-4	906329	928942	928942	21.20	17

Notation: LP-value is the value of the LP-relaxation of the last solved Benders' master problem. $v(MP_{T_P})$ and v_P denote the same values as in algorithm 2.1 when the procedure is stopped after "it." iterations (time in seconds)

could be verified with the ε-method mentioned in section 2 when the algorithm was stopped by criteria (i) (for almost all examples in only one iteration, but at most in two iterations of the ε-method). For a more detailed description of the ε-method see [Geoffrion and Graves 1974] or [Wentges 1994].

As can be seen in table 1, only 20 of the 37 problems could be solved optimally within 70 iterations when the Benders' cuts were constructed with algorithm 3.1. In contrast, for all problems only a maximum of 30 (59) iterations were needed to get the optimal solution when the Benders' cuts were modified with algorithm 3.3 (or algorithm 3.2). Furthermore, in all examples less iterations were needed for the Benders' decomposition procedure with algorithm 3.3 for strengthening the Benders' cuts than for Benders' decomposition with algorithm 3.1. The LP-relaxation of the last solved Benders' master problem was also tighter for all test problems when the Benders' cuts were modified with algorithm 3.3, even though less Benders' cuts were included in the respective master problems because of the smaller number of performed iterations.

Similar results were achieved in [Wentges 1994] for 25 test problems with 19–108 potential facilities and 65–174 customers on the Swiss road network. It has to be added that, for these more complicated test problems, there were occasions when more than two iterations of the ε-method were needed to prove the optimality of the best primal solution found so far if the Benders' decomposition algorithm was stopped by criteria (i) or (ii). Nevertheless, the total number of iterations could never be reduced by replacing from the beginning the Balas-Martin heuristic with the ε-method.

Table 2. Benders' decomposition with alg. 3.2 and 3.3 to strengthen the Benders' cuts

	Algorithm 3.2				Algorithm 3.3 (e_i = third smallest el.)			
Prob.	LP-value	$v(MP_{T_p}) = v_P$	time	it.	LP-value	$v(MP_{T_p}) = v_P$	time	it.
IV-1	1040444	1040444	0.33	2	1040444	1040444	0.33	2
IV-2	1095480	1098000	0.33	2	1096327	1098000	0.33	2
IV-3	1150608	1153000	0.38	2	1149715	1153000	0.61	2
IV-4	1230510	1235500	0.33	2	1229617	1235500	0.33	2
V-1	1021365	1025208	1.09	5	1021961	1025208	1.16	5
VI-1	931725	932616	0.39	3	931393	932616	0.22	2
VI-2	977176	977799	0.44	3	976173	977799	0.27	2
VI-3	1009946	1014062	5.05	14	1010979	1014062	3.74	11
VI-4	1039346	1045650	2.75	9	1043051	1045650	1.92	7
VII-1	930882	932616	0.16	2	930848	932616	0.38	3
VII-2	977200	977799	0.44	3	976694	977799	0.28	2
VII-3	1006248	1010641	55.91	36	1009383	1010641	0.33	2
VII-4	1021584	1034977	9.94	18	1024241	1034977	0.88	4
VIII-1	835817	838499	4.67	11	835505	838499	5.71	13
VIII-2	905945	910890	8.07	14	906326	910890	5.49	12
VIII-3	971690	975890	8.13	13	971176	975890	6.70	13
VIII-4	1061146	1069370	17.25	20	1062356	1069370	7.58	13
IX-1	793559	796648	4.01	10	794329	796648	3.13	9
IX-2	852262	855734	10.65	16	852744	855734	4.29	11
IX-3	890560	896618	21.92	22	893525	896618	5.66	12
IX-4	932982	946051	19.99	23	938801	946051	13.95	18
X-1	793730	796648	14.72	21	793719	796648	8.57	16
X-2	850284	854704	39.93	31	850759	854704	4.01	10
X-3	885963	893782	149.84	45	890892	893782	5.33	12
X-4	913489	928942	31.42	27	926094	928942	5.49	12
XI-1	819033	826125	329.72	47	820935	826125	13.19	10

Table 2. (cont.)

	Algorithm 3.2				Algorithm 3.3 (e_i = third smallest el.)			
Prob.	LP-value	$v(MP_{T_P}) = v_P$	time	it.	LP-value	$v(MP_{T_P}) = v_P$	time	it.
XI-2	894981	901377	68.38	25	894708	901377	31.58	19
XI-3	958949	970568	50.09	23	959422	970568	54.82	24
XI-4	1052155	1063356	102.27	30	1052999	1063356	52.89	22
XII-1	787747	793440	135.83	35	791272	793440	7.75	9
XII-2	844355	852525	65.41	27	848231	852525	21.80	18
XII-3	886970	895302	46.80	22	888976	895302	53.55	27
XII-4	932149	946051	145.22	38	935476	946051	70.69	30
XIII-1	786947	793440	473.29	59	789932	793440	10.49	11
XIII-2	843522	851495	202.73	45	846376	851495	12.74	13
XIII-3	881925	893077	206.41	40	885051	893077	14.50	13
XIII-4	919381	928942	39.60	22	920016	928942	6.76	9

Notation: As in table 1

In conclusion, Benders' decomposition for solving (*CFLP*) can be accelerated substantially when the Benders' cuts are modified with algorithm 3.2 or algorithm 3.3. Furthermore, it seems to be advantageous to solve the master problems at the beginning only heuristically (e.g. with the Balas-Martin heuristic). In any case, because of the acceleration achieved the Benders' decomposition procedure can be recommended for the exact or – for problems with many potential facility locations – heuristic solution of the capacitated facility location problem, especially since the additional consideration of further restrictions on the set of potential facility locations is possible.

Acknowledgement: The author gratefully acknowledges the valuable support of Professors R. Dillmann and P. Stähly.

References

Balas E (1965) An additive algorithm for solving linear programs with zero-one variables. Operations Research 13:517–546

Balas E, Martin CH (1980) Pivot and complement – A heuristic for 0-1 programming. Management Science 26(1): 86–96
Beasley JE (1988) An algorithm for solving large scale capacitated warehouse location problems. European Journal of Operational Research 33: 314–325
Beasley JE (1990) OR-library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41(11): 1069–1072
Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4: 238–252
Boffey TB (1989) Location problems arising in computer networks. Journal of the Operational Research Society 40(4): 347–354
Christofides N, Beasley JE (1983) Extensions to a lagrangean relaxation approach for the capacitated warehouse location problem. European Journal of Operational Research 12: 19–28
Cornuejols G, Sridharan R, Thizy JM (1991) A comparison of heuristics and relaxations for the capacitated plant location problem. European Journal of Operational Research 50: 280–297
Domschke W (1989) Logistik: Transport (third edition). Oldenbourg Verlag, München
Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. Management Science 20(5): 822–844
Jacobsen SK (1983) Heuristics for the capacitated plant location model. European Journal of Operational Research 12: 253–261
Kochmann GA, McCallum CJ (1981) Facility location models for planning a transatlantic communications network. European Journal of Operational Research 6: 205–211
Lasdon LS (1970) Optimization theory for large systems. Macmillan, London
Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research 29(3): 464–484
Magnanti TL, Wong RT (1990) Decomposition methods for facility location problems. In: Mirchandani PB, Francis RL (Eds) Discrete Location Theory, chapter 5, 209–262. John Wiley & Sons, New York et al
McDaniel D, Devine M (1977) A modified Benders' partitioning algorithm for mixed integer programming. Management Science 24(3): 312–319
Minoux M (1986) Mathematical programming – Theory and algorithms. John Wiley & Sons, Chichester et al
Mirzaian A (1985) Lagrangian relaxation for the star-star concentrator location problem: Approximation algorithm and bounds. Networks 15: 1–20
Nauss RM (1978) An improved algorithm for the capacitated facility location problem. Journal of the Operational Research Society 29(12): 1195–1201
Pochey Y, Wolsey LA (1988) Lot-size models with backlogging: Strong reformulations and cutting planes. Mathematical Programming 40: 317–335
Ryu C (1993) Capacity-oriented planning and scheduling in production and distribution systems. PhD thesis, University of Pennsylvania
Van Roy TJ (1986) A cross decomposition algorithm for capacitated facility location. Operations Research 34(1): 145–163
Wentges P (1994) Standortprobleme mit Berücksichtigung von Kapazitätsrestriktionen: Modellierung und Lösungsverfahren. PhD thesis Nr. 1620 University of St. Gallen

Received: April 1995
Revised version received: September 1995