SPECIAL CASES AND EFFICIENT ALGORITHMS FOR MULTITHREADED DIMENSION REDUCTION IN LOCATION THEORY

JÖRG FLIEGE*

Abstract. Due to inevitable nondifferentiabilities, the minisum multifacility location problem is regarded as hard to solve. It is shown that a certain chararacterization of points at which non-differentiabilities occur is checkable in polynomial time. Therefore, it is possible to check efficiently if several variables can be identified with each other, thereby reducing the dimension of the problem and the number of nondifferentiable terms in the objective function simultaneously. Numerical results confirm that by using this strategy improved solutions can be found in less computation time.

Key words. facility location, coincidence conditions, dimension reduction, nondifferentiability detection, polynomial time algorithms, numerical comparison

AMS subject classifications. 49K10, 90B80, 90B85, 90C06, 90C35, 90C48

In this paper we extend the results presented in [22] by deriving a polynomial time coincidence checking algorithm for multifacility location problems. It is shown that this algorithm detects more coincidences than the conditions proposed in the literature so far. Numerical results are presented for problems with up to 40.000 variables.

1. Preliminaries. We recall in short the notation and the results obtained in [22]. For an arbitrary set Λ an an arbitrary index set I define

$$\Lambda^I := \{(\lambda_i)_{i\in I} \mid \forall\, i \in I : \lambda_i \in \Lambda\},$$

the set of all families $(\lambda_i)_{i\in I}$ with $\lambda_i \in \Lambda$ for all $i \in I$. For a fixed set $X \neq \emptyset$ we define the set Δ by $\Delta := \{(x,x) \in X \times X \mid x \in X\}$, the *diagonal* of $X \times X$.

1.1. Distance Costs. A function $d : X \times X \longrightarrow \mathsf{IR}$ is called *convex*, if for all $x, y, z \in X$ the *triangle inequality*

$$d(x,y) \leq d(x,z) + d(z,y)$$

holds. A function $d : X\times X \longrightarrow \mathsf{IR}$ is called *concave* if $-d$ is convex. For $d : X\times X \longrightarrow \mathsf{IR}$ we define $\overline{d}$ by $\overline{d}(x,y) := d(y,x)$. A function $d : X\times X \longrightarrow \mathsf{IR}$ is called *semi-definite* if for all $x \in X$ the equation $d(x,x) = 0$ holds. For a semi-definite function d we define

$$[d\,] := \begin{cases} d & \text{if } d \text{ is convex,} \\ |\overline{d}| & \text{else.} \end{cases}$$

A semi-definite function $d : X \times X \longrightarrow \mathsf{IR}$ which is either convex or concave is called *distance cost function* on X. A nonnegative convex distance cost function d is also called a *weak metric* [37, 39, 40, 36, 21].

In what follows, we write for given functions $c, d : X \times X \longrightarrow \mathsf{IR}$ the inequality $c \leq d$, if this inequality holds for all arguments $(x,y) \in X \times X$. For semi-definite functions c, d the inequality above is called *strict*, if it is strict for all arguments $(x,y) \in (X \times X) \setminus \Delta$.

*Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany. e-mail `fliege@math.uni-dortmund.de`

1.2. Graphs. Let $F \neq \emptyset$ be an arbitrary set and $E \subseteq F \times F$. Then $G := (F, E)$ is called *graph* with *node set* F and *edge set* E. A 2-tuple $(f, g) \in F \times F$ will also be denoted by fg. For $f \in F$ and $U \subseteq F$ we define the following subsets of F:

$$\begin{aligned} F_G(f, U) &:= \{g \in U \mid (f, g) \in E\}, \\ F_G(U, f) &:= \{g \in U \mid (g, f) \in E\}, \\ \boldsymbol{F}_G(f, U) = \boldsymbol{F}_G(U, f) &:= F_G(f, U) \cup F_G(U, f). \end{aligned}$$

For $U_1, U_2 \subseteq F$ and $f \in F$ define $E_G(U_1, U_2) := E \cap (U_1 \times U_2)$,

$$\boldsymbol{E}_G(U_1, U_2) = \boldsymbol{E}_G(U_2, U_1) := E_G(U_1, U_2) \cup E_G(U_2, U_1),$$

$E_G(f, U) := E_G(\{f\}, U)$, $E_G(U, f) := E_G(U, \{f\})$, as well as

$$\boldsymbol{E}_G(f, U) = \boldsymbol{E}_G(U, f) := E_G(f, U) \cup E_G(U, f).$$

If the graph G is fixed or it is clear which given graph is meant, we will omit the index on $E_G, \boldsymbol{E}_G, F_G$ and $\boldsymbol{F}_G$. Now let $G' := (F', E')$ be a subgraph of G, i. e. we have $F' \subseteq F$ and $E' \subseteq E \cap (F' \times F')$. Then we define $F'(f, U) := F_{G'}(f, U)$ and modify the rest of the notation analogously. The same modifications will be used if a graph is denoted by G_0 or some other symbols.

1.2.1. Flows. Let $G = (F, E)$ be a directed simple graph. Let there be given a nonempty *source set* $Q \subset F$, $Q \neq F$, a nonempty *sink set* $S \subset F$, $S \neq F$ with $Q \cap S = \emptyset$, and for each edge $e \in E$ a *capacity interval* $I_e = [a_e, b_e] \subset \mathrm{IR}$. Then, a family $(\varphi_e)_{e \in E} \in \mathrm{IR}^E$ of real numbers $\varphi_e \in \mathrm{IR}$ $(e \in E)$ is called a (Q, S)*-flow*, if

$$\sum_{e \in E(Q,F)} \varphi_e - \sum_{e \in E(F,Q)} \varphi_e = \sum_{e \in E(F,S)} \varphi_e - \sum_{e \in E(S,F)} \varphi_e,$$

$$\sum_{e \in E(f,F)} \varphi_e = \sum_{e \in E(F,f)} \varphi_e$$

for all $f \in F \setminus (Q \cup S)$ and

$$\varphi_e \in I_e$$

holds for all $e \in E$. A (Q, S)-flow $(\varphi_e)_{e \in E} \in \mathrm{IR}^E$ *saturates* an edge $e \in E$ with interval of capacity $I_e = [a_e, b_e]$, if $\varphi_e = b_e$ holds. If the corresponding capacity intervals for the edges of G, a set of sources and a set of sinks are given, the graph G is also called a *transportation network*. In case the edges in E are *weighted*, i. e. for all $e \in E$ there exists a weight $\omega_e \in \mathrm{IR}$, a family $(\varphi_e)_{e \in E} \in \mathrm{IR}^E$ of real numbers $\varphi_e \in \mathrm{IR}$ $(e \in E)$ is called a *weighted* (Q, S)*-flow*, if

$$\sum_{e \in E(Q,F)} \omega_e \varphi_e - \sum_{e \in E(F,Q)} \omega_e \varphi_e = \sum_{e \in E(F,S)} \omega_e \varphi_e - \sum_{e \in E(S,F)} \omega_e \varphi_e,$$

$$\sum_{e \in E(f,F)} \omega_e \varphi_e = \sum_{e \in E(F,f)} \omega_e \varphi_e$$

for all $f \in F \setminus (Q \cup S)$ and

$$\varphi_e \in I_e$$

for all $e \in E$ holds.

1.3. The Problem. In what follows, let $G = (F, E)$ be a directed simple graph with finite node set F. This node set should be partitioned into two nonempty sets A and V. For all $a \in A$ there should be given an $x_a \in X$. For all $e \in E$ there should be given a distance cost function $d_e : X \times X \longrightarrow \mathbb{R}$. A location $x_v \in X$ for a node $v \in V$ is called *new location* for the object v. The locations $x_a \in X$ $(a \in A)$ are the *known locations* of the optimization problem defined below. Moreover, let there be given a set $C \subseteq X^V$ of feasible families of new locations.

If, in addition to the fixed locations $x_a \in X$ $(a \in A)$, a family of new locations $(x_v)_{v\in V} \in X^V$ is given, we can define for each $(f, g) \in F \times F$ the 2-tuple $x_{fg} := (x_f, x_g) \in X \times X$. If $e = (f, g) \in E$ we will write $x_e := x_{fg}$. If, in addition to $(x_v)_{v\in V}$, a second family $(\tilde{x}_v)_{v\in V} \in X^V$ is given, we will use the obvious definition $\tilde{x}_a := x_a$ $(a \in A)$ and write $\tilde{x}_{fg} := (\tilde{x}_f, \tilde{x}_g)$ for $(f, g) \in F \times F$ and $\tilde{x}_e := \tilde{x}_{fg}$ for $e = (f, g) \in E$. This notation will also be used for additional families $(\hat{x}_v)_{v\in V}, (y_v)_{v\in V} \in X^V$ etc. For example, $\hat{x}_{fg} := (\hat{x}_f, \hat{x}_g)$, $y_e := y_{fg} = (y_f, y_g)$ if $e = (f, g) \in E$ etc.

Definition 1.1 (Minisum Multifacility Location Problem). *The* constrained minisum multifacility location problem with mixed distance cost functions *is the following optimization problem:*

Find a minimum $(\hat{x}_v)_{v\in V} \in X^V$ of the function

$$
\begin{aligned}
\Psi : X^V &\longrightarrow \mathbb{R} \\
(x_v)_{v\in V} &\longmapsto \sum_{e\in E} d_e(x_e)
\end{aligned}
\tag{1.1}
$$

subject to the constraint $(\hat{x}_v)_{v\in V} \in C$.

In what follows, we may assume without loss of generality that all functions d_e $(e \in E)$ are unequal to the constant function 0. On the other hand, we may use on each "nonexisting" edge $(f, g) \notin E$ the distance cost function $d_{fg} \equiv 0$ and add (f, g) to the edge set E. This will be done as soon as it simplifies the notation.

For other assumptions which can be made without loss of generality, the reader is referred to Plastria [34] and Fliege [17].

1.4. Compatible Partitions. Let $E_{\leq} \subseteq E$ and $E_{\geq} \subseteq E$ be two given subsets of E with $E_{\leq} \cap E_{\geq} = \emptyset$. Then, the notation of § 1.2 can be used here immediately. As an example, we set $E_{\leq}(f, U) := E_{\leq} \cap E(f, U)$, $F_{\geq}(U, f) := F_{(F,E_{\geq})}(U, f)$ etc. The obviously arbitrary indices $\leq$ and $\geq$ on the sets $E_{\leq}$ and $E_{\geq}$ will be motivated in § 3. Here, it suffices to say that it will be meaningful to define $E_{\geq}$ as the set of all edges $e \in E$ such that d_e is concave.

Let a number $n \in \mathbb{N}$, $n \geq 2$, be given and $\mathcal{W} = \{W_0, \ldots, W_n\}$ be a given *partition* of F, i. e. we have $W_i \cap W_j = \emptyset$ for $i \neq j$, $i, j = 0, \ldots, n$, $W_i \neq \emptyset$ for $i = 0, \ldots, n$, and $\bigcup_{i=0}^{n} W_i = F$. Then $\mathcal{W}$ is called *compatible with* Ψ, if $|W_0 \cap A| \leq 1$ and $\bigcup_{i=1}^{n-1} W_i \subseteq V$. In this case the set $K := W_0$ is called *kernel* of $\mathcal{W}$ and the set $S := W_n$ is called *shell* of $\mathcal{W}$. For $i \leq n$ define $U_i := \bigcup_{j=0}^{i} W_j$. For $f \in F$ let $W(f)$ be the (well-defined) set $W_i \in \mathcal{W}$ such that $f \in W_i$. Now let $f \in F \setminus S$ and $W_i = W(f)$. Then the set $\mathrm{pre}(f) := U_{i-1}$ is called the *predecessor* of f and the set $\mathrm{succ}(f) := U_{n-1} \setminus U_i$ is called the *successor* of f. If a partition $\mathcal{W} = \{W_0, \ldots, W_n\}$ is given we assume without loss of generality that $\boldsymbol{E}(U_i, F \setminus U_i) = E(U_i, F \setminus U_i)$ $(0 \leq i \leq n-1)$ holds.

Now let $\mathcal{W} = \{W_0, \ldots, W_n\}$ be a partition compatible with Ψ with kernel K and shell S. If another set $E_s \subseteq E(F \setminus S, F \setminus S)$ is given, we define the subgraph $G' = (F, E')$ by $E' := E \setminus E_s$.

1.5. A Coincidence Condition. Let $E_{\leq} \subseteq E$ and $E_{\geq} \subseteq E$ be two arbitrary fixed edge sets with $E_{\leq} \cup E_{\geq} = E$ and $E_{\leq} \cap E_{\geq} = \emptyset$. For example, we may choose $E_{\leq} := \{e \in E \mid d_e \text{ convex}\}$. The following theorem has been proved in [22]. It contains a new coincidence condition useful for reducing the dimension of a problem and the number of nondifferentiable terms in the objective function.

THEOREM 1.2. *Let $\mathcal{W} = \{W_0, \ldots, W_n\}$ be a partition of F which is compatible with Ψ and which has the kernel K and the shell S. Define $U := F \setminus (K \cup S)$. Let $E_s \subseteq E(F \setminus S, F \setminus S)$ be an edge set such that $d_e \geq 0$ holds for all $e \in E_s$ and define $E' := E \setminus E_s$. Assume that $F'_{\leq}(\mathrm{pre}(v), v) \neq \emptyset$ holds for all $v \in U$.*

For all $v \in U$, $w \in W(v) \setminus \{v\}$, $f \in F'_{\leq}(\mathrm{pre}(v), v)$, $g \in F'_{\leq}(\mathrm{pre}(w), w)$, $h \in \mathrm{pre}(v) \setminus F'_{\leq}(\mathrm{pre}(v), v)$, and $s \in S$ there should exist distance cost functions

$$d_s^{f,v}, \quad d_{v,w}^{f,g}, \quad d_{w,v}^{g,f}, \quad d_h^{f,v}, \quad \text{and} \quad d^{f,v},$$

such that the following conditions hold:

1. *The attraction conditions*

$$\sum_{s \in S} [d_s^{f,v}] + \sum_{\substack{w \in W(v) \setminus \{v\} \\ g \in F'_{\leq}(\mathrm{pre}(w), w)}} \left([d_{v,w}^{f,g}] + \left[\overline{d_{w,v}^{g,f}}\right] \right) + \sum_{h \in \mathrm{pre}(v) \setminus F'_{\leq}(\mathrm{pre}(v), v)} \left[\overline{d_h^{f,v}}\right] \leq d^{f,v} \tag{1.2a}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$,

2. *the consistency conditions*

$$\sum_{f \in F'_{\leq}(\mathrm{pre}(v), v)} d_s^{f,v} = \sum_{vs \in E} d_{vs} + \sum_{u \in F'_{\leq}(v, \mathrm{succ}(v))} d_s^{v,u} \tag{1.2b}$$

for all $v \in U$ and $s \in S$,

$$\sum_{\substack{f \in F'_{\leq}(\mathrm{pre}(v), v) \\ g \in F'_{\leq}(\mathrm{pre}(w), w)}} d_{v,w}^{f,g} = \sum_{vw \in E'} d_{vw} + \sum_{\substack{u \in F'_{\leq}(w, \mathrm{succ}(w)) \\ \text{with } vu \notin E'_{\leq}}} d_v^{w,u} + \sum_{\substack{t \in F'_{\leq}(v, \mathrm{succ}(v)) \\ u \in F'_{\leq}(w, \mathrm{succ}(w)) \\ W(t) = W(u), t \neq u}} d_{t,u}^{v,w} \tag{1.2c}$$

for all $v, w \in U$ with $w \in W(v) \setminus \{v\}$,

$$\begin{aligned} \sum_{f \in F'_{\leq}(\mathrm{pre}(v), v)} d_h^{f,v} = \sum_{hv \in E'_{\geq}} d_{hv} &+ \sum_{\substack{u \in F'_{\leq}(v, \mathrm{succ}(v)) \\ \text{with } h \notin F'_{\leq}(\mathrm{pre}(u), u)}} d_h^{v,u} \\ &+ \sum_{\substack{u \in F'_{\leq}(h, \mathrm{succ}(v)) \\ \text{with } v \notin F'_{\leq}(\mathrm{pre}(u), u)}} \overline{d_v^{h,u}} + \sum_{\substack{t \in F'_{\leq}(v, \mathrm{succ}(v)) \\ u \in F'_{\leq}(h, \mathrm{succ}(v)) \\ W(t) = W(u), t \neq u}} \left(d_{u,t}^{h,v} + \overline{d_{t,u}^{v,h}} \right) \end{aligned} \tag{1.2d}$$

for all $v \in U$ and $h \in \mathrm{pre}(v) \setminus F'_{\leq}(\mathrm{pre}(v), v)$, and

$$\begin{aligned} d^{f,v} = \sum_{fv \in E'} d_{fv} &+ \sum_{\substack{u \in F'_{\leq}(v, \mathrm{succ}(v)) \\ \text{with } f \notin F'_{\leq}(\mathrm{pre}(u), u)}} d_f^{v,u} \\ &+ \sum_{\substack{u \in F'_{\leq}(f, \mathrm{succ}(v)) \\ \text{with } v \notin F'_{\leq}(\mathrm{pre}(u), u)}} \overline{d_v^{f,u}} + \sum_{\substack{t \in F'_{\leq}(f, \mathrm{succ}(v)) \\ u \in F'_{\leq}(v, \mathrm{succ}(v)) \\ W(t) = W(u), t \neq u}} \left(d_{t,u}^{f,v} + \overline{d_{u,t}^{v,f}} \right) \end{aligned} \tag{1.2e}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$.

Now, let an $x \in X$ and a family $(x_v)_{v\in V} \in X^V$ be given with $x_k = x$ for all $k \in K$. Define $(\hat{x}_v)_{v\in V} \in X^V$ by $\hat{x}_v := x$ for all $v \in V \setminus S$ and $\hat{x}_s := x_s$ for all $s \in V \cap S$. Then, it follows that

$$\Psi((\hat{x}_v)_{v\in V}) \leq \Psi((x_v)_{v\in V}).$$

This inequality is strict, if (1.2a) is strict for an edge $(f, v) \in E'_{\leq}$ with $x_f \neq x_v$ or if there exists an edge $e \in E_s$ such that $d_e(x_e) \neq 0$ holds.

2. Multiplicative Distance Cost Functions. The coincidence condition from the last section will now be applied to several special cases of the minisum multifacility location problem. For these special cases, we will simplify Theorem 1.2 in a considerable way. In § 3.1 and 4.1, we will compare the coincidence conditions derived here with other conditions from the literature. Using Example 3.2 from § 3, it will be shown that all the older conditions are not as flexible as the new ones. Moreover, it will become clear that all the older coincidence criteria can be derived as special cases from the ones presented here.

If several different distance cost functions d_e are used on the edges $e \in E$ of the graph G, it will usually not be very easy to check the coincidence condition (1.2), even if the partition $\mathcal{W}$ of F and a set $E_s \subseteq E$ are given. But consider the case of a fixed symmetric distance cost function d and given real numbers $\omega_e \in \mathbb{R}$ ($e \in E(K \cup U, U)$) such that

$$\omega_e d \leq d_e \qquad (e \in E(K \cup U, U))$$

holds. If we replace on each edge $e \in E(K \cup U, U)$ the function d_e by $\omega_e d$, we obtain a new objective function $\tilde{\Psi} : X^V \longrightarrow \mathbb{R}$ with $\tilde{\Psi}((y_v)_{v\in V}) \leq \Psi((y_v)_{v\in V})$ for all $(y_v)_{v\in V} \in X^V$ and $\tilde{\Psi}((\hat{x}_v)_{v\in V}) = \Psi((\hat{x}_v)_{v\in V})$. Suppose that we can proof one of the inequalities $\tilde{\Psi}((\hat{x}_v)_{v\in V}) \leq \tilde{\Psi}((x_v)_{v\in V})$ or $\tilde{\Psi}((\hat{x}_v)_{v\in V}) < \tilde{\Psi}((x_v)_{v\in V})$. Then we can immediately conclude that $\Psi((\hat{x}_v)_{v\in V}) \leq \Psi((x_v)_{v\in V})$ or $\Psi((\hat{x}_v)_{v\in V}) < \Psi((x_v)_{v\in V})$ holds. Of course, we can also replace the functions d_e by simpler ones on the edges $e \in \boldsymbol{E}(U, S)$. It may often be the case that the fulfillment of a coincidence condition with respect to the new problem is sufficient for a coincidence with respect to the original problem. For example, if all d_e ($e \in \boldsymbol{E}(U, S)$) are convex and d is a weak metric, we may replace each d_e by $\omega_e d$, where the weights ω_e are chosen such that $d_e \leq \omega_e d$ holds (it is obvious that a coincidence is now "not as easy as before"). The functions $d_s^{f,v}$ ($s \in S$, $v \in U$, $f \in F'_{\leq}(\mathrm{pre}(v), v)$) needed in (1.2a) and (1.2b) may then be constructed by using information collected while checking coincidence conditions for the function $\tilde{\Psi}$.

In what follows, let d be a symmetric weak metric and $d_e = \omega_e d$ for all $e \in E$ with certain $\omega_e \in \mathbb{R}$. As it has been already remarked on page 3, we may assume without loss of generality $d \neq 0$ and $\omega_e \neq 0$ for all $e \in E$. On the other hand, we may add any edge $e \notin E$ to the graph simply by setting $\omega_e = 0$.

Let there be given two subsets $E_{\leq}, E_{\geq} \subseteq E$ as on page 4 and a set E_s as in Theorem 1.2. Our goal is to derive simple, sufficient conditions for the system (1.2).

A natural approach uses the functions

$$d_s^{f,v} = \omega_s^{f,v} d, \quad d_{v,w}^{f,g} = \omega_{v,w}^{f,g} d, \quad d_h^{f,v} = \omega_h^{f,v} d, \quad \text{and} \quad d^{f,v} = \omega^{f,v} d$$

for the corresponding nodes $v, w, f, g, h, s \in F$. For a given partition

$$\mathcal{W} = \{K, W_1, \ldots, W_{n-1}, S\}$$

compatible with Ψ, we are now in search for weights

$$\omega_s^{f,v}, \omega_{v,w}^{f,g}, \omega_h^{f,v}, \omega^{f,v} \in \mathrm{IR}$$

such that the following conditions hold:

$$\sum_{s\in S} |\omega_s^{f,v}| + \sum_{\substack{w\in W(v)\setminus\{v\}\\ g\in F'_{\leq}(\mathrm{pre}(w),w)}} \left(|\omega_{v,w}^{f,g}| + |\omega_{w,v}^{g,f}|\right) + \sum_{h\in\mathrm{pre}(v)\setminus F'_{\leq}(\mathrm{pre}(v),v)} |\omega_h^{f,v}| \leq \omega^{f,v} \tag{2.1a}$$

for all $v \in U = F \setminus (K \cup S)$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$,

$$\sum_{f\in F'_{\leq}(\mathrm{pre}(v),v)} \omega_s^{f,v} = \sum_{vs\in E} \omega_{vs} + \sum_{u\in F'_{\leq}(v,\mathrm{succ}(v))} \omega_s^{v,u} \tag{2.1b}$$

for all $v \in U$ and $s \in S$,

$$\sum_{\substack{f\in F'_{\leq}(\mathrm{pre}(v),v)\\ g\in F'_{\leq}(\mathrm{pre}(w),w)}} \omega_{v,w}^{f,g} = \sum_{vw\in E'} \omega_{vw} + \sum_{\substack{u\in F'_{\leq}(w,\mathrm{succ}(w))\\ \text{with } vu\notin E'_{\leq}}} \omega_v^{w,u} + \sum_{\substack{t\in F'_{\leq}(v,\mathrm{succ}(v))\\ u\in F'_{\leq}(w,\mathrm{succ}(w))\\ W(t)=W(u),t\neq u}} \omega_{t,u}^{v,w} \tag{2.1c}$$

for all $v, w \in U$ with $w \in W(v) \setminus \{v\}$,

$$\begin{aligned}\sum_{f\in F'_{\leq}(\mathrm{pre}(v),v)} \omega_h^{f,v} = \sum_{hv\in E'_{\geq}} \omega_{hv} &+ \sum_{\substack{u\in F'_{\leq}(v,\mathrm{succ}(v))\\ \text{with } h\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_h^{v,u} + \sum_{\substack{u\in F'_{\leq}(h,\mathrm{succ}(v))\\ \text{with } v\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_v^{h,u}\\ &+ \sum_{\substack{t\in F'_{\leq}(v,\mathrm{succ}(v))\\ u\in F'_{\leq}(h,\mathrm{succ}(v))\\ W(t)=W(u),t\neq u}} \left(\omega_{u,t}^{h,v} + \omega_{t,u}^{v,h}\right)\end{aligned} \tag{2.1d}$$

for all $v \in U$ and $h \in \mathrm{pre}(v) \setminus F'_{\leq}(\mathrm{pre}(v), v)$, and

$$\begin{aligned}\omega^{f,v} = \sum_{fv\in E'} \omega_{fv} &+ \sum_{\substack{u\in F'_{\leq}(v,\mathrm{succ}(v))\\ \text{with } f\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_f^{v,u} + \sum_{\substack{u\in F'_{\leq}(f,\mathrm{succ}(v))\\ \text{with } v\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_v^{f,u}\\ &+ \sum_{\substack{t\in F'_{\leq}(f,\mathrm{succ}(v))\\ u\in F'_{\leq}(v,\mathrm{succ}(v))\\ W(t)=W(u),t\neq u}} \left(\omega_{t,u}^{f,v} + \omega_{u,t}^{v,f}\right)\end{aligned} \tag{2.1e}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$.

Without loss of generality, we assume that the sets W_i ($i = 1, \ldots, n-1$) are singletons. That we do not lose any information by this assumption can be seen in the following way: let $W_i \neq K$, $W_i \neq S$ be the "deepest" set with more than one element, $i := \max\{j \mid 1 \leq j \leq n-1, |W_j| \geq 2\}$. For an arbitrary $w \in W_i$, we define the set $\tilde{W} := \{w\}$ and place this set $\tilde{W}$ between W_i and W_{i+1} into the partition $\mathcal{W}$. Simultaneously, we remove w from W_i. We can now show by some arduous calculations that the conditions (2.1) are fulfilled for the new partition $\tilde{\mathcal{W}}$ just constructed if and only if they are fulfilled for the original partition $\mathcal{W}$ [19, Chapter 3]. Using this idea inductively on all sets $W_i \notin \{K, S\}$, we see that we have to consider the case of one-elemental sets W_i only. But then, condition (2.1c) is trivially true. The conditions simplified in this way are

$$\sum_{s\in S}|\omega_s^{f,v}| + \sum_{h\in \mathrm{pre}(v)\setminus F'_{\leq}(\mathrm{pre}(v),v)} |\omega_h^{f,v}| \leq \omega^{f,v} \tag{2.2a}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$,

$$\sum_{f\in F'_{\leq}(\mathrm{pre}(v),v)} \omega_s^{f,v} = \sum_{vs\in E} \omega_{vs} + \sum_{u\in F'_{\leq}(v,\mathrm{succ}(v))} \omega_s^{v,u} \tag{2.2b}$$

for all $v \in U$ and $s \in S$,

$$\sum_{f\in F'_{\leq}(\mathrm{pre}(v),v)} \omega_h^{f,v} = \sum_{hv\in E'_{\geq}} \omega_{hv} + \sum_{\substack{u\in F'_{\leq}(v,\mathrm{succ}(v))\\ \text{with } h\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_h^{v,u} + \sum_{\substack{u\in F'_{\leq}(h,\mathrm{succ}(v))\\ \text{with } v\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_v^{h,u} \tag{2.2c}$$

for all $v \in U$ and $h \in \mathrm{pre}(u) \setminus F'_{\leq}(\mathrm{pre}(v), v)$, and

$$\omega^{f,v} = \sum_{fv\in E'} \omega_{fv} + \sum_{\substack{u\in F'_{\leq}(v,\mathrm{succ}(v))\\ \text{with } f\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_f^{v,u} + \sum_{\substack{u\in F'_{\leq}(f,\mathrm{succ}(v))\\ \text{with } v\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_v^{f,u} \tag{2.2d}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$. Moreover, the derivation of Theorem 1.2 [19, 22] shows that we may replace (2.2d) by

$$\omega^{f,v} \leq \omega_{fv} + \sum_{\substack{u\in F'_{\leq}(v,\mathrm{succ}(v))\\ \text{with} f\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_f^{v,u} + \sum_{\substack{u\in F'_{\leq}(f,\mathrm{succ}(v))\\ \text{with} v\notin F'_{\leq}(\mathrm{pre}(u),u)}} \omega_v^{f,u} \tag{2.3}$$

(cp. [19, Bemerkung 2.2.6]). This will be used later on in § 4.1.

Problems with arbitrary convex and concave functions cannot be treated in the same way, since sums of convex and concave functions need to be neither convex nor concave.

3. Nonnegative Weights. Because of (2.1a) we have necessarily $\omega^{f,v} \geq 0$. Since the sets W_i $(1 \leq i \leq n-1)$ are singletons by now, we do not need to find weights $\omega_{v,w}^{f,g}$ $(v \in U,\ w \in W(v) \setminus \{w\},\ f \in F'_{\leq}(\mathrm{pre}(v), v),\ g \in F'_{\leq}(\mathrm{pre}(w), w))$. By using the approach

$$E_{\geq} := \emptyset$$

and $\omega_h^{f,v} := 0$ $(v \in U,\ f \in F'_{\leq}(\mathrm{pre}(v), v),\ h \in \mathrm{pre}(v)\setminus F'_{\leq}(\mathrm{pre}(v), v))$, we see that (2.1d) resp. (2.2c) holds, too.

For the rest of this section we are only concerned with the case $\omega_e \geq 0$ for all $e \notin E(S, S)$. The corresponding functions d_e are convex, and we will use the additional constraints

$$\omega_s^{f,v} \geq 0$$

$(v \in U,\ f \in F'_{\leq}(\mathrm{pre}(v), v),\ s \in S)$. With this, the system (2.2) simplifies to

$$\sum_{s\in S} \omega_s^{f,v} \leq \omega^{f,v} \tag{3.1a}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$,

$$\sum_{f \in F'_{\leq}(\mathrm{pre}(v),v)} \omega_s^{f,v} = \sum_{vs \in E} \omega_{vs} + \sum_{u \in F'_{\leq}(v,\mathrm{succ}(v))} \omega_s^{v,u} \tag{3.1b}$$

for all $v \in U$ and $s \in S$, and

$$\omega^{f,v} = \omega_{fv} \tag{3.1c}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$.

Obviously, condition (3.1c) can be dropped after replacing the right-hand side in (3.1a) by ω_{fv}. Thus, the coincidence condition can be written as

$$\sum_{s \in S} \omega_s^{f,v} \leq \omega_{fv} \tag{3.2a}$$

for all $v \in U$ and $f \in F'_{\leq}(\mathrm{pre}(v), v)$ and

$$\sum_{f \in F'_{\leq}(\mathrm{pre}(v),v)} \omega_s^{f,v} = \sum_{vs \in E} \omega_{vs} + \sum_{u \in F'_{\leq}(v,\mathrm{succ}(v))} \omega_s^{v,u} \tag{3.2b}$$

for all $v \in U$ and $s \in S$. Note that we would have arrived at exactly these two relations if we would have replaced (2.2d) by (2.3).

As one may suspect, we lose some of the generality that we had beforehand due to the additional constraints $\omega_s^{f,v} \geq 0$, at least if the partition $\mathcal{W}$ is fixed:

EXAMPLE 3.1. Let $a \in S$, $u \in W_{n-1}$, $w \in W_{n-2}$, $v \in W_{n-3}$, $f, g \in U_{n-4}$, $\omega_{ua} = \omega_{vu} = \omega_{fv} = 1$, and $\omega_{va} = \omega_{wu} = \omega_{gw} = 2$. All other weights in the subgraph induced by a, f, g, u, v, and w are 0. Furthermore, let $F'_{\leq}(\mathrm{pre}(u), u) = \{v, w\}$. With $\omega_a^{w,u} := 2$ and $\omega_a^{v,u} := -1$ we see that we have attraction from u to v and to w. In the next step we have $\omega_a^{g,w} = 2$ and therefore attraction from w to g and in the step afterwards $\omega_a^{f,v} = 1$ and therefore attraction from v to f. But with the additional constraints $\omega_a^{v,u}, \omega_a^{w,u} \geq 0$ it is not possible to choose $\omega_a^{f,v}$ in such a way that (3.2) holds. However, after changing the partition such that $v \in W_{n-1}$, $u \in W_{n-2}$, and $w \in W_{n-3}$ we can work with nonnegative weights and get the desired result. We conjecture that this strategy of changing the partition leads to the desired result rather often.

The next example has already been used in [22] and will be used later on again to show that the coincidence conditions derived in other works do not find all coincidences which can be detected by the conditions derived here.

EXAMPLE 3.2. Let $A = \{a, b, c, d\}$, $V = \{u, v, w\}$,

$$E = \{(u, a), (u, b), (u, w), (v, a), (v, c), (v, w), (w, d)\},$$

$\omega_{ua} = \omega_{va} = 7$, $\omega_{ub} = \omega_{vc} = 2$, $\omega_{uw} = \omega_{vw} = 6$, and $\omega_{wd} = 8$. Furthermore, let $\Psi((x_u, x_v, x_w)) = \sum_{e \in E} \omega_e d(x_e)$ with a symmetric weak metric d.

Define $n = 4$, $K = \{a\}$, $S = \{b, c, d\}$, $W_1 = \{u\}$, $W_2 = \{v\}$, $W_3 = \{w\}$, and $E_s = \emptyset$. Because of the symmetry of d we may turn around edges as we wish. Since K contains only one element, we have trivially $x_k = x$ for all $k \in K$ and all families $(x_v)_{v \in V}$. Because (3.2b) has to hold for the weights $\omega_d^{u,w}$ and $\omega_d^{v,w}$, we have to choose these weights in such a way that the equality

$$\omega_d^{u,w} + \omega_d^{v,w} = \omega_{wd} = 8$$

holds. Likewise, because (3.2a) has to hold, we have to choose them in such a way that

$$\omega_d^{u,w} \le \omega_{uw} = 6$$

and

$$\omega_d^{v,w} \le \omega_{vw} = 6$$

holds. (Here, we may also consider (3.2a) for $f := a$. Because of $\omega_{aw} = 0$ we have then to choose $\omega_d^{a,w} := 0$.) As an example, we choose $\omega_d^{u,w} = 4$ and $\omega_d^{v,w} = 4$. With this, (3.2) holds with respect to $w \in U$ and the set $F'_{\le}(\text{pre}(w), w) = \{u, v\}$. Our choice leads subsequently to $\omega_d^{a,u} = 4$ and $\omega_d^{a,v} = 4$, and (3.2) holds for all nodes necessary. It follows that

$$\Psi((\hat{x}_v)_{v\in V}) \le \Psi((x_v)_{v\in V})$$

holds for all $(x_v)_{v\in V} \in X^V$. This inequality is strict if $x_w \neq x_u$ or $x_w \neq x_v$ or $x_v \neq x_a$ or $x_u \neq x_a$. An analogous derivation can be achieved by setting $n = 3$, $W_1 := \{u, v\}$ and $W_2 := \{w\}$.

The same result can be shown to hold if the edge (u, v) with the weight $\omega_{uv} = -1/2$ is added to the graph. In this case we have to use $E_{\ge} := \{(u, v)\}$, $\omega_u^{a,v} = -1/2$, and $\omega^{a,u} = 13/2$.

Consider now the case in which we add not only the edge (u, v) with $\omega_{uv} = -1/2$ to the graph, but also the edge (a, w) with the weight $\omega_{aw} = -1/2$. With this, we define $E_{\ge} := \{(a, w), (u, v)\}$. Using the same partition as above, we choose $\omega_d^{u,w} = 4$ and $\omega_d^{v,w} = 4$. But because of (2.2c), we have to choose weights $\omega_a^{u,w}$ and $\omega_a^{v,w}$ with

$$\omega_a^{u,w} + \omega_a^{v,w} = \omega_{aw} = -1/2,$$

such that with the weights $\omega_d^{u,w}$ and $\omega_d^{v,w}$ as above we still have (2.2d). This can be done, for example, by choosing $\omega_a^{u,w} := \omega_a^{v,w} := -1/4$. With this, the collapse as above is allowed. The equations $\omega^{a,v} = 27/4$ and $\omega^{a,u} = 25/4$ follow immediately. We still have strict attraction together with the final result.

3.1. Former Results: Attraction Trees and Attraction Set Trees. The most general coincidence conditions for minisum multifacility location problems with mixed weak metrics and nonnegative weights known so far can be found in [18]. There, the concept of *attraction set trees* was introduced. We consider this concept here for the special case $d_e = \omega_e d$, d a symmetric weak metric. Let $\mathcal{U} = \{R, U_1, \ldots, U_m, \mathcal{O}\}$ be a partition of F with $|R \cap A| \le 1$ and $\bigcup_{i=1}^m U_i \subseteq V$ and let $T_{\mathcal{U}}$ be a tree with node set $\{R, U_1, \ldots, U_m\}$, root R, and edge set Γ. We denote the father node of a node U of $T_{\mathcal{U}}$ by father(U). Such a tree is called attraction set tree, if for each node U of the tree, $U \neq R$, the sum of weights on edges leading to the father of U is greater than or equal to the the sum of weights on edges leading to children of U or to nodes in $\mathcal{O}$:

$$\sum_{e\in \boldsymbol{E}(\text{father}(U),U)} \omega_e \ge \sum_{(U,W)\in\Gamma}\ \sum_{e\in \boldsymbol{E}(U,W)} \omega_e + \sum_{(W,U)\in\Gamma}\ \sum_{e\in \boldsymbol{E}(W,U)} \omega_e + \sum_{e\in \boldsymbol{E}(U,\mathcal{O})} \omega_e.$$

We have a coincidence result similar to Theorem 1.2, i. e. if $T_{\mathcal{U}}$ is an attraction set tree, then $\Psi((\hat{x}_v)_{v\in V}) \le \Psi((x_v)_{v\in V})$ holds, where $(\hat{x}_v)_{v\in V}$ is obtained from $(x_v)_{v\in V}$ by identifying the nodes in U_i, $i = 1, \ldots, m$, with the ones in R (see [18]). In [18] it has also been shown that the *attraction trees* introduced by Lefebvre, Michelot, and Plastria [28, 29] are special cases of attraction set trees, since each coincidence detected by an attraction tree is also detected by some attraction set tree, but not every coincidence detected by an attraction set tree can be detected by an attraction tree.

Consider Example 3.2 without the negative weighted edges. If one tries to construct an attraction tree or an attraction set tree, it is immediately clear that none of the nodes b, c, or d may serve as the root node. Likewise, there exists no attraction (set) tree with root u, v, or w. Each attraction (set) tree with root a which attracts u to a has to suppress the edge (u, w). This means that the edge (v, w) and therefore the edge (v, a), too, has to be contained in the tree. But then we do not have attraction between a and v. Each attraction tree with root a that suppresses the edge (u, a) has to attract u to w. Therefore, the edge (u, w) has to be in the tree. But then w has to be attracted to v. Because of $\omega_{vw} = 6$ and $\omega_{wd} = 8$ we have a contradiction. As a consequence, there is no attraction tree and no attraction set tree in Example 3.2.

We will now show that the existence of an attraction set tree implies the existence of a partition compatible with Ψ and corresponding weights feasible to the system (3.2).

Let $\mathcal{U} = \{R, U_1, \ldots, U_m, \mathcal{O}\}$ be a partition of F and $T_{\mathcal{U}}$ be an attraction set tree as above. Let $n-1$ be the height of $T_{\mathcal{U}}$. (The root R has height 0, each son of the root has height 1, etc.) We have to construct a partition $\mathcal{W} = \{K, W_1, \ldots, W_{n-1}, S\}$. To this end, define W_i $(0 \leq i \leq n-1)$ as the set of all nodes $w \in \bigcup_{j=1}^{m} U_j$ which lie in those sets U_j that have height i in the tree $T_{\mathcal{U}}$. Furthermore, set $K := R$ and $S := \mathcal{O}$. As usual, we assume without loss of generality $E(W_i, W_j) = \boldsymbol{E}(W_i, W_j)$ for $i < j$. The set E_s is defined as the set of all edges suppressed in $T_{\mathcal{U}}$. For $i = n-1, \ldots, 1$ and $s \in S$ we will define the weights $\omega_s^{f,v}$ $(v \in W_i, f \in \mathrm{pre}(v))$ inductively as follows.

1. $i = n-1$: Because of the attraction in $T_{\mathcal{U}}$ we have for all $v \in W_{n-1}$ that

$$\sum_{s \in S} \omega_{vs} \leq \sum_{e \in E'(W_{n-2}, v)} \omega_e. \tag{3.3}$$

We define for each $v \in W_{n-1}$ a transportation problem as follows: the nodes $s \in S$ are the producers with capacity ω_{vs}, the edges $e \in E'(W_{n-2}, v)$ are the consumers with demand ω_e. If we do not have equality in (3.3), we introduce, as usual in such a situation, a fictitious producer $\hat{s}$ with capacity

$$\hat{\omega} := \sum_{e \in E'(W_{n-2}, v)} \omega_e - \sum_{s \in S} \omega_{vs} > 0.$$

For an arbitrary cost matrix we have a nonempty set of feasible points for the corresponding transportation problem. Let us fix an arbitrary feasible point and let $\omega_s^{f,v}$ be the demand of (f, v) served from s at this point. Because all demands are satisfied, we have (3.2a). Since $\mathrm{succ}(v) = \emptyset$ and all capacities are completely used we have (3.2b).

2. $i \longrightarrow i-1$, $i \geq 2$: By the induction hypothesis we have constructed weights $\omega_s^{f,v}$ $(v \in \bigcup_{j=i}^{n-1} W_j,\ f \in F'_{\leq}(\mathrm{pre}(v), v),\ s \in S)$, for which (3.2) holds. Let $v \in W_{i-1}$. Because of the attraction condition and the definition of suppressed edges in $T_{\mathcal{U}}$ we have

$$\sum_{e \in E'(v, S)} \omega_e + \sum_{e \in E'(v, W_i)} \omega_e \leq \sum_{e \in E'(W_{i-2}, v)} \omega_e.$$

For each $u \in F'(v, W_i)$ there exists weights $\omega_s^{v,u}$ such that

$$\sum_{s \in S} \omega_s^{v,u} \leq \omega_{vu}$$

holds. This can be deduced from (3.2a) and the induction hypothesis. Therefore,

$$\sum_{s \in S} \left(\omega_{vs} + \sum_{u \in F'_{\leq}(v, W_i)} \omega_s^{v,u} \right) \leq \sum_{e \in E'(W_{i-2}, v)} \omega_e$$

follows. Analogously to the start of the induction we can construct weights $\omega_s^{f,v}$ ($v \in W_{i-1}$, $f \in F'_{\leq}(\mathrm{pre}(v), v) = F'(W_{i-2}, v)$, $s \in S$) such that

$$\sum_{s \in S} \omega_s^{f,v} \leq \omega_{fv}$$

holds for all $v \in W_{i-1}$, $f \in F'_{\leq}(W_{i-2}, v)$, and such that

$$\sum_{f \in F'_{\leq}(W_{i-2}, v)} \omega_s^{f,v} = \omega_{vs} + \sum_{u \in F'_{\leq}(v, W_i)} \omega_s^{v,u}$$

holds for all $v \in W_{i-1}$, $s \in S$. This completes the proof.

A discussion about other, not as general criteria (like the ones in [30, 11]) can be found in [19].

3.2. Flows: An Efficient Checking Algorithm. If $\mathcal{W}$ and E' are given, the determination of weights which fulfill the system (2.1) is a simple feasibility problem in linear programming. Especially, a Phase I of the simplex method will be sufficient to check for the existence of these weights. Moreover, specially geared interior-point algorithms [42, 41] will lead to methods with polynomial run-time. However, up to now it is not clear how the combinatorial component of choosing the correct partition once the kernel K and the shell S are specified can be incorporated into the problem. This question will be answered in this subsection. At first, we will only consider problems with nonnegative weights. The results derived here will be generalized to the case of arbitrary weights later on (cp. § 4).

We will make use of our general assumption $\omega_e \geq 0$ for all $e \in \boldsymbol{E}(F \setminus (K \cup S), F)$. Let there be given a partition $\mathcal{W} = \{K, W_1, \ldots, W_{n-1}, S\}$ compatible with Ψ with kernel K, shell S, and a set $E_s \supseteq \bigcup_{i=0}^{n-1} E(W_i, W_i)$. As usual, we set $E' = E \setminus E_s$. We define a transportation network $\mathcal{T}(\mathcal{W}, E_s) = (F, \hat{E}(\mathcal{W}, E_s))$ as follows: $\mathcal{T}(\mathcal{W}, E_s)$ has the set of sources K, the set of sinks S and with $U := F \setminus (K \cup S)$ the set of edges

$$\hat{E}(\mathcal{W}, E_s) := E'(K, U) \cup E'(U, U) \cup (U \times S).$$

The interval of capacity of an edge $e \in \hat{E}(\mathcal{W}, E_s)$ is $[0, \omega_e]$ for $e \in E'$ and $\{0\}$ else. For a fixed partition $\mathcal{W}$ and fixed set E_s we abbreviate $\hat{E} := \hat{E}(\mathcal{W}, E_s)$.

THEOREM 3.1. *There exist weights $\omega_s^{f,v} \geq 0$ ($v \in U$, $f \in F'_{\leq}(\mathrm{pre}(v), v)$, $s \in S$) which fulfill (3.2) if and only if there exists a (K, S)-flow $(\varphi_e)_{e \in \hat{E}}$ on $\mathcal{T}(\mathcal{W}, E_s)$ saturating all edges in $U \times S$.*

Proof. First, let there be given weights $\omega_s^{f,v} \geq 0$ ($s \in S$, $v \in U$, $f \in F'_{\leq}(\mathrm{pre}(v), v)$) such that (3.2) holds. For $e = (k, v) \in E'(K, U)$ set

$$\varphi_e := \sum_{s \in S} \omega_s^{k,v}$$

and, analogously,

$$\varphi_e := \sum_{s \in S} \omega_s^{f,g}$$

for all $e = (f,g) \in E'(U,U)$. For $e = (v,s) \in U \times S$ define $\varphi_e := \omega_e$ if $e \in E'$ and $\varphi_e := 0$ else. It is clear that $(\varphi_e)_{e\in\hat{E}}$ saturates all edges in $U \times S$. Let $v \in U$. Then it follows that $\hat{E}(F,v) = E'(\mathrm{pre}(v),v)$ and $\hat{E}(v,F) = E'(v,\mathrm{succ}(v)) \cup (\{v\} \times S)$. With (3.2b), we see that $(\varphi_e)_{e\in\hat{E}}$ is conservative. Because of (3.2a) it is clear that each edge-flow φ_e is feasible.

Second, let there be given a conservative, feasible (K,S)-flow $(\varphi_e)_{e\in\hat{E}}$ which saturates all edges in $U\times S$ (here, the set E_s is fixed). Since we have $E_s \supseteq \bigcup_{i=0}^{n-1} E(W_i, W_i)$, the edge set $\hat{E}$ contains no cycles. For $(f,g) \in E'(U \cup K, U)$ define the weight $\omega_s^{f,g}$ as the part of the total flow that flows via (f,g) to $s \in S$. The feasibility of the flow implies (3.2a). The conservativeness implies (3.2b). The nonnegativity is trivial. □

Remark 3.2. Using the Theorem of Gale (cp., e. g., Berge and Ghuila-Houri [6, p. 192]) we see: there exist nonnegative weights which fulfill the conditions (3.2) if and only if for all subsets $\tilde{U} \subseteq U$ we have that

$$\sum_{e\in E'(F\setminus S,\tilde{U})} \omega_e \geq \sum_{e\in E'(\tilde{U},S)} \omega_e$$

holds.

For the maximal flow problem there exist efficient algorithms (cp., e. g., [1] for an overview). Moreover, we can test in time $O(|\hat{E}|) = O(|E|)$ if a given (K,S)-flow saturates the sink edges. The question remains if there exists an efficient algorithm to compute a partition $\mathcal{W} = \{K, W_1, \ldots, W_{n-1}, S\}$ compatible with Ψ and a flow satisfying the conditions above if only K and S are given. We will see that such an efficient algorithm exists.

Let $K, S \subseteq F$ be fixed with $|K \cap A| \leq 1$, $K \cap S = \emptyset$ and $U := F \setminus (K \cup S) \subseteq V$. Without loss of generality, let $E(K,U) = \boldsymbol{E}(K,U)$ and $E(U,S) = \boldsymbol{E}(U,S)$. We define a transportation network $\mathcal{T}_0 = (F, E_0)$ as follows: $\mathcal{T}_0$ shall have the node set F, the edge set

$$E_0 := E(K,U) \cup E(U,U) \cup E(U,S),$$

the set of sources K, and the set of sinks S. The edges $e \in E_0$ shall have the intervals of capacity $[0, \omega_e]$ if $e \in E(K,U) \cup E(U,S)$ and $[-\omega_e, \omega_e]$ if $e \in E(U,U)$.

Let $(\varphi_e)_{e\in E_0}$ be a feasible (K,S)-flow on $\mathcal{T}_0$. First of all, we will "turn around" all edges $e = (f,g) \in E(U,U)$ with $\varphi_e < 0$: in the set E_0, the edge e is replaced by $\overline{e} = (g,f)$ and we define $\varphi_{\overline{e}} := -\varphi_e$. By doing this, the flow $(\varphi_e)_{e\in E_0}$ remains a conservative, feasible (K,S)-flow. But now we have $\varphi_e \geq 0$ for all $e \in E_0$. Let $Z \subseteq E(U,U)$ be a directed cycle with $\varphi_e > 0$ for all $e \in Z$. (i. e. it is $Z = \{(f_1,g_1), \ldots, (f_m,g_m)\}$ with $f_j = g_{j-1}$ $(j = 2, \ldots, m)$ and $g_m = f_1$.) We can "lower" the flow $(\varphi_e)_{e\in E_0}$ on Z by $\delta_Z := \min_{e\in Z} \varphi_e > 0$ without destroying conservation or feasibility. Possible saturation properties on $E(U,S)$ are not disturbed. After treating all cycles Z with $\delta_Z > 0$ in this way (there is only a finite number of them), we define

$$E' := \{e \in E(U,U) \mid \varphi_e > 0\}$$

and $E_s := E(U,U) \setminus E'$. The set E' is the edge set of a directed graph with node set U which has no cycles. This means that we may topologically sort the node set U with respect to the partial ordering

$$f \le g :\Longleftrightarrow \begin{cases} \text{it is } f = g \text{ or there exists in } E' \text{ a} \\ \text{directed path from } f \text{ to } g. \end{cases}$$

Let $f_1, \ldots, f_{|U|}$ be such a topological ordering of U, such that $f_i \le f_j$ implies $i \le j$. Then, define

$$W_i := \{f_i\}$$

$(i = 1, \ldots, |U|)$.

We have just shown that for each (K,S)-flow on $\mathcal{T}_0$ saturating all edges in $E(U,S)$ we can construct a partition $\mathcal{W} := \{K, \{f_1\}, \ldots, \{f_{|U|}\}, S\}$ and a (K,S)-flow on the transportation network $\mathcal{T}_0$ that saturates all edges in $E(U,S)$. This (K,S)-flow in turn induces a set E' and a set $E_s = E(U,U) \setminus E'$ such that there exists a (K,S)-flow on $\mathcal{T}(\mathcal{W}, E_s)$ saturating all edges in $U \times S$.

On the other hand, for a given partition $\mathcal{W}$ and a given set $E_s \subseteq E(U,U)$, it is easy to see that each (K,S)-flow on $\mathcal{T}(\mathcal{W}, E_s)$ which saturates all edges in $E(U,S)$ induces a (K,S)-flow on $\mathcal{T}_0$ saturating all edges in $E(U,S)$.

To check the saturation of the edges, there is no need to modify the flow or sort the nodes topologically. The existence of the saturating flow is already ensured by the construction described above.

The derivation above shows that we have the following theorem.

THEOREM 3.3. *Let K and S be given with $|K \cap A| \le 1$, $K \cap S = \emptyset$ and $U := F \setminus (K \cup S) \subseteq V$. Then, there exists a partition of U and corresponding nonnegative weights which fulfill system* (3.2) *if and only if there exists a flow on the transportation network $\mathcal{T}_0$ saturating all the edges in $E(U,S)$.*

THEOREM 3.4. *Let K and S be given with $|K \cap A| \le 1$, $K \cap S = \emptyset$ and $U := F \setminus (K \cup S) \subseteq V$. Then, we can check in polynomial time if there exists a partition of U and corresponding nonnegative weights which fulfill* (3.2).

Proof. For K and S given, we can rebuild the transportation network in polynomial time to an equivalent network containing exactly one sink and one source. A more detailed description of the process depends on the data structure in use and the representation of the graph G. We can now compute in polynomial time a maximal (K,S)-flow on $\mathcal{T}_0$. For example, this can be done in time $O(|U|^2\sqrt{|\boldsymbol{E}(U,F)|})$ [1]. If a maximal (K,S)-flow does not saturate the sink edges, there does not exist a feasible flow which saturates the sink edges. If, as it is usually done so in flow problems, all nodes in S are collapsed to one node in time $O(|\boldsymbol{E}(S,F)|)$ and all nodes in K are collapsed to one node in time $O(|\boldsymbol{E}(K,F)|)$ and a backflow edge is added, the saturation test can be performed in constant time. Otherwise, we need $O(|\boldsymbol{E}(U,S)|)$ steps for it. $\square$

Note that the checking algorithm outlined above is constructive in the sense that the nonnegative weights and the partition of U are constructed, if they exist.

In a practical setting, we will only be given a set $\tilde{U} \subset F$ such that $|\tilde{U} \cap A| \le 1$ holds, and this set $\tilde{U}$ will be only a candidate for a coincidence. A set K has still to be constructed. If $\tilde{U} \cap A = \{a\} \neq \emptyset$ we have to check only the case $K = \{a\}$, $U = \tilde{U} \setminus K$. Otherwise, we will use as K successively the one-elemental subsets of $\tilde{U}$. Therefore, our checking algorithms will usually have another factor of $|\tilde{U}|$ in their run times. More details about an actual implementation can be found in [19] and in § 5.

REMARK 3.5. The derivation above shows that the set E_s used in Theorem 1.2 can be defined as $E_s := \{e \in E(K \cup U, K \cup U) \mid \varphi_e = 0,\, \omega_e > 0\}$, where $(\varphi_e)_{e \in E_0}$ is a constructed maximal flow. Then, the results derived for this set hold. Moreover, (3.2a) and thus (2.1a) as well as (1.2a) are strict for (f, v), if we have $|\varphi_{fv}| < \omega_{fv}$ for the constructed flow. In this way, we can distinguish between necessary coincidences and coincidences which do not occur necessarily in a solution of the facility location problem.

If an attraction structure with a set $U \subseteq V$ and a kernel K is known and K is a singleton, we will usually simplify the problem by identifying all nodes in U with the node in K. The next two remarks deal with the consequences of such a collapse on future coincidence tests.

REMARK 3.6. Let $K_1, K_2, U_1, U_2 \subseteq F$ be given sets with $U_i \subseteq V$, K_i singletons, $S_i := F \setminus (K_i \cup U_i)$ $(i = 1, 2)$, $K_2 \subseteq K_1 \cup U_1$, and $U_2 \subseteq U_1$. If the nodes in U_2 are identified with the node in K_2, the graph G is replaced by a graph $\hat{G} = (\hat{F}, \hat{E})$, while U_1 is replaced by a set $\hat{U}_1$. Each (K_1, S_1)-flow on the transportation network with node set F and edge set $E_0 = E(K_1, U_1) \cup E(U_1, U_1) \cup E(U_1, S_1)$ saturating all sink edges induces a (K_1, S_1)-flow on the transportation network with node set $\hat{F}$ and edge set $\hat{E}_0 = \hat{E}(K_1, \hat{U}_1) \cup \hat{E}(\hat{U}_1, \hat{U}_1) \cup \hat{E}(\hat{U}_1, S_1)$, saturating all sink edges. This means that collapsing an attraction structure inside of another attraction structure does not destroy the outer structure.

We can also examine the behaviour of two "competing" attraction structures which attract a subset of F to different kernels.

REMARK 3.7. Let $K_1, K_2, U_1, U_2 \subseteq F$ be given sets such that $|K_i \cap A| \leq 1$, $U_i \cap K_i = \emptyset$, $U_i \subseteq V$ as well as $K_1 \cap K_2 = \emptyset$, $K_i \cap U_j = \emptyset$ $(i \neq j)$, and $\tilde{U} := U_1 \cap U_2 \neq \emptyset$. Set $S_i := F \setminus (K_i \cup U_i)$ $(i = 1, 2)$ and $\tilde{S} := F \setminus (U_1 \cup U_2 \cup K_1 \cup K_2)$. As above, we assume without loss of generality $\boldsymbol{E}(U_i, \tilde{S}) = E(U_i, \tilde{S})$ and $\boldsymbol{E}(K_i, U_i) = E(K_i, U_i)$ $(i = 1, 2)$. For $i = 1, 2$ there shall exist a (K_i, S_i)-flow $(\varphi_e^{(i)})_{e \in E_0^{(i)}}$ on the transportation network with node set F and edge set $E_0^{(i)} = E(K_i, U_i) \cup E(U_i, U_i) \cup (U_i \times S_i)$ saturating all sink edges. (Note that the two transportation networks induce different intervals of capacity for the edges $e \in E$. For $e \in E(\tilde{U}, U_1 \setminus \tilde{U})$ we have $[-\omega_e, \omega_e]$ as the interval of capacity for the first transportation network, but $[0, \omega_e]$ as the interval of capacity for the second network.) Now it is

$$
\begin{aligned}
\sum_{e \in \boldsymbol{E}(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} \omega_e &\geq \sum_{e \in \boldsymbol{E}(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} |\varphi_e^{(1)}| \\
&\geq \sum_{e \in E(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} \varphi_e^{(1)} - \sum_{e \in E(\tilde{U}, K_1 \cup (U_1 \setminus \tilde{U}))} \varphi_e^{(1)} \\
&= \sum_{e \in E(\tilde{U}, \tilde{S})} \varphi_e^{(1)} + \sum_{e \in (\tilde{U} \times (K_2 \cup (U_2 \setminus \tilde{U})))} \varphi_e^{(1)} \\
&= \sum_{e \in E(\tilde{U}, \tilde{S})} \omega_e + \sum_{e \in \boldsymbol{E}(\tilde{U}, K_2 \cup (U_2 \setminus \tilde{U}))} \omega_e \\
&\geq \sum_{e \in E(\tilde{U}, \tilde{S})} \omega_e + \sum_{e \in \boldsymbol{E}(\tilde{U}, K_2 \cup (U_2 \setminus \tilde{U}))} |\varphi_e^{(2)}|.
\end{aligned}
$$

Likewise, it follows that

$$
\sum_{e \in \boldsymbol{E}(\tilde{U}, K_2 \cup (U_2 \setminus \tilde{U})))} |\varphi_e^{(2)}| \geq \sum_{e \in E(\tilde{U}, \tilde{S})} \omega_e + \sum_{e \in \boldsymbol{E}(\tilde{U}, K_1 \cup (U_1 \setminus \tilde{U}))} \omega_e,
$$

which means that

$$\sum_{e \in \boldsymbol{E}(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} \omega_e \geq 2 \sum_{e \in E(\tilde{U}, \tilde{S})} \omega_e + \sum_{e \in \boldsymbol{E}(\tilde{U}, K_1 \cup (U_1 \setminus \tilde{U}))} \omega_e$$

and therefore

$$\sum_{e \in E(\tilde{U}, \tilde{S})} \omega_e = 0$$

holds. Consequently, we have $\omega_e = 0$ for all $e \in E(\tilde{U}, \tilde{S})$. The chain of inequalities above simplifies to

$$\sum_{e \in \boldsymbol{E}(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} \omega_e \geq \sum_{e \in \boldsymbol{E}(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} \omega_e,$$

which means that we have equality in each step, especially at the first two inequalities. This results in

$$\sum_{e \in E(K_1 \cup (U_1 \setminus \tilde{U}), \tilde{U})} (\omega_e - \varphi_e^{(1)}) + \sum_{e \in E(\tilde{U}, K_1 \cup (U_1 \setminus \tilde{U}))} (\omega_e + \varphi_e^{(1)}) = 0,$$

and after turning around all edges with $\varphi_e^{(1)} < 0$ we see that the flow $(\varphi_e^{(1)})_{e \in E_0^{(1)}}$ saturates all edges in $E(\tilde{U}, K_1 \cup (U_1 \setminus \tilde{U}))$. Likewise, after turning around those edges e with $\varphi_e^{(2)} < 0$, we see that the flow $(\varphi_e^{(2)})_{e \in E_0^{(2)}}$ saturates all edges in $E(\tilde{U}, K_2 \cup (U_2 \setminus \tilde{U}))$. By this, the set $\tilde{U}$ is completely determined. Furthermore, from each (K_i, S_i)-flow $(\varphi_e^{(i)})_{e \in E_0^{(i)}}$ we can construct by trivial modifications a $(K_i, S_i \cup \tilde{U})$-flow on the transportation network with edge set

$$E(K_i, U_i \setminus \tilde{U}) \cup E(U_i \setminus \tilde{U}, U_i \setminus \tilde{U}) \cup \Big((U_i \setminus \tilde{U}) \times (S_i \cup \tilde{U})\Big)$$

which saturates all sink edges. This means that the set $\tilde{U}$ can be "deleted" from each attraction structure. If only one attraction structure and a corresponding nonnegative flow are given, the only candidates for such a restructuring process are exactly those sets $\tilde{U}$ for which the flow saturates all edges in $\boldsymbol{E}(\tilde{U}, F)$. A collapse of an attraction structure is no hindrance in the search for other attraction structures. These other attraction structures can still be found by the tools presented here.

In light of the two remarks above we should opt for a collapse of an attraction structure as soon as it has been found. To spare computation time, we should collapse nodes even if we have hope of finding a bigger attraction structure later on. The two last remarks show that we will not destroy any information needed later on by collapsing.

4. Arbitrary Weights. In this section, we will generalize the efficiently checkable reduction conditions derived in the last subsection such that they can handle problems with positive and negative weights. So, we are in search for sufficient conditions for the relations (2.1) that are easy to check. We will start by using the approach

$$E_{\geq} := \{e \in E \mid 0 > \omega_e\}.$$

If we now replace all weights $\omega_e < 0$, $e \in E(U,S)$, by their absolute values, we get a new problem in which we may introduce the additional constraints $\omega_s^{f,v} \geq 0$ ($s \in S$, $v \in U$, $f \in F'_{\leq}(\mathrm{pre}(v), v)$). This is exactly what we have done in the nonnegative case. If there exist weights for this new problem which fulfill (2.1), then there exist weights for the original problem which fulfill these conditions. The following example shows that the converse is not true.

EXAMPLE 4.1. Let $A = \{a, b\}$, $V = \{v, w\}$, $E = \{(a,v),(v,b),(v,w),(w,b)\}$, $\omega_{va} = \omega_{vw} = 1$, $\omega_{vb} = 2$, and $\omega_{wb} = -1$. If we choose $K = \{a\}$, $S = \{b\}$, $W_1 = \{v\}$, and $W_2 = \{w\}$, we can first identify w with v and then v with a. If we use the absolute values of weights on the edges in $E(U,S) = \{(v,b),(u,b)\}$, our first step of identifying w with v is still feasible. However, the resulting weight $\omega_{vb} = 3$ means that (2.1a) cannot hold for the node v.

The question remains if coincidence conditions for problems with positive and negative weights can also be checked efficiently by constructing a maximal (K,S)-flow. In what follows, we will give an affirmative answer to this question.

We will make use of our general assumption $|W_i| = 1$, $i = 1, \ldots, |U|$. That we may do so without loss of generality has been discussed on page 6. We can therefore consider the conditions (2.2).

Let $W_\ell \neq S$ be the "deepest" set in $\mathcal{W}$ which is connected with a node in U by a negative weighted edge: $\ell := \max\{i < n \mid \exists\, h \in U : \omega_{hv} < 0 \text{ with } W(v) = W_i\}$. Let $W_\ell = \{v\}$. Furthermore, let $h \in U$ be the "deepest" node which is connected with W_ℓ by a negative weighted edge: $W(h) = W_j$ and $j := \max\{i < \ell \mid W_i = \{h\} \text{ and } \omega_{hv} < 0\}$. We add two nodes s_h and s_v to S and replace the edge (h,v) by the edges (h, s_h) and (v, s_v), using the weights $\omega_{(h,s_h)} = \omega_{(v,s_v)} = \omega_{hv}$. The result is a modified graph $\tilde{G}$ and a modified partition $\tilde{\mathcal{W}}$. We will now show the following statement: if there exist weights which fulfill the conditions (2.2) with respect to $\tilde{G}$ and $\tilde{\mathcal{W}}$, then there exist weights which fulfill the conditions (2.2) with respect to G and $\mathcal{W}$.

We perform the construction from the proof of Theorem 1.2 (cp. [19, 22]) for the graph G, assuming that this construction is possible for $\tilde{G}$. For the steps $i = n-1, \ldots, \ell+1$ there will be no change compared to the construction for $\tilde{G}$. For $i = \ell$ we define $\omega_h^{f,v} := \omega_{s_v}^{f,v}$ ($f \in F'_{\leq}(\mathrm{pre}(v), v)$) in order to construct the graph G_ℓ in an analogous way to $\tilde{G}_\ell$. Then, the construction of the G_i, $i = \ell-1, \ldots, j+1$, can be done in an analogous way to the construction of the $\tilde{G}_i$. Since (2.2) holds for $\tilde{G}$, the graph $\tilde{G}_j$ can be constructed. But the conditions necessary to construct G_j are even easier to fulfill, since terms from (2.2b) vanish in (2.2a) if they stem from the node $s = s_v$. Exactly those weights $\omega_{s_v}^{h,f} < 0$ for which there is a path in G from f to v whose edges have nonnegative weights only do not occur. Likewise, for $i = j-1, \ldots, 1$ we can construct G_i in the same way as $\tilde{G}_i$. As a consequence, (2.2) holds for G and $\mathcal{W}$.

If all edges in $E(U,U)$ with negative weights are treated in this way, we have to work with a graph whose edges in $E(U,U)$ have nonnegative weights only. If we use the absolute values of the weights on the edges in $E(U,S)$, we can easily check the flow condition (Theorem 3.3 and 3.4) *without* knowledge of the partition $\mathcal{W}$. The derivation above shows that this is a sufficient criterion for a coincidence on $K \cup U$. Since we have only added nodes, redefined some edges and changed the sign on some numbers, we still have polynomial time complexity for our checking algorithm. Moreover, Remark 3.5 carries over to the case of arbitrary weights.

4.1. Former Results. To the author's knowledge, the only coincidence conditions for multifacility location problems with positive and negative weights can be found in [21]. We will now examine if these conditions fit into the scheme presented in the last two sections. For a detailed statement of the former results we refer to [21].

Let $\mathcal{U} := \{R, W_1, \ldots, W_{n-1}, \mathcal{O}\}$ be a partition compatible with Ψ and $T_{\mathcal{U}}$ be an extended attraction set tree on G. Then, for all i, $1 \le i \le n-1$, there exists a partition $E(W_i, W_i) = T_i \cup S_i$ such that $S_i \cap T_i = \emptyset$ and $\omega_e \ge 0$ for $e \in S_i$. It is allowed that one or even both sets are empty. As a preliminary definition, define E_s as the set of all edges suppressed in $T_{\mathcal{U}}$, $K := R$ and $S := \mathcal{O}$. We will now add more edges to E_s. For $i \in \{1, \ldots, n-1\}$ let Z be a directed cycle in T_i whose edges have positive weights only. Consider now those edges $\overline{e} \in Z$ for which we have $\omega_{\overline{e}} = \delta_Z := \min\{\omega_e \mid e \in Z\} > 0$. We remove these edges from T_i and add them to E_s. This modification is done for all directed cycles with positive edge weights only and all $i \in \{1, \ldots, n-1\}$. We sort the nodes in W_i $(1 \le i \le n-1)$ topologically according to the order induced by $E'_{\le}(T_i \setminus E_s, T_i \setminus E_s)$ (cp. p. 13): $W_i = \{v_1^{(i)}, v_2^{(i)}, \ldots, v_{m_i}^{(i)}\}$. Since $T_{\mathcal{U}}$ is a tree, we can sort the sets $W_1, \ldots, W_{n-1}$ topologically according to the order induced by the depth of the W_i. We assume without loss of generality that the given numbering already reflects this ordering. Both sortings induce a sorting on $\bigcup_{i=1}^{n-1} W_i$. Set

$$\mathcal{W} := \{K, \{v_1^{(1)}\}, \{v_2^{(1)}\}, \ldots, \{v_{m_1}^{(1)}\}, \{v_1^{(2)}\}, \ldots, \{v_1^{(n-1)}\}, \ldots, \{v_{m_{n-1}}^{(n-1)}\}, S\}.$$

As it has been done in the case of attraction set trees (p. 10), we can now construct weights which fulfill (2.2a)–(2.2c) and (2.3). By the derivation of the last section it is clear that each extended attraction set tree can be detected by the conditions developed here.

5. Numerical Results. We will consider the following problem type. In this section let $X := \mathbb{R}^N$ with $2 \le N \le 20$, $C = (\mathbb{R}^N)^V$ as well as $d_e = \omega_e d$, $\omega_e \in \mathbb{R}$. Here, d is the metric induced by a p-norm $(1.1 \le p \le 5)$. Almost all test cases of the literature known to the author belong to this class of problems. For a discussion of the modeling aspects with respect to the choice of the parameter p, we refer, e. g., to [5, 7, 8] and [35, p. 232]. The polynomial time algorithm described in § 3 and 4 works on problems from this class. We define, as it has already been done in § 4,

$$E_{\le} := \{e \in E \mid 0 < \omega_e\}$$

and

$$E_{\ge} := \{e \in E \mid 0 > \omega_e\}.$$

5.1. The Nonlinear Programming Method. What kind of numerical algorithm should be used to solve a problem from the given class of nonconvex nondifferentiable optimization problems? In the literature known to the author only algorithms for the convex case, i. e. the case of positive weights $(\omega_e > 0, e \in E)$ (cp., e. g., [9, 27, 3]) or for the case $|V| = 1$ [10] are considered. Since our closest attention lies on the treatment of comparatively large scale unconstrained optimization problems, we use the conjugate gradient method in the modified form of Polak and Ribière (PRCG$^+$ method) to minimize the objective function [25, Section 4.8]. Several tests in the literature confirm the competitiveness of this method, especially for large problems [33]. Moreover, the method needs to store only 4 vectors of length $N|V|$. Wesolowsky and Love [38] have already applied a CG-method on a special minisum multilocation problem. However, they have not documented any implementation details. Of course, the method may converge to a local minimum only. Therefore, in actual applications it has to be used with multistart techniques or in conjunction with a method from global optimization.

Since the PRCG$^+$ method works on continuously differentiable objective functions only, we smooth the norms by using hyperbolic approximation [38, 15, 32, 23]. The approximation parameter ε is set to $\varepsilon := 1$ at the start of each iteration and damped by the factor 0.75 in each iteration step.

A restart is made after each mth iteration by using the negative gradient as the descent direction. The parameter m was set to $m := 1+\lfloor |V|N/2 \rfloor$. If the number $|V|$ is reduced by way of a successful coincidence test (cp. § 5.2), the number m is computed anew. Other restarts are made after unsuccessful line searches. In this line search, the simple Armijo criterion [4] with $\sigma := 1 \times 10^{-4}$ and quadratic Hermite interpolation has been used.

If the number of iterations is large, the approximation parameter ε used in the hyperbolic approximation will get so small that rounding errors occurring at coincidences will prevent a precise computation of $\nabla\Psi$. If, on the other hand, the approximation parameter stays too large, we cannot use solutions of the approximated problem to come to conclusions about solutions of the original problem. This means that the usage of the approximated gradient in a stopping criterion of the iteration has to be augmented by other criteria. Therefore, we stop with the approximation $X^{(t)} = (x_v^{(t)})_{v\in V}$ in the tth iteration, if the last line search was successful and for $X^{(t)}$ and the approximation $X^{(t-1)}$ from the $(t-1)$th iteration we have that

$$\|\nabla\Psi(X^{(t)})\|_2 < \varepsilon_g(1 + |\Psi(X^{(t)})|) \tag{5.1}$$

or

$$\Psi(X^{(t-1)}) - \Psi(X^{(t)}) < \varepsilon_f(1 + |\Psi(X^{(t)})|) \tag{5.2}$$

holds with certain problem specific stopping parameters $\varepsilon_f, \varepsilon_g$ (cp., e. g. [25, S. 300ff.]). If the last line search was not successful or the last step length fulfills $\alpha < \sqrt{\texttt{eps}}$ with the machine precision `eps`, the only stopping criterion used is (5.1). The sensitivity of the method with respect to the stopping parameters was found to be rather small [19, Tabelle 5.2–5.3].

To compute a starting family $(x_v)_{v\in V} \in (\mathrm{IR}^N)^V$, we define $\underline{c}_i := \min_{a\in A}(x_a)_i$, $\overline{c}_i := \max_{a\in A}(x_a)_i$ $(1 \le i \le N)$ and choose the ith coordinate of $x_v \in \mathrm{IR}^N$ $(v \in V)$ randomly by using the uniform probability distribution on the interval $[\underline{c}_i, \overline{c}_i]$. Therefore, we have

$$x_v \in [\underline{c}_1, \overline{c}_1] \times \cdots \times [\underline{c}_N, \overline{c}_N] \subset \mathrm{IR}^N$$

for all $v \in V$. If $\omega_e \ge 0$ holds for all edges $e \in E$, at least one minimum of the objective function Ψ is contained in the product of these cubes [26, 16]. Naturally, this property does no longer hold if negative weights exist. However, in the test cases used below only a comparatively small number of weights is negative. As a consequence, the generation technique just described still seems to be an efficient way to generate starting points.

The program was developed under SPARCompiler Pascal 3.0 on a Sun SPARCstation-10 with SunOS Release 5.5.1 and compiled with the compiler option `-O4`. Double precision arithmetic was used throughout the whole program. Crosscompiling the source to C by using `p2c` version 1.20 posed no problems. However, program maintenance and testing was done with the original code.

5.2. Implementation of the Coincidence Tests. If a coincidence test shall be carried out with the methods from § 3 and 4, we have to provide a candidate set $U \subset F$ with $|U \cap A| \leq 1$. Recent results from complexity theory [31] make it likely that choosing the candidate set is more complex than detecting the coincidence itself. We will therefore use the following heuristic: after each iteration of the PRCG$^+$ method in which a new approximation $(x_v)_{v \in V}$ has been computed, we define the subgraph $\tilde{G} := (V, \tilde{E})$ of G as follows. Two nodes $v, w \in V$ are connected by an edge in $\tilde{G}$ if and only if the corresponding locations are "close to each other" and the nodes are already connected by a positive weighted edge in G. If $\underline{c}_i$ and $\overline{c}_i$ $(i = 1, \ldots, N)$ are defined as in § 5.1 we have with the current iteration vector $(x_v)_{v \in V}$ the relation $(v, w) \in \tilde{E}$ if and only if

$$\sum_{i=1}^{N} \frac{|(x_v)_i - (x_w)_i|}{\max\{1, \overline{c}_i - \underline{c}_i\}} < \delta$$

and

$$(v, w) \in E_{\leq}.$$

Here, $\delta \in \mathrm{IR}$ is a parameter which has to be chosen by the user. Each connectivity component of $\tilde{G}$ is then tested for coincidences. The combinatorial aspect to be found in all known coincidence tests enters here due to the the possibility that these connectivity components may be too large or too small. These sets may have too many or too few elements. One possibility consists now in applying the test derived on all subsets of each connectivity component. To reduce the computational effort, we use the following strategy instead: let $\tilde{U} \subseteq V$ be the node set of a connectivity component of $\tilde{G}$. All subsets $U \subseteq \tilde{U}$ with n elements, $n = 1, 2, 3, 4, 5$ and $n = |\tilde{U}| - 1$ as well as $n = |\tilde{U}|$ will be used as candidate sets for the algorithm described in § 3. As kernel K, $|K| = 1$, we will use subsequently all sets

$$K \subseteq (\tilde{U} \setminus U) \cup \bigcup_{v \in \tilde{U}} \boldsymbol{F}_{\leq}(v, A).$$

As it has been shown in § 3, we only need to construct a maximal $(K, F \setminus (U \cup K))$-flow on a certain transportation network and test saturation on all edges leading to a sink. Remark 3.6 and 3.7 show that it is advantageous to consider increasing cardinalities of the candidate set U and test all kernel K possible first before using another candidate set. The implementation follows this idea. If a coincidence test is successful, the nodes from the corresponding test set are identified with the node in the kernel immediately. The edges incident to those nodes are treated in an analogous way.

The strategy just described for computing the graph $\tilde{G}$ and the subsequent coincidence tests are carried out after each step of the CG-method until the Tth iteration. Here, $T \in \mathrm{IN}_0$ is a parameter that has to be supplied by the user. That no coincidence test at all will be carried out can be achieved by setting $T = 0$.

The algorithm used to construct a maximum flow is the one from Edmonds and Karp [13] in the presentation of Gibbons [24]. More efficient algorithms exist [1, 2], but first test runs showed already such a clear improvement on earlier results that further implementations or the use of other methods have been postponed. Note that the coincidence test routine makes use of the topology of the graph G and the edge weights only, i. e. the implementation is independent of the weak metric used.

5.3. Test Problems. Most of the problems described in the literature are rather small. Moreover, negative weights are seldom considered, because the algorithms proposed by other authors usually do not work for this case. Therefore, we concentrate here on stochastically generated large scale test cases with positive and negative weights. A detailed discussion of the test results for test problems from the literature can be found in [19, Section 5.4].

In this section, the phrase "to choose from the interval $[\alpha, \beta\,]$" is used whenever a number in the interval $[\alpha, \beta\,]$ is computed by means of a random number generator simulating the uniformly distributed density on $[0, 1\,]$.

For a given dimension N of the locational space, given $|V|$, $|V| > 1$, and given $|A|$, we will generate random problems as follows: The probability for the existence of an edge connecting two nodes in V with each other is $5/(|V| + 10)$. The probability for an edge $e \in E(V, A)$ is $1/(|V| + |A|)$. A weight $\omega_e \in \mathbb{R}$ $(e \in E)$ is negative with probability 0.05. If this is the case, it is chosen from the interval $[-0.8 \ln |V|, -1\,]$. Otherwise, it is chosen with probability 0.95 from the interval $[1, 1 + \ln |V|\,]$ and with probability 0.05 from the interval $[1, 1 + (\ln |V|)^2\,]$. The average relative number of negative weights is now 5%. Raising this average or reducing the negative weights often leads to an objective function not bounded below.

After completing the process of generating edges and weights as described above, we determine two sets of nodes to simulate the clusters of nodes often found in applications. These two sets are defined by edges that carry relatively large weights. For $n := \lfloor |V|/5 \rfloor$ we choose randomly two $n+1$-tuples $(f_0, \ldots, f_n), (g_0, \ldots, g_n) \in F \times V^n$ with pairwise independent components $f_0, g_0 \ldots, f_n, g_n$. In addition to the edges generated above we add the edges $(f_i, f_{i+1}), (g_i, g_{i+1})$ $(i = 0, \ldots, n-1)$ to the graph and choose corresponding weights from the interval $[1, 1+|V|\,]$. If such an edge (f, g) or its counterpart (g, f) already exists, we identify the edges and add the weights together. After generating all edges we round off the weights to integer numbers.

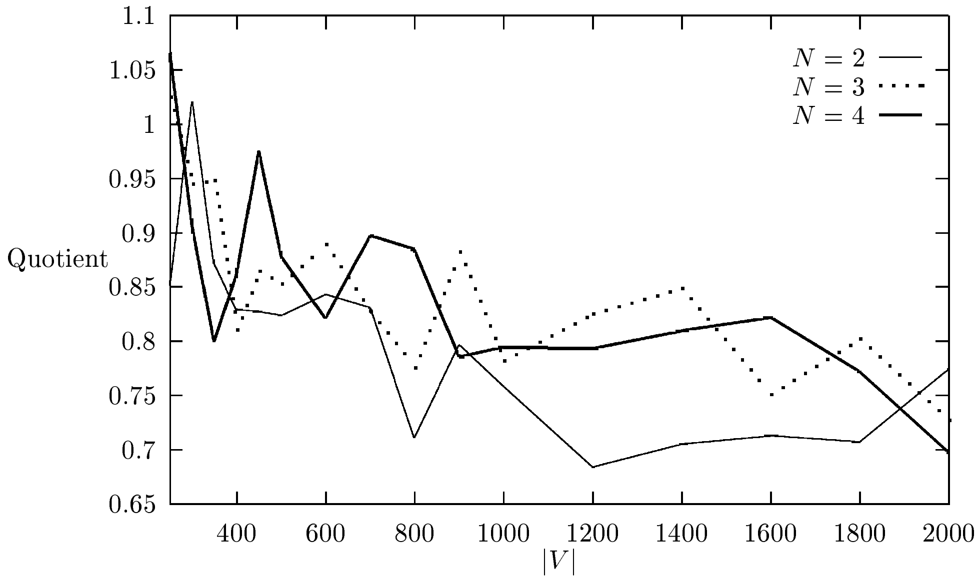
Note that the test cases generated in this way are separable in the sense that the graph $(V, E \cap (V \times V))$ may contain several connectivity components. Moreover, it may be the case that $\boldsymbol{E}(V, a) = \emptyset$ holds for an $a \in A$, which means that this object is not connected to the rest of the graph. Both of these problems will be detected by the code; the optimization process will take place on the objective functions induced by the connectivity components [34, 17, 20]. However, the test problems generated never had more than six connectivity components, even in the case $|V| = 2000$. For the problem mentioned with six components, five of these components contained exactly one node. Similar numbers hold for all other problems generated.

The ith coordinate of $x_a \in \mathbb{R}^N$ $(a \in A)$ is chosen from the interval $[0, 100\,]$. With this, we have $x_a \in [0, 100]^N$ for the given dimension $N \in \mathbb{N}$. The p-norm used is determined by choosing the parameter p from the interval $[1.1, 5\,]$.

The problem generation process just described is carried out for the dimensions $N = 2, 3, 4, 5, 10, 15, 20$ and the following $(|V|, |A|)$-pairs: $(|V|, |A|) = (10k, 10(k + 1)$ for $k = 1, \ldots, 8$, $(|V|, |A|) = (100 + 10k, 100 + 10(k + 2))$ for $k = 0, \ldots, 9$, $(|V|, |A|) = (200, 230)$, $(|V|, |A|) = (100k, 100k + 50)$ for $k = 2, \ldots, 5$, $(|V|, |A|) = (100k + 50, 100(k + 1))$ for $k = 2, 3, 4$, $(|V|, |A|) = (100k, 100(k + 1))$ for $k = 6, \ldots, 9$ as well as $(|V|, |A|) = (1000 + 200k, 1000 + 200(k+1))$ for $k = 0, \ldots, 5$. Consequently, we consider 252 problems with up to 40000 unknowns.

The parameters described in § 5.1 and 5.2 are chosen problem specific as follows: $\varepsilon_f := \varepsilon_g := 5 \times 10^{-6}$, $\delta := d/(|V| + |A|)$, and $T := 1 + \lfloor |V|/15 \rfloor$.

FIG. 5.1. *Smallest objective function value found with coincidence tests divided by the smallest objective function value found without coincidence tests ($N = 2, 3, 4$).*



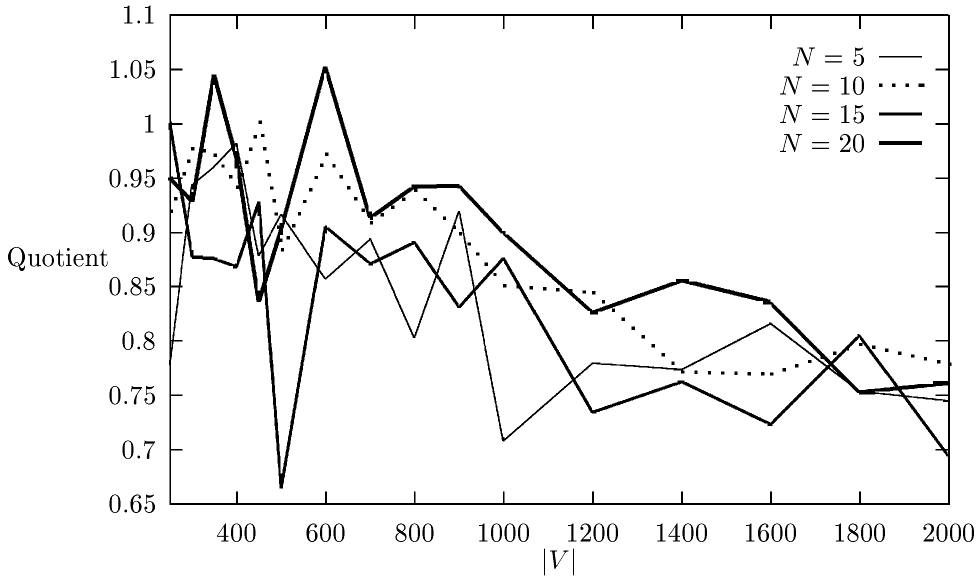
5.4. Test Results. All test runs have been made on a SUNSparc Ultra-2 with 256 MB main memory and a machine constant of $\texttt{eps} \approx 2.22 \times 10^{-16}$, using SunOS Release 5.5.1. A comparison shows that the PRCG$^+$ method employed here seems to be competitive to algorithms from the literature [19, Tabelle 5.1].

In what follows, we briefly describe the performance of the code on the test problems with respect to the best objective function values found, the number of iterations and function evaluations, and the computation times. A more detailed discussion can be found in [19].

5.4.1. Objective Function Values. Figure 5.1 and 5.2 show the quotient of the smallest objective function value found with coincidence tests and the smallest objective function value found without coincidence tests for all problems with $|V| \geq 300$. Reported are the objective function values for the largest connectivity component in the graph only. In all problems considered, the other connectivity components were trivial. Quotients less than 1 correspond to a decrease in the best objective function value found due to coincidence tests. As described above, the number of coincidence tests T was set anew for each problem considered.

In 191 of the 252 problems considered there was no deterioration or an improvement of up to 50%. For one problem, the best objective function value was more than halved, for two problems the decrease was between 50 and 60%, for four problems the decrease was between 60 and 70%, for 33 problems the decrease was between 70 and 80%, for 61 problems the decrease was between 80 and 90%, and for 66 problems the decrease was between 90 and 100%. For problems with $|V| \geq 1000$ all quotients lie between 0.68 and 0.9. This means that in these cases the objective function values have been significantly reduced. On the other hand, an increase in objective function values when using coincidence tests has been observed for 61 problems. However, in 42 of these 61 problems, the increase was less than 10%. Moreover,

FIG. 5.2. *Smallest objective function value found with coincidence tests divided by the smallest objective function value found without coincidence tests ($N = 5, 10, 15, 20$).*



no deterioration was observed for problems with $|V| > 600$. For $200 < |V| \leq 600$, all increases in objective function values were less than 6%.

As an example, Figures 5.3–5.6 (p. 24–26) show the typical behaviour of the algorithm used with and without coincidence tests on large scale problems. In the first iterations, decreases in function values when using coincidence tests appear to be as good as the decreases when not using coincidence tests. But without coincidence tests the decrease slows down until a suboptimal point is reached. With coincidence tests, the stepwise decrease remains relatively efficient, until again strong improvements do not occur any longer. In total, better function values are achieved with less iterations by way of coincidence tests.

The coincidence tests implemented seem to delay the typical phenomenon of premature stopping of algorithms from nonlinear optimization, well observed for this class of problems [23]. This is especially clear for the large scale problems considered. For stochastically generated problems with smaller dimension, the results are not as uniform.

5.4.2. Iterations and Function Evaluations. Table 5.1 shows the number of iterations, the number of function evaluations, and the number of edge evaluations, i. e. the number of computations of the form $\omega\|x - y\|_p$ or $\omega\nabla(\|z\|_p)|_{z=x-y}$. Since the trivial connectivity components possibly existent in the graph may distort the overall impression, the iteration numbers cover the largest connectivity component only. However, the number of function and edge evaluations is the total number of these evaluations for a given problem. For example, the test case with $N = 20$, $|V| = 2000$, $|A| = 2200$ has two trivial connectivity components with one node in V and one edge each. Without coincidence tests, 49 and 58 iterations were needed to minimize the corresponding two separated objective functions. With coincidence tests, 2 iterations for each were needed.

The average amount of reduction of edge evaluations per problem due to coin-

TABLE 5.1
Sum of iterations, function evaluations, and edge evaluations for different dimensions.

	iterations		function eval.		edge eval.	
N	$T>0$	$T=0$	$T>0$	$T=0$	$T>0$	$T=0$
2	3783	5305	15325	22802	35400607	52791090
3	4248	6731	16571	24857	39438837	66816811
4	5034	7427	19138	27099	48542483	73993688
5	5533	8208	20580	30467	48330410	78907387
10	7693	12653	25262	36689	61838056	128185983
15	8679	11975	28403	41025	72624636	121766039
20	14634	20207	41565	57226	99646015	179084508

TABLE 5.2
Sum of run times (in minutes) for different dimensions.

N	$T>0$	$T=0$
2	8.3583	10.4715
3	10.3818	15.8697
4	14.4568	21.7636
5	16.0166	24.9682
10	32.0816	65.5998
15	52.4100	97.1467
20	88.7216	174.0734

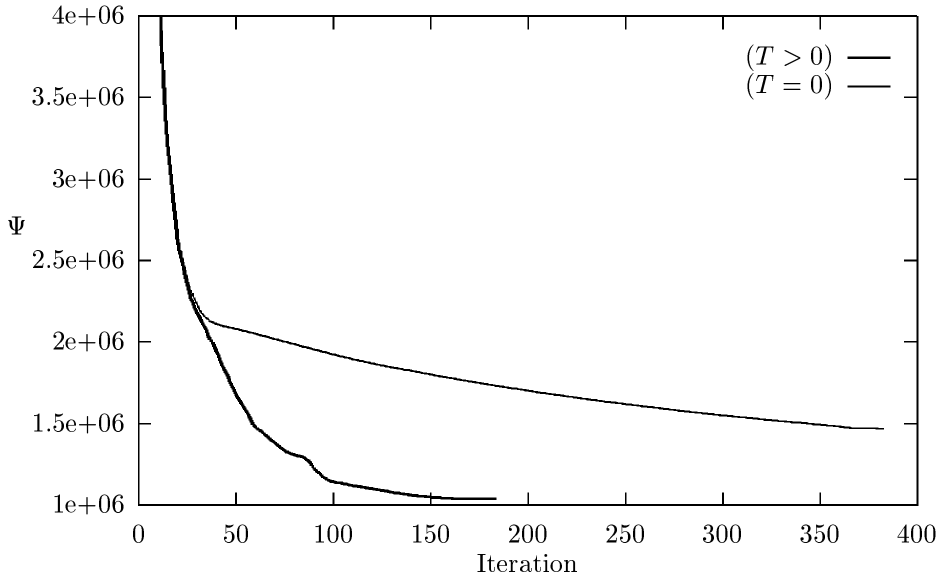
cidence tests was 8.1% for $N = 2$, 14.3% for $N = 3$, 14.4% for $N = 4$, 10.4% for $N = 5$, and 16.5% for $N = 10$. For $N = 15$ there was an additional amount of work of 5.6%, while there was an additional amount of work of 1.5% for $N = 20$. These two deteriorations are due to singular bad results for $N = 15$, $|V| = 70$, $|A| = 80$ and $N = 20$, $|V| = 200$, $|A| = 250$. Without these two cases, we have an average amount of reduction of 3.8% for $N = 15$ and of 8.5% for $N = 20$. Analogous results hold for the number of function evaluations and the number of iterations. The total number of evaluations is reduced by a factor of ca. 2/3.

5.4.3. Computation Times. The computation times for the randomly generated problems considered are reported in Table 5.2. The documented times have been obtained with the UNIX `time` command. Input and output times are included in the figures given, since the set-up of the data structures needed (dynamically linked adjacency lists necessary for representing G) is an integral part of the algorithm. Profiles of the program developed show that ca. 10% of the computation time is needed for input, output, and set-up of data structures.

It can be clearly seen that the computation times with coincidence tests are far below the computation times without coincidence tests. In 192 of the 252 problems considered there have been observed computation time savings of up to a factor of 16. For fixed N, the average amount of reduction per problem is 7% for $N = 2$, 17% for $N = 3$, 20% for $N = 4$, 14% for $N = 5$, 19% for $N = 10$, -3% for $N = 15$ (This result is due to a singular deterioration for $|V| = 70$, $|A| = 80$. Without this case, the amount of reduction is 6%.), and 3% for $N = 20$. If only problems with $|V| > 400$ are considered, the average amount of reduction per problem is 16% for $N = 2$, 28% for $N = 3$, 30% for $N = 4$, 27% for $N = 5$, 38% for $N = 10$, 31% for $N = 15$, and 32% for $N = 20$. It seems that especially for large problems coincidence tests are responsible for a considerable reduction of the work needed. The total amount of computation time is approximately halved.

Profiles of the code during the solution of the larger test problems agree with the

FIG. 5.3. *Objective function values until the end of the iteration for $N = 2$, $|V| = 1800$, and $|A| = 2000$, with ($T > 0$) and without ($T = 0$) coincidence tests.*



reductions found in computation time and edge evaluations. Only 3% of the total computation time is used for coincidence tests. In contrast, function and gradient evaluations need about 80% of the total computation time.

5.5. Conclusions. In almost all problems with $|V| > 300$, using coincidence tests resulted in a decrease of best objective function values, iterations, function and edge evaluations, and computation time. These effects became more evident when the number of unknowns was increased.

More gains in efficiency with respect to computation times of methods which use coincidence tests have to be expected if the parameters T and δ, which control the number and the probability of success of these tests, are chosen optimal with respect to the problem class given. To this end, however, extensive numerical tests are necessary. Of course, the heuristic used here (cp. § 5.2) can be refined or replaced entirely.

If the metrics used are more realistic or more complex than the ones induced by p-norms, we will encounter a corresponding higher amount of work needed to compute a distance. For example, in discrete multilocation problems the locational space X is a graph and the distance between two points is induced by the weak metric of shortest paths. Therefore, we have to compute a shortest path between two points as soon as we want to compute the distance between them and the graph is sufficiently large [14]. Here, it is of prime importance to reduce the number of unnecessary distance calculations. The algorithms employed to construct candidate sets for coincidence tests may in this case be rather inefficient.

If global optimization problems of the scale considered here are investigated, restarts of the optimization method used are unavoidable to compute good approximative solutions. But for these restarts, coincidence tests are no longer necessary, since we can expect that we have already detected all detectable coincidences in former runs. Therefore, the computation times documented here correspond to the first

FIG. 5.4. *Objective function values until the end of the iteration for $N = 2$, $|V| = 2000$, and $|A| = 2200$, with ($T > 0$) and without ($T = 0$) coincidence tests.*

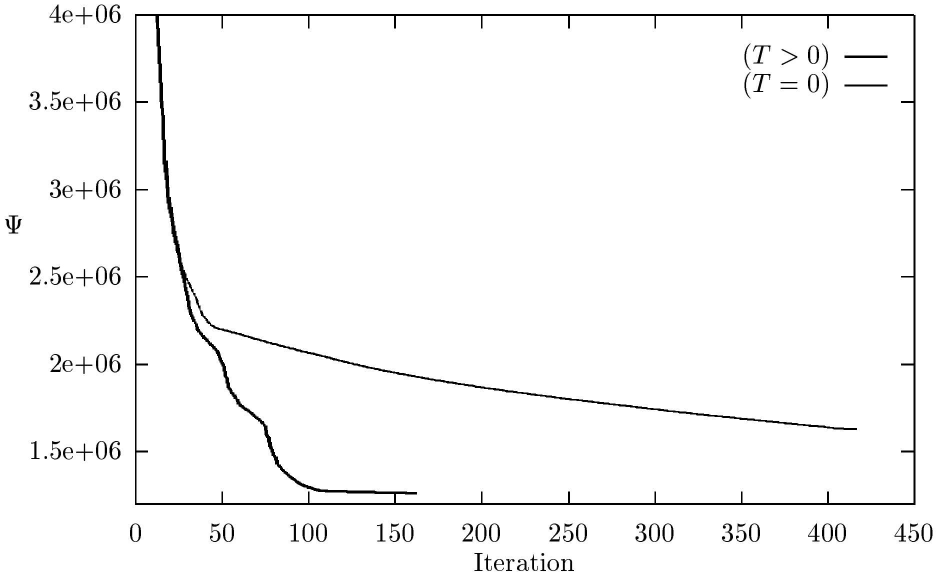
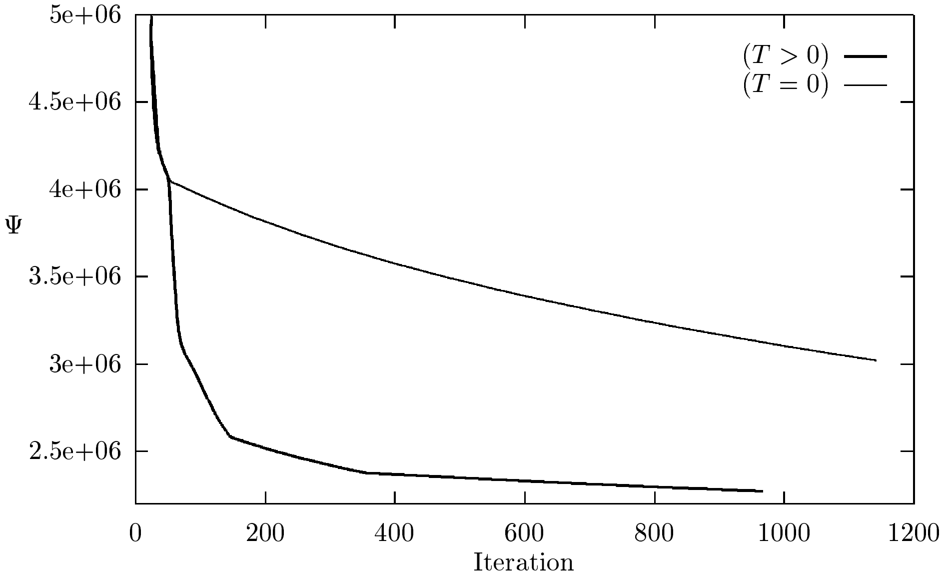


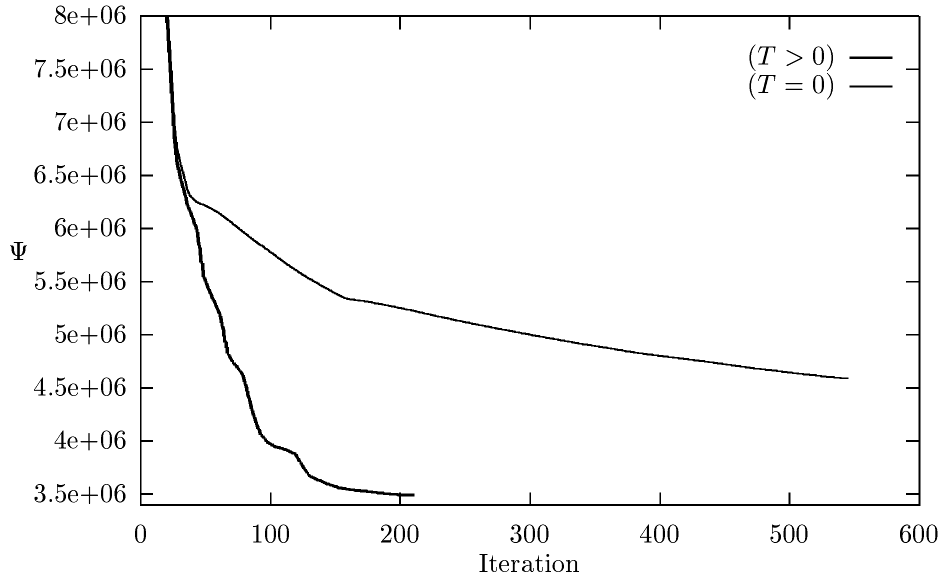
FIG. 5.5. *Objective function values until the end of the iteration for $N = 20$, $|V| = 1800$, and $|A| = 2000$, with ($T > 0$) and without ($T = 0$) coincidence tests.*



run of a multistart method. In all further runs, the collapsed problems will be solved faster than the original ones.

As a consequence, we can recommend the coincidence tests proposed here, as long as we are not only in search for a very rough approximation to an optimal solution. The tests work especially well when large scale problems are considered.

FIG. 5.6. *Objective function values until the end of the iteration for $N = 20$, $|V| = 2000$, and $|A| = 2200$, with ($T > 0$) and without ($T = 0$) coincidence tests.*



REFERENCES

[1] R. K. AHUJA, T. L. MAGNANTI, AND J. B. ORLIN, *Some Recent Advances in Network Flows*, SIAM Review, 33 (1991), pp. 175–219. Also published as [2].

[2] ———, *Network Flows; Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] K. D. ANDERSEN, *An efficient Newton barrier method for minimizing a sum of Euclidean norms*, SIAM Journal of Optimization, 6 (1996), pp. 74–95.

[4] L. ARMIJO, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of Mathematics, 16 (1966), pp. 1–3.

[5] W. BERENS, *The suitability of the weighted l_p-norm in estimating actual road distances*, European Journal of Operation Research, 34 (1988), pp. 39–43.

[6] C. BERGE AND A. GHUILA-HOURI, *Programming, Games and Transportation Networks*, Methuen and Co. Ltd., John Wiley and Sons Inc., London, New York, 1965. First published in France as *Programmes, Jeux et Réseaux de Transport* by Dunod, Paris, 1962.

[7] J. BRIMBERG AND R. F. LOVE, *Directional bias of the l_p-norm*, European Journal of Operation Research, 67 (1993), pp. 287–294.

[8] ———, *Estimating distances*, in Drezner [12], ch. 1, pp. 9–32.

[9] P. H. CALAMAI AND A. R. CONN, *A projected Newton method for l_p norm location problems*, Mathematical Programming, 38 (1987), pp. 75–109.

[10] P.-C. CHEN, P. HANSEN, B. JAUMARD, AND H. TUY, *Weber's problem with attraction and repulsion*, Journal of Regional Science, 32 (1992), pp. 467–486.

[11] R. COMINETTI AND C. MICHELOT, *Sufficient conditions for coincidence in ℓ_1-minisum multifacility location problems*, Operations Research Letters, 20 (1997), pp. 179–185.

[12] Z. DREZNER, ed., *Facility Location: A Survey of Applications and Methods*, Springer Series in Operations Research, Springer Verlag, New York, 1995.

[13] J. EDMONDS AND R. M. KARP, *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of the Association of Computing Machinery, 19 (1972), pp. 248–262.

[14] G. ERTL, *Shortest path calculation in large road networks*, Bericht 69, Spezialforschungsbereich F 003 (Optimierung und Kontrolle), Projektbereich Diskrete Optimierung, Karl-Franzens-Universität Graz & Technische Universität Graz, May 1996. Electronically distributed via `http://www.opt.math.tu-graz.ac.at/cgi-bin/replist`.

[15] J. W. EYSTER, J. A. WHITE, AND W. W. WIERWILLE, *On Solving Multifacility Location Problems Using a Hyperboloid Approximation Procedure*, AIIE Transactions, 5 (1973), pp. 1–6.

[16] A. Fajfar, *Verbesserung von Schnittebenenverfahren zur Lösung von Minisum-Multilokationsproblemen*, Diplomarbeit, Universität Dortmund, Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany, March 1996.

[17] J. Fliege, *Nichtdiskrete Multilokationsprobleme: Koinzidenzbedingungen und effiziente Punkte*, Diplomarbeit, Universität Dortmund, Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany, February 1993.

[18] ———, *Some new coincidence conditions in minisum multifacility location problems with mixed gauges*, in EWGLA 7 Proceedings, B. Boffey and J. Karkazis, eds., no. 7 in Studies in Locational Analysis, Smirnis 1, Zografu, Athens 15772, Greece, November 1994, Constantine Porphyrogenetus International Association, pp. 49–60.

[19] ———, *Effiziente Dimensionsreduktion in Multilokationsproblemen*, PhD thesis, Universität Dortmund, Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, June 1997. Shaker Verlag, Aachen, Germany. (An english translation is available from the author on request.).

[20] ———, *Nondifferentiability Detection and Dimensionality Reduction in Minisum Multifacility Location Problems*, Journal of Optimization Theory and Applications, 94 (1997), pp. 365–380.

[21] ———, *Coincidence conditions in multifacility location problems with positive and negative weights*, European Journal of Operations Research, 104 (1998), pp. 310–320. Feature Issue. Twelveth [sic!] EURO Summer Institute: Locational Analysis. Guest Editor: J[aime] Barceló.

[22] ———, *Multithreaded Dimension Reduction in Location Theory*, Ergebnisberichte Angewandte Mathematik 153, Universität Dortmund, Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany, April 1998. Submitted to *SIAM Journal on Optimization.*

[23] J. B. G. Frenk and M. J. Kleijn, *On a Class of Algorithms solving the Multifacility Location Problem with Perturbed l_p-distances*, Discussion paper TI 94–74, Tinbergen Institute, Tinbergen Institute Rotterdam, Oostmaaslaan 950–952, 3063 DM Rotterdam, The Netherlands, 1994.

[24] A. Gibbons, *Algorithmic graph theory*, Cambridge University Press, Cambridge, London, New York, New Rochelle, Melbourne, Sidney, 1985.

[25] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*, Academic Press Limited, Harcourt Brace and Company, London, 1995.

[26] P. Hansen, J. Perreur, and J.-F. Thisse, *Location Theory, Dominance, and Convexity: Some Further Results*, Operation Research, 28 (1980), pp. 1241–1250.

[27] H. Idrissi, O. Lefebvre, and C. Michelot, *Duality for constrained multifacility location problems with mixed norms and applications*, vol. 18 of Annals of Operations Research, Basel, Switzerland, February 1989, J. C. Baltzer AG Scientific Publishing Company, pp. 71–92.

[28] O. Lefebvre, C. Michelot, and F. Plastria, *Sufficient conditions for coincidence in minisum multifacility location problems with a general metric*, Operation Research, 39 (1991), pp. 437–442.

[29] ———, *Boolean linear programming formulation of the attraction tree detection problem*, in Proceedings of the VIth meeting of the EURO working group on locational analysis, J. A. M. Pérez, ed., Puerto de La Cruz (Tenerife), Spain, October 1992, pp. 123–134.

[30] C. Michelot. Private communication, September 26 1995. Lambrecht, EWGLA8 — EURO Working Group on Locational Analysis.

[31] C. Michelot and F. Plastria, *Attraction Tree Detection is NP-Complete.* Rapport du CERMSEM 9107, Université de Paris I — Panthéon-Sorbonne.

[32] J. G. Morris and W. A. Verdini, *Minisum l_p-distance location problems solved via a perturbed problem and Weizsfeld's algorithm*, Operation Research, 27 (1979), pp. 1180–1188.

[33] J. Nocedal, *The performance of several algorithms for large scale unconstrained optimization*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM Proceedings Series, Philadelphia, Pennsylvania, 1990, SIAM Society for Industrial and Applied Mathematics, pp. 138–151.

[34] F. Plastria, *When Facilities Coincide: Exact Optimality Conditions in Multifacility Location*, Journal of Mathematical Analysis and Applications, 169 (1992), pp. 476–498.

[35] ———, *Continuous Location Problems*, in Drezner [12], ch. 11, pp. 225–262.

[36] ———, *Asymmetric distances, semidirected networks and majority in Fermat-Weber problems*, Report BEIF/96, Center for Management Informatics, Vrije Universiteit Brussel, Belgium, April 1997. Submitted.

[37] H. Ribeiro, *Sur les espaces à métrique faible*, Portugaliae Mathematica, 4 (1943), pp. 21–40.

[38] G. O. Wesolowsky and R. F. Love, *A nonlinear approximation method for solving a gener-*

alized rectangular distance Weber problem, Management Science, 18 (1972), pp. 656–663.

[39] C. Witzgall, *Optimal Location of a Central Facility, Mathematical Models and Concepts*, Tech. Report 8388, National Bureau of Standards, Applied Mathematics Division, Operations Research Section, U. S. Department of Commerce, Washington, D. C., June 30 1964.

[40] ———, *On Convex Metrics*, JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematics and Mathematical Physics, 69B (1965), pp. 175–177.

[41] X. Xu, P. fang Hung, and Y. Ye, *A simplified homogeneous and self-dual linear programming algorithm and its implementation*, Annals of Operations Research, 62 (1996), pp. 151–171.

[42] Y. Ye, M. J. Todd, and S. Mizuno, *An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm*, Mathematics of Operations Research, 29 (1994), pp. 53–67.