Multicriteria Network Location Problems with Sum Objectives

Horst W. Hamacher[†,*]
Fachbereich Mathematik
Universität Kaiserslautern

Martine Labbé[*]
Service de Mathématiques de la Gestion
Université Libre de Bruxelles

Stefan Nickel[†,*]
Fachbereich Mathematik
Universität Kaiserslautern

November 7, 1996

Abstract

In this paper network location problems with several objectives are discussed, where every single objective is a classical median objective function. We will look at the problem of finding Pareto optimal locations and lexicographically optimal locations. It is shown that for Pareto optimal locations in undirected networks no node dominance result can be shown. Structural results as well as efficient algorithms for these multi-criteria problems are developed. In the special case of a tree network a generalization of Goldman's dominance algorithm for finding Pareto locations is presented.

[*]Partially supported by grant ERBCHRXCT930 of the European HC&M Programme.
[†]Partially supported by a grant of the Deutsche Forschungsgemeinschaft.

1 Introduction

In an effort to improve the performance of facility systems, planners have developed a host of operational models to deal with the location of a facility on a network (see [Labbé et al., 1995] for a recent survey). A large number of these models focuses on the minimization of a unique objective function which is increasing with the distance to travel. However, in many location problems, especially in the public sector, the decision must be made by a group of decision makers, e.g. when a group of small communities wants to build and share a public facility. Then, each decision maker is endowed with a specific objective function but different decision makers have different objectives. In this paper, we consider the median (or minisum) location problem in this context of several decision makers. Specifically, the goal of all decision makers is the same: to locate a facility in order to minimize the total weighted distance to the potential users but the value of weights assigned to those potential users varies from one decision maker to another, hence yielding different objective functions. Furthermore, we consider two types of solutions depending on the fact that a hierarchy among the decision makers exists or not. In the first case and if the corresponding ranking is known, the solution we consider is called a lexicographic location and optimises the objective functions of the decision makers sequentially according to the given order. If this order exists but is not known, then some help might be provided to the decision maker group by giving them the set of the lexicographic locations with respect to all possible rankings. In the second case, some help might be again provided to the group through the set of efficient or Pareto locations, i.e. solutions such that there exists no other location which is not worse for all decision makers and better for at least one of them. The literature on location models involving several decision makers is somewhat reduced at least when they all share the same type of objective functions. Hansen et al. [Hansen et al., 1986] consider a special case of the model presented here. More precisely, they provide an algorithm for finding the efficient points for the so-called point-objective problem: the objective function of each decision maker is given by the distance to his/her own location. Hamacher and Nickel [Hamacher and Nickel, 1996] consider the planar version of the problems treated here. Finally, Lowe [Lowe, 1978] considers the case of several decision makers endowed with objective functions which are not necessarily of the same type. Note that his analysis restricts to the family of tree networks. Other types of multicriteria location models have been proposed, see e.g. [Labbé et al., 1995].

As one notices, the nomenclature for location problems is not unique. Therefore we introduce in the following a classification scheme for location problems, which should help to get an overview over the manifold area of location problems.

We use a scheme which is analogous to the one introduced successfully in scheduling theory. The presented scheme for location problems was developed in [Hamacher, 1995] and [Hamacher and Nickel, 1996].
We have the following five position classification

$$pos1/pos2/pos3/pos4/pos5\ ,$$

where the meaning of each position is explained in the following table:

Position	Meaning	Usage (Examples)
1	number of new facilities	
2	type of problem	**P** planar location problem **D** discrete location problem **G** network location problem
3	special assumptions and restrictions	$w_m = 1$ all weights are equal $\mathcal{R}$ a forbidden region
4	type of distance function	l_1 Manhattan metric $d(\mathcal{V}, \mathcal{V})$ node to node distance
5	type of objective function	$\sum$ Median problem max Center problem

If we do not make any special assumptions in a position, we indicate this by a $\bullet$.

The rest of the paper is organized as follows: First we give some definitions and develop some basic concepts which are needed later on. Section 3 is devoted to the lex location problems. In Section 4 we present solution algorithms for finding the Pareto locations on a general network. Section 5 shows that for tree networks better procedures can be found. The paper ends with some conclusions.

2 Definitions and basic concepts

By $\mathcal{N} = (\mathcal{G}, l)$ we denote a network with underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_M\}$ is the node set and $\mathcal{E} = \{e_1, \ldots, e_N\}$ is the edge set. If the graph is a tree we use $\mathcal{T}$ instead of $\mathcal{G}$. If $\mathcal{G}$ is a directed graph we denote edge $e \in \mathcal{E}$ as $e = (v_i, v_j)$, $v_i, v_j \in \mathcal{V}$, where v_j is the head and v_i is the tail of e. If $\mathcal{G}$ is undirected we use the notation $e = [v_i, v_j] = [v_j, v_i]$, $v_i, v_j \in \mathcal{V}$.

For every edge $e \in \mathcal{E}$, a function $l : \mathcal{E} \to \mathbb{R}_+$ assigns a positive real number as length to e. By $d(v_i, v_j)$ we denote the distance between v_i and v_j (from v_i to v_j in a directed graph), which is given by the length of a shortest path between v_i and v_j.

For technical reasons edges with length 0 are not allowed. But all presented algorithms can be modified to include also the case where $l(e) \geq 0$, $\forall e \in \mathcal{E}$.

A **point** x on a directed edge $e = (v_i, v_j)$ is defined as a couple $x = (e, t)$, $t \in [0, 1]$, with

$$d(v_k, x) := d(v_k, v_i) + tl(e) \text{ and } d(x, v_k) := (1 - t)l(e) + d(v_j, v_k) \ ,$$

for any $v_k \in \mathcal{V}$. A **point** on an undirected edge $e = [v_i, v_j]$ is defined as a couple $x = (e, t)$, $t \in [0, 1]$, with

$$d(v_k, x) := d(x, v_k) := \min \{d(v_k, v_i) + tl(e), (1 - t)l(e) + d(v_j, v_k)\} \ ,$$

for any $v_k \in \mathcal{V}$.

In particular $d(v_i, x) = tl(e)$ and $d(x, v_j) = (1 - t)l(e)$ holds for all points $x = (e, t)$. Since $(e, 0) = v_i$ and $(e, 1) = v_j$ all nodes of the network are also points of the network.

Note that points can only be defined on networks and not on graphs without edge lengths.

The set of all points of a network $(\mathcal{G}, l)$ is denoted by $\mathcal{P}(\mathcal{G})$. The sets $\{(e, t) \in \mathcal{P}(\mathcal{G}) \ : \ t \in (t_1, t_2) \ , \ t_1, t_2 \in [0, 1]\}$ and $\{(e, t) \in \mathcal{P}(\mathcal{G}) \ : \ t \in [t_1, t_2] \ , \ t_1, t_2 \in [0, 1]\}$, forming **subedges** of e, are denoted $(e, (t_1, t_2))$ and $(e, [t_1, t_2])$, respectively, for any $e \in \mathcal{E}$. Of course these sets are empty if $t_2 < t_1$. For the sake of simplicity, we write e instead of $(e, [0, 1])$, whenever this causes no confusion.

We assign a vector of weights $w_i = \begin{pmatrix} w_i^1 \\ \vdots \\ w_i^Q \end{pmatrix} \neq 0$ to every node $v_i \in \mathcal{V}$, with $w_i^q \geq 0$, $q \in \mathcal{Q} := \{1, \ldots, Q\}$.

A point $x = (e, t)$ on an edge $e = [v_i, v_j]$ is called a **bottleneck point** for component q if there exists some node v_k with $w_k^q > 0$, such that

$$d(v_k, x) = d(v_k, v_i) + d(v_i, x) = d(v_k, v_j) + d(v_j, x) \ .$$

Let B_{ij} denote the set of bottleneck points on the edge $[v_i, v_j]$. Notice that the set B_{ij} contains at most $|\mathcal{V}|$ elements.

We measure the quality of a point $x \in \mathcal{P}(\mathcal{G})$ with the multi-criteria median objective, defined by

$$f(x) := \begin{pmatrix} f^1(x) \\ \vdots \\ f^Q(x) \end{pmatrix} := \begin{pmatrix} \sum_{v_m \in \mathcal{V}} w_m^1 d(x, v_m) \\ \vdots \\ \sum_{v_m \in \mathcal{V}} w_m^Q d(x, v_m) \end{pmatrix}$$

in the undirected case and

$$f(x) := \begin{pmatrix} f^1(x) \\ \vdots \\ f^Q(x) \end{pmatrix} := \begin{pmatrix} \sum_{v_m \in \mathcal{V}} w_m^1 \left(d(x, v_m) + d(v_m, x)\right) \\ \vdots \\ \sum_{v_m \in \mathcal{V}} w_m^Q \left(d(x, v_m) + d(v_m, x)\right) \end{pmatrix}$$

in the directed case. From [Levy, 1967] it follows that $f^q(t) := f^q((e,t))$ is concave on $[0,1]$ for all $q \in \mathcal{Q}$.

In order to use $f(x)$ in an optimization context an order on $\mathbb{R}^Q$ has to be defined. In this paper two different orders are considered.

As default order we consider the component-wise one defined by

$$z_1 := \left(z_1^1, \ldots, z_1^Q\right) \leq \left(z_2^1, \ldots, z_2^Q\right) := z_2$$

if and only if $z_1^q \leq z_2^q$, $\forall q = 1, \ldots, Q$.

If at least one of the latter inequalities is strict, the notation $z_1 < z_2$ is used, and z_1 is said to **dominate** z_2. Consider a subset $\mathcal{W} \subseteq \mathbb{R}^Q$. A vector $z_1 \in \mathcal{W}$ is called **non-dominated** with respect to $\mathcal{W}$ if there is no vector $z_2 \in \mathcal{W}$ such that $z_2 < z_1$. The set of all Pareto vectors with respect to $\mathcal{W}$ is denoted by $\mathcal{W}_{par}$.

Let $\mathcal{S}$ be a subset of points of $\mathcal{P}(\mathcal{G})$. If $\mathcal{W} = \{f(x) : x \in \mathcal{S}\}$, then the set of all locations $x \in \mathcal{S}$ such that $f(x) \in \mathcal{W}_{par}$ is denoted by $\mathcal{X}_{par}(\mathcal{S})$. If $\mathcal{S} = \mathcal{P}(\mathcal{G})$, we simply write $\mathcal{X}_{par}$. A point $x \in \mathcal{X}_{par}(\mathcal{S})$ is called a **Pareto location** with respect to $\mathcal{S}$.

Following the classification introduced in the introduction $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{par}$ and $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{par}$ are the problems of finding all Pareto locations (i.e. with respect to $\mathcal{P}(\mathcal{G})$) in the undirected and directed networks, respectively.

Correspondingly, if $\mathcal{W} = \{f(v) : v \in \mathcal{V}\}$, the problems classified as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{par}$ and $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{par}$ consist in finding the set $\mathcal{X}_{par}(\mathcal{V})$ of nodes $v \in \mathcal{V}$ in undirected and directed networks, respectively, such that $f(v)$ is not dominated by any $f(v')$, $v' \in \mathcal{V}$. Such nodes are called **Pareto nodes**.

Another possibility of comparing vectors in $\mathbb{R}^Q$ is the lexicographical order. For two vectors $z_1 := (z_1^1, \ldots, z_1^Q), z_2 := (z_2^1, \ldots, z_2^Q) \in \mathbb{R}^Q$, we say that

$$z^1 \quad \underset{lex}{\leq} \quad z^2$$
$$\text{if and only if}$$
$$z = z' \qquad \text{or} \qquad z_1^p < z_1^p \quad \text{for} \quad p := \min_{q \in \mathcal{Q}} \{q : z_1^q \neq z_2^q\} \ .$$

If $\Pi(\mathcal{Q})$ is the set of all permutations of $\mathcal{Q} = \{1, \ldots, Q\}$, $x \in \mathcal{P}(\mathcal{G})$ is called **lex**(icographic minimum) **location** or **lex optimal** with respect to a permutation $\pi \in \Pi(\mathcal{Q})$ if

$$\left(f^{\pi(1)}(x), \ldots, f^{\pi(Q)}(x)\right) \underset{lex}{\leq} \left(f^{\pi(1)}(y), \ldots, f^{\pi(Q)}(y)\right)$$

for all $y \in \mathcal{P}(\mathcal{G})$. Notice that x is well-defined, since the lex order is a total order in $\mathbb{R}^Q$. Lex locations are useful if a preference in the Q objective functions f^q corresponding to a specific permutation $\pi \in \Pi(\mathcal{Q})$ is known. The set of optimal locations with respect to π is denoted by $\mathcal{X}^\pi_{lex}$. If the preference is not known in advance one may be interested to know the set $\mathcal{X}_{lex}$ of all lex locations with respect to all possible permutations of $\Pi(\mathcal{Q})$. Analogously to the Pareto location sets, we define, for $\mathcal{S} \subseteq \mathcal{P}(\mathcal{G})$, $\mathcal{X}_{lex}(\mathcal{S})$ as the set of lex locations with respect to $\mathcal{S}$ and $\mathcal{X}_{lex} := \mathcal{X}_{lex}(\mathcal{P}(\mathcal{G}))$.

3 Lexicographic Locations on Networks

In the first part of this section we consider the problems $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$ and $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$, where new facilities can only be placed in the nodes.

By f_i^q we denote the q-th objective value for node v_i, i.e.

$$f_i^q = f^q(v_i) = \sum_{v_m \in \mathcal{V}} w_m^q d(v_i, v_m) \ . \tag{3.1}$$

The values f_i^q can be computed in $O(|\mathcal{V}|^3)$ by applying an all-pair shortest path algorithm (see [Ahuja et al., 1993]). For $\pi \in \Pi(\mathcal{Q})$, we denote

$$f_i(\pi) := \begin{pmatrix} f_i^{\pi(1)} \\ \vdots \\ f_i^{\pi(Q)} \end{pmatrix} \text{ and } f_i := f_i(id) := \begin{pmatrix} f_i^1 \\ \vdots \\ f_i^Q \end{pmatrix} ,$$

where id denotes the identity permutation. Without loss of generality we assume that $f_i \neq f_j$, $\forall v_i, v_j \in \mathcal{V}, i \neq j$, such that for any $\pi \in \Pi(\mathcal{Q})$

the lex location is uniquely defined. Therefore $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$ and $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$ are equivalent to finding the lexicographically minimal vectors of the finite set $\{f_i(\pi) \ : \ v_i \in \mathcal{V} \ , \ \pi \in \Pi(\mathcal{Q})\}$, whose cardinality is $O(|\mathcal{V}|Q!)$.

We will show next how to check whether for a given vector, say f_1, there exists a permutation π such that $f_1(\pi)$ is lex optimal.

To that aim we compute for all $v_i \in \mathcal{V}$

$$Min_i := \left\{q \in \mathcal{Q} \ : \ f_i^q = \min\{f_j^q \ : \ v_j \in \mathcal{V}\}\right\}$$

and

$$\mathcal{J}(1) := \{j \ : \ Min_j \supseteq Min_1\} \ .$$

Lemma 3.1. *If $\mathcal{J}(1) = \{1\}$, then there exists some permutation π, such that $f_1(\pi)$ is lex optimal with respect to π.*

Proof. Choose any permutation π, such that for $M_1 = |Min_1|$, $\{\pi(1), \ldots, \pi(M_1)\} = Min_1$. Since $\mathcal{J}(1) = \{1\}$ implies $Min_j \cap Min_1 \subset Min_1$ for all $j \neq 1$, we get $f_1(\pi) <_{lex} f_j(\pi)$.

□

Lemma 3.2. *Let $\mathcal{J}(1) \supset \{1\}$. Then there exists some permutation π, such that $f_1(\pi)$ is lex optimal with respect to π if and only if*

$$\left(f_1^{\pi(q)}\right)_{q \in \mathcal{Q}\setminus Min_1}$$

is lex optimal with respect to the reduced permutation of $\mathcal{Q}\setminus Min_1$ in the set

$$\left\{\left(f_j^{\pi(q)}\right)_{q \in \mathcal{Q}\setminus Min_1} \ : \ j \in \mathcal{J}(1)\right\} \ .$$

Proof. The claim follows by the definition of the lexicographic order, since $Min_j \supset Min_1$ implies that $f_1^q = f_j^q$, for all $q \in Min_1$ and all $j \in \mathcal{J}(1)$.

□

Lemma 3.3. *If $Min_1 = \emptyset$, then there exists no π, such that $f_1(\pi)$ is lex optimal.*

Proof. $Min_1 = \emptyset$ implies that for any $\pi \in \Pi(\mathcal{Q})$, there exists some $v_j \in \mathcal{V}$ such that $f_1^{\pi(1)} > f_j^{\pi(1)}$.

□

Based on an iterative application of Lemmata 3.1 , 3.2 and 3.3 the following algorithm finds the set $\mathcal{X}_{lex}(\mathcal{V})$ of all lex optimal nodes with respect to $\mathcal{V}$.

Algorithm 3.1. *Solving* $1/\mathcal{G}, \mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$

Input: *Network* $\mathcal{N} = (\mathcal{G}, l)$ *or* $(\mathcal{G}_D, l)$
Output: $\mathcal{X}_{lex}(\mathcal{V})$

1. *Compute the distance Matrix* $D = (d_{ij})$.
2. *Compute* f_i^q, $v_i \in \mathcal{V}$, $q \in \mathcal{Q}$.
3. $Lex := \{1, \ldots, |\mathcal{V}|\}$. *Recall that the vectors* $f(v_i)$ *are assumed to be pairwise distinct. If this is not the case only one representing node is included in Lex.*
4. *for each* $i \in Lex$ *do*
   (a) *Determine* Min_i.
   (b) *if* $Min_i = \emptyset$, *then* $Lex := Lex \backslash \{i\}$.
5. $\mathcal{X}_{lex}(\mathcal{V}) := \emptyset$.
6. *for each* $i \in Lex$ *do*
   (a) $\mathcal{J}' := Lex$
   (b) $\mathcal{Q}' := \mathcal{Q}$
   (c) *for each* $j \in \mathcal{J}'$ *do compute*
   $$Min_j := \left\{q \in \mathcal{Q}' \ : \ f_j^q = \min\{f_l^q \ : \ l \in \mathcal{J}'\}\right\}$$
   (d) *if* $Min_i = \emptyset$ *then* $Lex := Lex \backslash \{i\}$, *goto* 6 *with next* i.
   (e) *compute* $\mathcal{J}'(i) := \{j \in \mathcal{J}' \ : \ Min_j \supseteq Min_i\}$.
   (f) *if* $\mathcal{J}(i) = \{i\}$ *then* $\mathcal{X}_{lex}(\mathcal{V}) := \mathcal{X}_{lex}(\mathcal{V}) \cup \{v_i\}$, *goto* 6 *with next* i.
   (g) $\mathcal{J}' := \mathcal{J}'(i)$, $\mathcal{Q}' := \mathcal{Q}' \backslash Min_i$, *goto* 6 *(c)*.
7. **Output:** $\mathcal{X}_{lex}(\mathcal{V})$

Step 1 requires $O(|\mathcal{V}|^3)$ elementary operations. The loop in Step 6 of Algorithm 3.1 can be run at most Q-times, since $|\mathcal{Q}'|$ is strictly decreasing in each iteration. Step 6 (c) is within that loop the most time consuming step and can be done in $O(Q|\mathcal{V}|\log|\mathcal{V}|)$. Hence the overall complexity of Algorithm 3.1 is $O(|\mathcal{V}|^3 + |\mathcal{V}|^2 Q^2 \log|\mathcal{V}|)$.

In a directed network, it is easy to see that $f^q(x) > f^q(v_j)$ for all $q \in \mathcal{Q}$ and all $x = (e,t)$ with $e = (v_i, v_j)$ and $0 < t < 1$ (see also Section 4.1). We therefore conclude with the following proposition.

Proposition 3.4. *The solution set for* $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{lex}$ *is the same as for* $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{lex}$.

It remains to consider the problems $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{lex}$.

For a given permutation $\pi \in \Pi(\mathcal{Q})$ the definition of the lexicographic order implies that $\mathcal{X}_{lex}^{\pi}$ can be found by a sequence of at most Q location problems, where the set of feasible solution in Step $q+1$ is given by the set of optimal locations of Step q.

The concavity property of the median functions f^q on $[0,1]$ (see [Levy, 1967]), implies that for any $x = (e,t)$, $e = [v_i, v_j]$, $0 < t < 1$ and for all $q \in \mathcal{Q}$:

$$f^q(x) \geq \min\{f^q(v_i), f^q(v_j)\} \ .$$

Consequently a point $x \in \mathcal{P}(\mathcal{G})\backslash\mathcal{V}$ can be a lex location if and only if $f^p(v_i) = f^p(v_j) = f^p(x)$, $1 \leq p \leq Q$ and $v_i, v_j \in \mathcal{X}_{lex}(\mathcal{V})$.

Proposition 3.5. *Let* $x = (e,t) \in \mathcal{P}(\mathcal{G})\backslash\mathcal{V}$, *with* $e = [v_i, v_j]$. *Then* x *is lex location for* $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{lex}$ *if and only if* $f(x) = f(v_i) = f(v_j)$ *and* $v_i, v_j \in \mathcal{X}_{lex}(\mathcal{V})$.

Therefore $\mathcal{X}_{lex}$ can also be found with Algorithm 3.1 in $O(|\mathcal{V}|^3 + |\mathcal{V}|^2 Q^2 \log|\mathcal{V}|)$.

4 Pareto location problems on general networks

4.1 The easy cases: Solving $1/\mathcal{G},\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{par}$ and $1/\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{G})/Q\text{-}\sum_{par}$

In the case where the new facility can be placed only at the nodes of the given network we can determine $\mathcal{X}_{par}(\mathcal{V})$ by the following straightforward approach in $O(Q|\mathcal{V}|^2)$, given the distance matrix.

Algorithm 4.1. *Solving* $1/\mathcal{G},\mathcal{G}_D/\bullet/d(\mathcal{V},\mathcal{V})/Q\text{-}\sum_{par}$

Input: *Network* $\mathcal{N} = (\mathcal{G}, l)$, *Distance Matrix* $D = (d_{ij})$
Output: $\mathcal{X}_{par}(\mathcal{V})$

1. $\mathcal{X}_{par}(\mathcal{V}) := \mathcal{V}$

2. *for* $i := 1$ *to* M *do*

   for $j := 1$ *to* M *do*
   if $f(v_j) < f(v_i)$ *then* $\mathcal{X}_{par}(\mathcal{V}) := \mathcal{X}_{par}(\mathcal{V}) \setminus \{v_i\}$

3. **Output:** $\mathcal{X}_{par}(\mathcal{V})$

In the case of a directed network where the new facility can be placed anywhere on the network the following holds:

Proposition 4.1. *For a directed network* $\mathcal{N} = (\mathcal{G}_D, l)$ *we have* $\mathcal{X}_{par}(\mathcal{V}) = \mathcal{X}_{par}$.

Proof. For $x \in (v_i, v_j)$, with $x \neq v_i$, $i \neq j$, $(v_i, v_j) \in \mathcal{E}$ and $q \in \mathcal{Q}$ we can write in a directed network:

$$f^q(x) - f^q(v_i) = \sum_{k=1}^{M} w_k^q \left(d(x, v_k) - d(v_i, v_k)\right) + \sum_{k=1}^{M} w_k^q \left(d(v_k, x) - d(v_k, v_i)\right)$$

Since the triangle inequality holds and any path to x necessarily contains node v_i,

$$\begin{aligned} \sum_{k=1}^{M} w_k^q \left(d(x, v_k) - d(v_i, v_k)\right) &\geq -\sum_{\substack{k \in \mathcal{M} \\ k \neq i}} w_k^q t l(e) + w_i^q \left((1-t)l(e) + d(v_j, v_i)\right) \\ &= -\sum_{k \in \mathcal{M}} w_k^q t l(e) + w_i^q \left(l(e) + d(v_j, v_i)\right) \end{aligned}$$

and

$$\sum_{k \in \mathcal{M}} w_k^q \left(d(v_k, x) - d(v_k, v_i)\right) = \sum_{k \in \mathcal{M}} w_k^q t l(e) \ .$$

Hence,

$$f^q(x) - f^q(v_i) \geq w_i^q \left(l(e) + d(v_j, v_i)\right) \geq 0 \ .$$

Since $w_i \neq 0$, this means that v_i dominates all $x \in (v_i, v_j)$.

□

Therefore we can restrict ourselves to the node set $\mathcal{V}$, when searching for Pareto locations. Hence we can use Algorithm 4.1 for solving $1/\mathcal{G}_D/\bullet/d(\mathcal{V}, \mathcal{G})/Q\text{-}\sum_{par}$ in $O(Q|\mathcal{V}|^2)$.

4.2 Solving $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/2\text{-}\Sigma_{par}$

What is left, is a solution algorithm for $\mathcal{X}_{par}$ in the undirected case. We first look at the bi-criteria problem to understand the difficulties and identify the differences to the three other easy cases we just saw.

Analogously to the results of Section 4.1, we may hope for some node dominance result or at least that only edges $[v_i, v_j]$ with $v_i, v_j \in \mathcal{X}_{par}(\mathcal{V})$ can contain Pareto locations in their interior. The following example shows that none of those two claims is true in general.

Example 4.1. *Consider the network of Figure 4.1.*

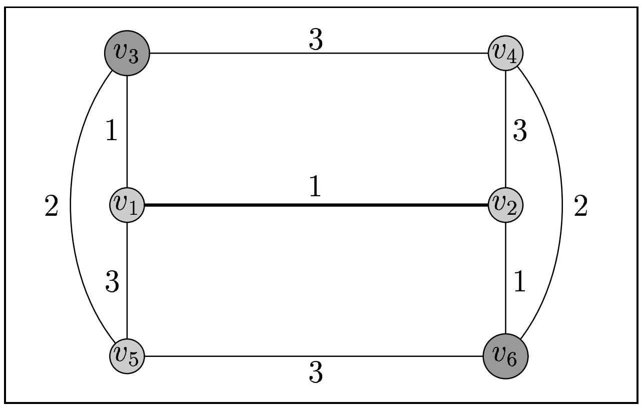

Figure 4.1: The network of Example 4.1. The bold part constitutes the set of Pareto locations.

The distance matrix is given by

$$D = \begin{pmatrix} 0 & 1 & 1 & 4 & 3 & 2 \\ 1 & 0 & 2 & 3 & 4 & 1 \\ 1 & 2 & 0 & 3 & 2 & 3 \\ 4 & 3 & 3 & 0 & 5 & 2 \\ 3 & 4 & 2 & 5 & 0 & 3 \\ 2 & 1 & 3 & 2 & 3 & 0 \end{pmatrix} .$$

With $w^1 := (1,2,1,2,2,2)$ *and* $w^2 := (2,1,2,2,2,1)$ *we get* $f(v_1) = \binom{21}{19}$, $f(v_2) = \binom{19}{21}$, $f(v_3) = \binom{21}{17}$, $f(v_4) = \binom{27}{29}$, $f(v_5) = \binom{29}{27}$, *and* $f(v_6) = \binom{17}{21}$. *So we have*

$$\mathcal{X}_{par}(\mathcal{V}) = \{v_3, v_6\} .$$

If we investigate the edges of $\mathcal{G}$ *we find that by the concavity (see Section 2) of* f^q *on the edges:*

- *v_3 dominates all points on the edges $[v_3, v_5]$, $[v_3, v_4]$ and $[v_3, v_1]$.*
- *v_6 dominates all points on the edges $[v_6, v_2]$, $[v_6, v_5]$ and $[v_6, v_4]$.*
- *v_2 dominates all points on the edge $[v_2, v_4]$.*
- *v_1 dominates all points on the edge $[v_1, v_5]$.*

We also observe that none of the Pareto nodes can dominate a point with both objectives smaller than 21. The objective functions f^1 and f^2 for $[v_1, v_2]$ — the only edge left — are shown in Figure 4.2.

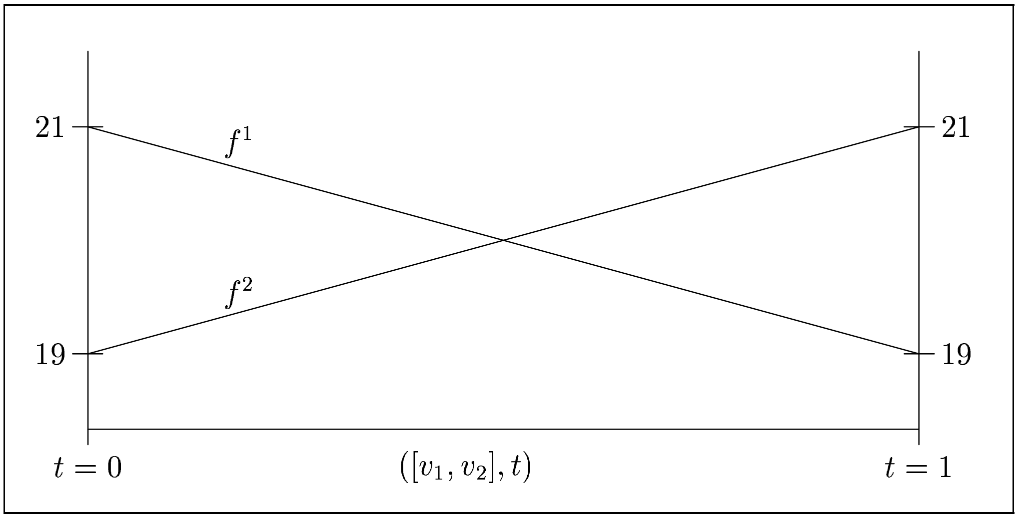

Figure 4.2: The objective functions f^1 and f^2 on $[v_1, v_2]$.

We recognize that

1. *All points $x \in [v_1, v_2]$ with $x \neq v_1$, $x \neq v_2$ have $f^1(x) < 21$ and $f^2(x) < 21$.*
2. *No point $x \in [v_1, v_2]$ dominates a point $y \in [v_1, v_2]$.*

So we have in total

$$\mathcal{X}_{par} = \{v_3, v_6\} \cup ([v_1, v_2], (0, 1)) \ .$$

By changing the weights to $w^1 := (1, 2, 1, 3, 2, 2)$, $w^2 := (2, 1, 2, 2, 3, 1)$ we have as $\mathcal{X}_{par}$ a proper subset of $([v_1, v_2], (0, 1))$ plus $\{v_3, v_6\}$. By further changing the weights to $w^1 := (1, 2, 1, 4, 2, 2)$ and $w^2 := (2, 1, 2, 2, 4, 1)$ we get $\mathcal{X}_{par} = \{v_3, v_6\}$.

4.2.1 Searching for local Pareto locations along the edges

Since situations such as in Example 4.1 may happen, we have to investigate $f^q(x)$ on the interior of the edges. In order to exclude edges or parts of edges from the search, the following results will be useful.

Let

$$\begin{pmatrix} z_{12}^1 \\ z_{12}^2 \end{pmatrix} = f(x_{12}) \ , \text{ with } x_{12} \in \mathcal{X}_{lex}^{1,2}$$

and

$$\begin{pmatrix} z_{21}^1 \\ z_{21}^2 \end{pmatrix} = f(x_{21}) \ , \text{ with } x_{21} \in \mathcal{X}_{lex}^{2,1} \ .$$

Then the following holds:

Theorem 4.2 (Box Theorem). *For all $x \in \mathcal{X}_{par}$*

1. $f^1(x) \in [z_{12}^1, z_{21}^1]$ *and*
2. $f^2(x) \in [z_{21}^2, z_{12}^2]$.

Proof. We only show that $f^1(x) \in [z_{12}^1, z_{21}^1]$. The other part can be proved in the same way. Let $x \in \mathcal{X}_{par}$ with $f(x) = (z^1, z^2)$. First, $z^1 < z_{12}^1$ is impossible, since $f^1(x) \geq z_{12}^1 \ \forall x \in \mathcal{P}(\mathcal{G})$. Second, $z^1 > z_{21}^1$ implies that x is dominated by the lex location x_{21} with $f(x_{21}) = \begin{pmatrix} z_{21}^1 \\ z_{21}^2 \end{pmatrix}$.

$\square$

Lemma 4.3. *If for an edge $[v_i, v_j]$, v_i dominates v_j or v_j dominates v_i, then there is no Pareto location $x \in ([v_i, v_j], (0, 1))$.*

Proof. Without loss of generality let v_i dominate v_j with $f^1(v_i) < f^1(v_j)$ and $f^2(v_i) \leq f^2(v_j)$. Since f^1 and f^2 are concave and f^1 is not constant on $[v_i, v_j]$, we have

$$f^1(v_i) < f^1(x) \text{ and } f^2(v_i) \leq f^2(x)$$

for all $x \in [v_i, v_j]$, with $x \neq v_i$. Therefore v_i dominates x.

$\square$

So, in what follows, we can restrict ourselves to cases where the endnodes of an edge do not dominate each other.

Proposition 4.4. *For an edge $e = [v_i, v_j]$ with v_i and v_j not dominating each other, let f^1 and f^2 be non constant on e, let*

$$f^1(v_i) > f^1(v_j) \text{ and } f^2(v_i) < f^2(v_j) \ .$$

and let

$$t^1 := \max\left\{t \in [0,1] \ : \ f^1(v_i) = f^1\left((e,t)\right)\right\}$$

and

$$t^2 := \min\left\{t \in [0,1] \ : \ f^2(v_j) = f^2\left((e,t)\right)\right\} \ .$$

Then

$$\mathcal{X}_{par}\left([v_i, v_j]\right) = \{v_i\} \cup \{v_j\} \cup \left(e, (t^1, t^2)\right) \ .$$

Proof. The endnodes v_i and v_j do not dominate each other and, by the concavity of f^1 and f^2, there cannot be a point in $([v_i, v_j], (0,1))$ dominating v_i or v_j. Also by the concavity of the objective functions on $[v_i, v_j]$ we know that f^1 is decreasing and f^2 is increasing in $[t^1, t^2]$. Therefore no point $y' = (e, t')$, with $t' \in [t^1, t^2]$ can dominate another point $y'' = (e, t'')$, with $t'' \in [t^1, t^2]$. The points (e,t) with $t \in (0, t^1)$ are dominated by node v_i (by definition of t^1). The points (e,t) with $t \in (t^2, 1)$ are dominated by node v_j (by definition of t^2). Also by definition of t^1 and t^2, no point (e,t) with $t \in [t^1, t^2]$ is dominated by v_i or v_j.

$\square$

Remark that we do not use the piecewise linearity property of f^q in the proof of Proposition 4.4, so that this result can be extended to any objective function which is concave in the distances.

Lemma 4.5. *For an edge $e = [v_i, v_j]$ with v_i and v_j not dominating each other, let f^1 or f^2 be constant on e and let*

$$f^1(v_i) \geq f^1(v_j) \text{ and } f^2(v_i) \leq f^2(v_j) \ .$$

Then $f(v_i) = f(v_j)$ and

- *if only one objective function is constant, then*

$$\mathcal{X}_{par}\left([v_i, v_j]\right) = \{v_i, v_j\} \ .$$

- *If both objective functions are constant, then*

$$\mathcal{X}_{par}\left([v_i, v_j]\right) = [v_i, v_j] \,.$$

Proof. Assume that f^1 is constant. Then $f^1(v_i) = f^1(v_j)$. If $f^2(v_i) > f^2(v_j)$ then v_j dominates v_i. If $f^2(v_i) < f^2(v_j)$ then v_i dominates v_j. So $f(v_i) = f(v_j)$.

□

Corollary 4.6. *For every edge $e = [v_i, v_j]$ in the network, $\mathcal{X}_{par}([v_i, v_j])$ is a (possibly empty) single subedge of $[v_i, v_j]$ plus one or both endnodes.*

Now we can combine the results of that section to an algorithm for finding $\mathcal{X}_{par}([v_i, v_j])$.

Algorithm 4.2. <u>*Determining* $\mathcal{X}_{par}([v_i, v_j])$</u>

Input: *Edge* $e = [v_i, v_j]$.
Output: $\mathcal{X}_{par}([v_i, v_j])$

1. *if v_i dominates v_j then $\mathcal{X}_{par}([v_i, v_j]) := \{v_i\}$, stop.*
2. *if v_j dominates v_i then $\mathcal{X}_{par}([v_i, v_j]) := \{v_j\}$, stop.*
3. *if $f(v_i) = f(v_j)$ then*
   (a) *if $f((e, \frac{1}{2})) = f(v_i)$ then $\mathcal{X}_{par}([v_i, v_j]) := [v_i, v_j]$, stop.*
   (b) *if $f((e, \frac{1}{2})) \neq f(v_i)$ then $\mathcal{X}_{par}([v_i, v_j]) := \{v_i, v_j\}$, stop.*
4. *if $f^1(v_i) \leq f^1(v_j)$ and $f^2(v_i) \geq f^2(v_j)$ then exchange v_i and v_j*
5. *Compute t^1 and t^2 as defined in Proposition 4.4.*
6. *if $t^1 < t^2$*
   then $\mathcal{X}_{par}([v_i, v_j]) := \{v_i, v_j\} \cup (e, (t^1, t^2))$
   else $\mathcal{X}_{par}([v_i, v_j]) := \{v_i, v_j\}$.
7. **Output:**$\mathcal{X}_{par}([v_i, v_j])$.

On an edge e, the function f^q is piecewise linear with at most $|\mathcal{V}|$ breakpoints corresponding to the bottleneck points on this edge. Those, as well as their value for f^q can be computed in $O(|\mathcal{V}| \log |\mathcal{V}|)$, see e.g. [Hansen et al., 1991]. Then t^1 and t^2 can be determined by exploring the sorted list of bottleneck points once. So, Algorithm 4.2 runs in $O(|\mathcal{V}| \log |\mathcal{V}|)$.

Therefore we can compute $\bigcup_{[v_i,v_j]\in\mathcal{E}} \mathcal{X}_{par}([v_i, v_j])$ in $O(|\mathcal{E}||\mathcal{V}| \log |\mathcal{V}|)$. Note that in implementations of Algorithm 4.2, Theorem 4.2 can be used to further reduce the number of candidates for $\mathcal{X}_{par}$.

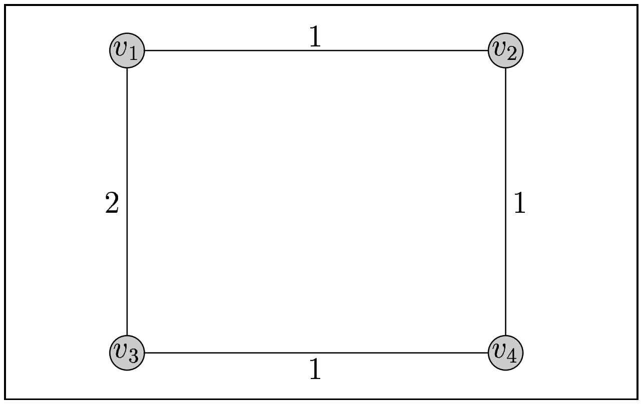

Figure 4.3: Network for Example 4.2.

Example 4.2. *Consider the network of Figure 4.3.*
The distance matrix is given by

$$D = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 \\ 2 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} .$$

With $w^1 := (1, 2, 2, 2)$ *and* $w^2 := (3, 1, 2, 1)$ *we get* $f(v_1) = \binom{10}{7}$, $f(v_2) = \binom{7}{8}$, $f(v_3) = \binom{8}{9}$, *and* $f(v_4) = \binom{6}{9}$. *So we have*

$$\mathcal{X}_{par}(\mathcal{V}) = \{v_1, v_2, v_4\} .$$

Since the optimal solutions for the single objective problems are unique we get $z^1_{12} = 6$, $z^2_{12} = 9$, $z^1_{21} = 10$ *and* $z^2_{21} = 7$. *This means that Theorem 4.2 does not allow to exclude edges. So we use Algorithm 4.2 for each edge.*

- *Edge* $[v_1, v_2]$. *None of the vertices* v_1 *and* v_2 *dominates the other and the objective functions are not constant, so we use Proposition 4.4 to determine* $\mathcal{X}_{par}([v_1, v_2])$. *We get* $t^1 = 0$ *and* $t^2 = \frac{1}{3}$ *(see Figure 4.4). Therefore* $\mathcal{X}_{par}([v_1, v_2]) = \{v_1, v_2\} \cup \left([v_1, v_2], (0, \frac{1}{3})\right)$.

- *Edge* $[v_2, v_4]$. *Since there are no bottleneck points on the edge and one function is strictly increasing while the other one is strictly decreasing we get that* $\mathcal{X}_{par}([v_2, v_4]) = [v_2, v_4]$.

- *Edge* $[v_3, v_4]$. *Since* v_4 *dominates* v_3 *we have* $\mathcal{X}_{par}([v_3, v_4]) = \{v_4\}$.

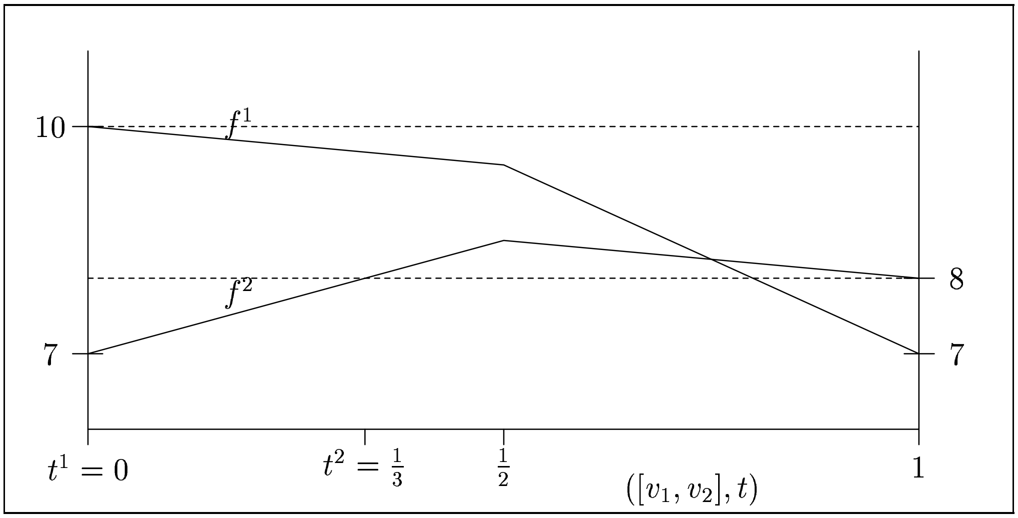

Figure 4.4: Illustration for Example 4.2. The determination of $\mathcal{X}_{par}([v_1, v_2])$.

- *Edge* $[v_1, v_3]$. v_1 *and* v_3 *do not dominate each other and the objective functions are not constant, so we use again Proposition 4.4 to determine* $\mathcal{X}_{par}([v_1, v_3])$. *We get* $t^2 = \frac{1}{2} < t^1 = \frac{4}{5}$ *(see Figure 4.5). Therefore* $\mathcal{X}_{par}([v_1, v_3]) = \{v_1, v_3\}$.

Let $e = [v_i, v_j]$ be an edge such that $f(v_i) \neq f(v_j)$ and set $z^1(t) = f^1(x_t)$ and $z^2(t) = f^2(x_t)$ for $x_t = (e,t) \in \mathcal{X}_{par}([v_i, v_j]) \cap (e, (0,1))$. Both functions are piecewise linear with the same set of possible breakpoints corresponding to bottleneck points. To simplify denotation we assume without loss of generality that f^1 and f^2 have the same set of breakpoints on $[v_i, v_j]$. (If this is not the case the union of the breakpoints has to be considered.) The breakpoints are denoted by t_j, $j = 0, \ldots, P$ with $(e, t_o) = v_i$, $(e, t_P) = v_j$ and $t_{j-1} < t_j$ for all $j = 1, \ldots, P$. For $t \in [t_{j-1}, t_j]$, $z^1(t)$ and $z^2(t)$ are therefore linear functions of the form, say,

$$z^1(t) = m_j^1 t + b_j^1 \text{ and } z^2(t) = m_j^2 t + b_j^2 \ .$$

Hence, since f^1 is not constant, we may write z^2 in terms of z^1. For each segment $[t_{j-1}, t_j]$, $1 \leq j \leq P$, we get

$$z^2 = \frac{m_j^2}{m_j^1} z^1 + b_j^2 - \frac{m_j^2 b_j^1}{m_j^1} \ .$$

Now, assume without loss of generality that $f^1(v_i) \leq f^1(v_j)$ and $f^2(v_i) \geq f^2(v_j)$. Since the functions f^1 and f^2 are concave and piecewise linear on

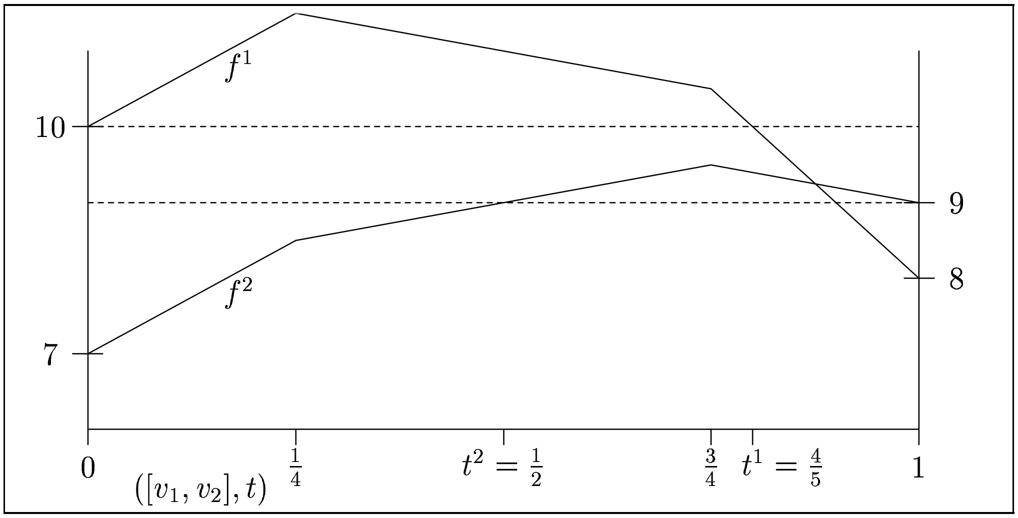

Figure 4.5: Illustration for Example 4.2. The determination of $\mathcal{X}_{par}([v_1, v_3])$

$[v_i, v_j]$ and since they are strictly monotone on $\mathcal{X}_{par}([v_i, v_j])$ (see Figure 4.6), we have that

$$m_1^1 > m_2^1 > \ldots > m_p^1 > 0$$

and

$$0 > m_1^2 > m_2^2 > \ldots > m_p^2$$

so that

$$0 > \frac{m_1^2}{m_1^1} > \frac{m_2^2}{m_2^1} > \ldots > \frac{m_p^2}{m_p^1},$$

which means that z^2 is a concave function of z^1 when $f(v_i) \neq f(v_j)$. According to Lemma 4.5, $\mathcal{X}_{par}([v_i, v_j])$ contains points in $([v_i, v_j], (0, 1))$ either if $f(v_i) \neq f(v_j)$ or if $f(x_t) = f(v_i) = f(v_j)$, for all $t \in (0, 1)$. In the latter case, the image of $[v_i, v_j]$ in the objective space reduces to a single point (z^1, z^2), which can be considered as a degenerate concave curve. We have thus proved the following lemma.

Lemma 4.7. *For $e = [v_i, v_j]$ with $\mathcal{X}_{par}([v_i, v_j]) \cap ([v_i, v_j], (0, 1)) \neq \emptyset$, z^2 is a piecewise linear and concave function in z^1.*

In order to obtain $\mathcal{X}_{par}$ we can draw $IM(f)$, the set of all images of all $\mathcal{X}_{par}([v_i, v_j])$, $e = [v_i, v_j] \in \mathcal{E}$, in the objective space. These images consist of **connected components**, i.e. isolated points or piecewise concave linear curves. Since, in the objective space, a point z dominates all other points in $z + \mathbb{R}_+^2 := \{z + y \ : \ y \in \mathbb{R}_+^2\}$ we extend all connected components by a horizontal line segment at its right-most point.

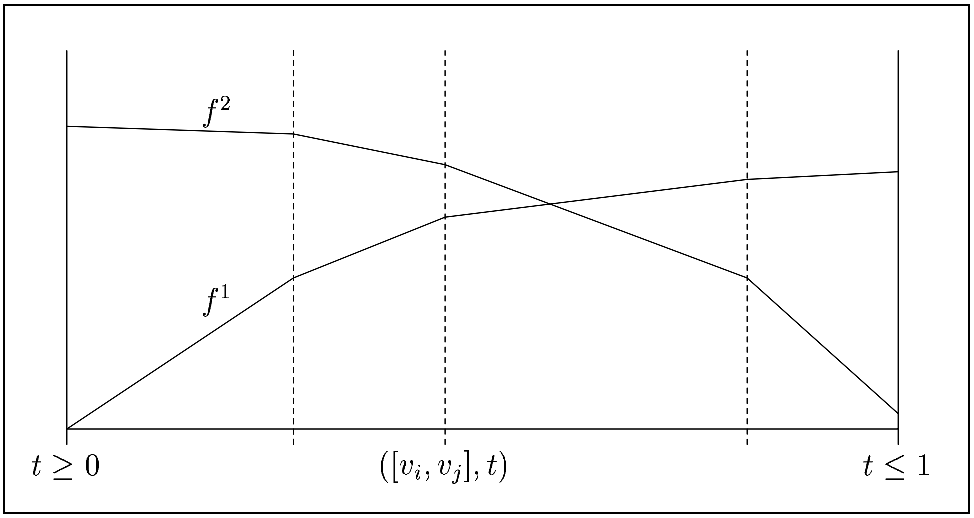

Figure 4.6: The objective functions f^1 and f^2 on a part of an edge producing local Pareto locations. Both functions are piecewise linear with the same breakpoints.

Algorithm 4.3. *Combining the local Pareto locations*

Input: $\mathcal{X}_{par}([v_i, v_j])$ *for all* $e = [v_i, v_j] \in \mathcal{E}$.
Output: $\mathcal{X}_{par}$

1. *Let* $z^1_{max} := \max\left\{ f^1(x) \ : \ x \in \bigcup_{[v_i,v_j]\in\mathcal{E}} \mathcal{X}_{par}([v_i, v_j]) \right\}.$

2. *Build* $IM(f)$*, which consists of* $\bigcup_{[v_i,v_j]\in\mathcal{E}} f(\mathcal{X}_{par}([v_i, v_j]))$.

3. *For each connected component* l *in* $IM(f)$*, let* (z^1_l, z^2_l) *be the right-most point and add to* $IM(f)$ *the horizontal segment* l' *with left-most point* (z^1_l, z^2_l) *and right-most point* (z^1_{max}, z^2_l).

4. *Compute the lower envelope* L *of* $IM(f)$*, which is the lower envelope of* $O(|\mathcal{E}||\mathcal{V}|)$ *line segments.*

5. *Eliminate every horizontal line segment of* L*, except its left-most point.*

6. **Output:** $\mathcal{X}_{par} = f^{-1}(L)$.

Step 1-3 are necessary to get really rid of all dominated points when applying the lower-envelope algorithm. Note that these steps can be done in linear time and that the order of the number of line segments is not affected. Step 4 of Algorithm 4.3 can be done in $O(|\mathcal{E}||\mathcal{V}|\log\max(|\mathcal{E}|,|\mathcal{V}|))$ (see [Hershberger, 1989]) and this equals the complexity of the whole algorithm, since the clean-up Step 5 can be done in linear time.

Note that we may take instead of an open subedge $(e,(t_1,t_2))$, the corresponding closed subedge $(e,[t_1,t_2])$ without affecting the result of Algorithm 4.3.

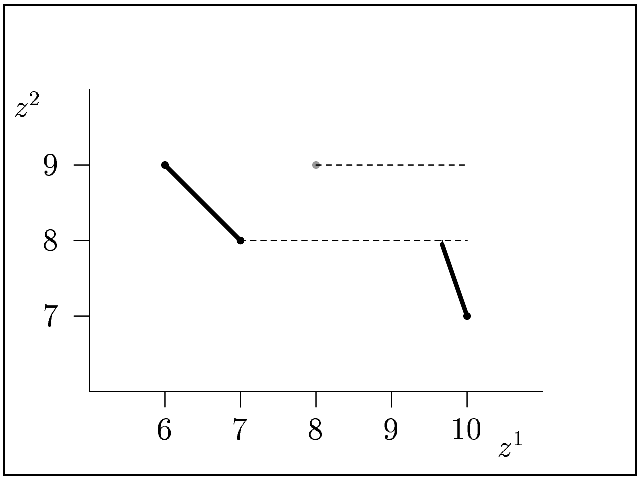

Figure 4.7: Illustration for Example 4.3. The bold part constitutes $\mathcal{X}_{par}$.

Example 4.3. *We use the results of Example 4.2 as input for Algorithm 4.3 and get*

$$\mathcal{X}_{par} = [v_2, v_4] \cup \left([v_1, v_2]\ ,\ [0, \frac{1}{3})\right)$$

(see Figure 4.7).

4.3 The Q-criteria case

We now turn to the general case of Q criteria. While the determination of $\mathcal{X}_{par}([v_i, v_j])$ is similar to the bi-criteria case the process of finding $\mathcal{X}_{par}$ turns out to be very much different.

Let $e = [v_i, v_j]$ be an edge in the network. First notice that Lemma 4.3 holds also for the Q-criteria case and can be proved just by replacing f^1 and f^2 by $f^{q'}$ and $f^{q''}$, $(q' \neq q'',\ q', q'' \in \mathcal{Q})$, respectively.

From now on assume that neither v_i dominates v_j nor v_j dominates v_i. Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be a partition of the set of criteria such that $\mathcal{Q}_1 \subset \mathcal{Q}$ with

$f^q(v_i) \geq f^q(v_j)$, $\mathcal{Q}_2 \subset \mathcal{Q}$ with $f^q(v_i) < f^q(v_j)$. Of course, $\mathcal{Q}_1 \neq \emptyset$, $\mathcal{Q}_1 \cap \mathcal{Q}_2 = \emptyset$ and $\mathcal{Q}_1 \cup \mathcal{Q}_2 = \mathcal{Q}$.

The following lemma is analogous to Lemma 4.5.

Lemma 4.8. *Let the functions f^q be constant on $e = [v_i, v_j]$ for all $q \in \mathcal{Q}_1$.*

- *If $\mathcal{Q}_2 \neq \emptyset$, then $\mathcal{X}_{par}([v_i, v_j]) = \{v_i, v_j\}$.*
- *If $\mathcal{Q}_2 = \emptyset$, then $\mathcal{X}_{par}([v_i, v_j]) = [v_i, v_j]$.*

Proposition 4.4 can also be adapted to the general case:

Proposition 4.9. *Let $f(v_i) \neq f(v_j)$ for an edge $e = [v_i, v_j]$. Further let*

$$t^1(f^q) := \max\{t \in [0,1] \; : \; f^q(v_i) = f^q((e,t))\} \quad \textit{for all } q \in \mathcal{Q}_1$$

and

$$t^2(f^q) := \min\{t \in [0,1] \; : \; f^q(v_j) = f^q((e,t))\} \quad \textit{for all } q \in \mathcal{Q}_2 \, .$$

Then

$$\mathcal{X}_{par}([v_i, v_j]) = \{v_i\} \cup \{v_j\} \cup \left(e, \left(\min_{q \in \mathcal{Q}_1} \left\{t^1(f^q)\right\}, \max_{q \in \mathcal{Q}_2} \left\{t^2(f^q)\right\}\right)\right) .$$

From these result we can conclude that Corollary 4.6 also holds for the Q-criteria case, i. e. $\mathcal{X}_{par}([v_i, v_j])$ is a (possibly empty) single subedge of $[v_i, v_j]$ plus one or both endnodes. This helps very much to decrease the expected amount of storage in implementations.

In order to determine $\mathcal{X}_{par}$, it remains to compare pairwise all elements of $\mathcal{X}_{par}([v_i, v_j])$, for all $[v_i, v_j] \in \mathcal{E}$.

First compare all vertices $v_i \in \mathcal{V}$ and edges where all f^q, $q \in \mathcal{Q}$ are constant. This has a complexity of $O(|\mathcal{E}|^2 Q + |\mathcal{V}|^2 Q + |\mathcal{E}||\mathcal{V}|Q)$. After that we can restrict ourselves to cases where at least one f^q is non-constant.

Consider a subedge $(e_l, [t_r, t_{r+1}])$, of $e_l \in \mathcal{E}$, with $(e_l, [t_r, t_{r+1}]) \subseteq \mathcal{X}_{par}(e_l)$, with t_r, t_{r+1} consecutive breakpoints on e_l. We know that $(e_l, [t_r, t_{r+1}])$ is a **linear subedge** (with respect to $f^1, \ldots, f^Q$), i. e.

$$f^q((e_l, t)) = b_r^q + m_r^q t \;\; \text{for all } q \in \mathcal{Q} \, , \quad t \in [t_r, t_{r+1}] \, .$$

We want to find out whether any of the points (e_l, t), $t \in [t_r, t_{r+1}]$, is dominated by some $x \in \bigcup_{e \in \mathcal{E}} \mathcal{X}_{par}(e)$.

First, we consider a different linear subedge $(e_k, [s_p, s_{p+1}]) \subseteq \mathcal{X}_{par}(e_k)$, where s_p, s_{p+1} are consecutive breakpoints on e_k, i. e.

$$f^q((e_k, s)) = b_p^q + m_p^q s \;\; \text{for all } q \in \mathcal{Q} \, , \quad s \in [s_p, s_{p+1}] \, .$$

A point $(e_l, t) \in (e_l, [t_r, t_{r+1}])$ is dominated by some point $(e_k, s) \in (e_k, [s_p, s_{p+1}])$, by definition, if and only if

$$b_p^q + m_p^q s \leq b_r^q + m_r^q t \text{ for all } q \in \mathcal{Q} ,$$

where at least one inequality is strict. If

$$\mathcal{F} := \left\{(s,t) \ : \ m_r^q t - m_p^q s \geq b_p^q - b_r^q \ , \ \forall q \in \mathcal{Q}\right\} \cap ([s_p, s_{p+1}] \times [t_r, t_{r+1}])$$

is empty, $(e_k, [s_p, s_{p+1}])$ does not contain a point dominating some $x \in (e_l, [t_r, t_{r+1}])$.

Otherwise $\mathcal{F} \neq \emptyset$ is taken as a feasible solution set of two 2-variable linear programs:

$$LB = \min \{t \ : \ (s,t) \in \mathcal{F}\}$$

and

$$UB = \max \{t \ : \ (s,t) \in \mathcal{F}\} \ .$$

Using [Megiddo, 1982] , LB and UB can be computed in $O(Q)$ time. Let s_{LB} and s_{UB} be optimal values for s corresponding to LB and UB, respectively.

If $b_p^q + m_p^q s_{LB} = b_r^q + m_r^q LB$ and $b_p^q + m_p^q s_{UB} = b_r^q + m_r^q UB$, then no $x \in (e_l, [t_r, t_{r+1}])$ is dominated by any $y \in (e_k, [s_p, s_{p+1}])$. Otherwise redefine $\mathcal{X}_{par}(e) = \mathcal{X}_{par}(e) \setminus [LB, UB]$. This comparison can also be done in linear time.

It should be noted that the same approach works if $(e_l, [t_r, t_{r+1}])$ is just a single point $x = (e_l, t')$. In this case $LB = UB = t'$ and

$$\mathcal{F}' := \left\{s \ : \ -m_p^q s \geq b_p^q - f^q(x) \ , \ \forall q \in \mathcal{Q}\right\} \cap [s_p, s_{p+1}] \ .$$

The case where $(e_k, [s_p, s_{p+1}])$ is just a single point y is treated correspondingly by using

$$\mathcal{F}'' := \{t \ : \ m_r^q t \geq f^q(y) - b_r^q \ , \ \forall q \in \mathcal{Q}\} \cap [t_r, t_{r+1}] \ ,$$

$LB := \min\{t \ : \ t \in \mathcal{F}''\}$, and $UB := \max\{t \ : \ t \in \mathcal{F}''\}$.

Since we have $O(|\mathcal{V}|)$ segments per edge to investigate and $|\mathcal{E}|$ edges we have to do $O(|\mathcal{E}|^2|\mathcal{V}|^2)$ comparisons each taking $O(Q)$ time. Therefore the complexity for finding the whole set $\mathcal{X}_{par}$ requires $O(|\mathcal{E}|^2|\mathcal{V}|^2 Q)$ time, which is the same complexity [Hansen et al., 1986] derived for the special case of point-objective problems.

5 Solving $1/\mathcal{T}/\bullet/d(\mathcal{V},\mathcal{T})/Q\text{-}\Sigma_{par}$

A lot of difficult problems on general networks become quite easy to solve if the underlying graph has a tree structure. We will show in the following that this is also true for the multicriteria problems. We will also relate our results with the work which has previously been done on trees and end up with a generalization of Goldman's algorithm (see [Goldman, 1971]).

The major concept which makes things easier on trees is convexity. We first introduce this concept based on [Dearing et al., 1976].

Let $\mathcal{N} = (\mathcal{T}, l)$ be a tree network, with $\mathcal{T} = (\mathcal{V}, \mathcal{E})$. For points $a, b \in \mathcal{P}(\mathcal{T})$ we define the **line segment** $L[a,b]$ between a and b as

$$L[a,b] := \{x \in \mathcal{P}(\mathcal{T}) | d(a,x) + d(x,b) = d(a,b)\} \ ,$$

which contains all points on the unique path between a and b. A subset $C \subseteq \mathcal{P}(\mathcal{T})$ is called **convex**, iff, for all $a, b \in C$, $L[a,b] \subseteq C$.

Now let $C \subseteq \mathcal{P}(\mathcal{T})$ be convex and let $h : \mathcal{P}(\mathcal{T}) \to \mathbb{R}$ be a real valued function. This function h is called **convex on** C, iff, for all $a, b \in C$,

$$h(x_\lambda) \leq \lambda h(a) + (1-\lambda) h(b) \ , \ \forall \lambda \in [0,1] \ ,$$

where x_λ is uniquely defined by

$$d(x_\lambda, b) = \lambda d(a,b) \text{ and } d(x_\lambda, a) = (1-\lambda) d(a,b) \ . \tag{5.1}$$

A function is called convex on $\mathcal{T}$ if it is convex on $C = \mathcal{P}(\mathcal{T})$.

Now we state some results about convex functions on trees which we will need in the following. Proofs can be found in [Dearing et al., 1976].

Lemma 5.1. *Let $c \in \mathcal{P}(\mathcal{T})$. Then $h(x) := d(x,c)$ is convex on $\mathcal{T}$.*

Note that it is possible to define the concept of convexity also on general networks. Then one can show that $d(x,c)$ is convex if and only if the underlying graph is a tree.

Lemma 5.2. *Let $C \subseteq \mathcal{P}(\mathcal{T})$ be a convex set and h, h_1, h_2 be real valued convex functions on C. Then the following holds:*

1. *If h is convex on C then wh is convex on C, for all $w \geq 0$.*

2. *If h_1, h_2 are convex on C then $h_1 + h_2$ is convex on C.*

3. *If x^* is a local minimizer of h then x^* is a global minimizer of h.*

4. *If $x^*, y^* \in \operatorname{argmin}_{x\in C} h(x)$ then $L[x^*, y^*] \subseteq \operatorname{argmin}_{x\in C} h(x)$.*

Corollary 5.3. *The single criterion median objective function f^q is convex on $\mathcal{T}$, for all $q \in \mathcal{Q}$.*

Now let $L(a,b) := L[a,b]\backslash\{a,b\}$, $L(a,b] := L[a,b]\backslash\{a\}$ and $L[a,b) := L[a,b]\backslash\{b\}$. Further denote by $\mathcal{X}^*(h^q)$ the set of minimizers for objective function h^q. The following theorem generalizes part 4 of Lemma 5.2 to multiple criteria.

Theorem 5.4. *Let $a, b \in \mathcal{P}(\mathcal{T})$ and $h := (h^1, \ldots, h^Q)$ be a vector of Q objective functions, with h^q convex on T, for all $q \in \mathcal{Q}$.Then the following holds*

$$\{a,b\} \subseteq \mathcal{X}_{par} \text{ iff } L[a,b] \subseteq \mathcal{X}_{par} \ .$$

Proof. If

$$\bigcap_{q\in\mathcal{Q}} \mathcal{X}^*(h^q) \neq \emptyset$$

then

$$\mathcal{X}_{par} = \bigcap_{q\in\mathcal{Q}} \mathcal{X}^*(h^q)$$

and the claim is true by the convexity of the objective functions and Lemma 5.2.

Now assume

$$\bigcap_{q\in\mathcal{Q}} \mathcal{X}^*(h^q) = \emptyset \ .$$

Further let $a, b \in \mathcal{X}_{par}$ and let $x_\lambda \in L[a,b]$ be defined as in (5.1). Suppose there exists a point $y \in \mathcal{P}(\mathcal{T})$, which dominates x_λ. Assume first that $h(a) = h(b)$. Then, since y dominates x_λ and given the convexity of h^q, $q \in \mathcal{Q}$, we have

$$h^q(y) \leq h^q(x_\lambda) \leq \lambda h^q(a) + (1-\lambda)h^q(b) = h^q(a) = h^q(b)$$

for all $q \in \mathcal{Q}$ and

$$h^{q'}(y) < h^{q'}(x_\lambda) \leq \lambda h^{q'}(a) + (1-\lambda)h^{q'}(b) = h^q(a) = h^q(b)$$

for at least one $q' \in \mathcal{Q}$. Therefore, y dominates a and b, a contradiction to $a, b \in \mathcal{X}_{par}$.

Now, if $h(a) \neq h(b)$ let $\mathcal{T}_a$, $\mathcal{T}_b$ and $\mathcal{T}_c$ be subtrees of $\mathcal{T}$, which are uniquely defined by the following conditions

$$\mathcal{P}(\mathcal{T}_a) \cup \mathcal{P}(\mathcal{T}_b) \cup \mathcal{P}(\mathcal{T}_c) = \mathcal{P}(\mathcal{T}) \ , \ \mathcal{P}(\mathcal{T}_a) \cap \mathcal{P}(\mathcal{T}_b) \cap L(a,b) = \emptyset \ ,$$

$a \in \mathcal{P}(\mathcal{T}_a)$, $b \in \mathcal{P}(\mathcal{T}_b)$ and $L(a,b) \subseteq \mathcal{P}(\mathcal{T}_c)$. Since a is a Pareto location there must be some $q_a \in \mathcal{Q}$ with $\mathcal{X}^*(h^{q_a}) \subseteq \mathcal{P}(\mathcal{T}_a)$. Otherwise we could move from a to b not increasing any objective function and decreasing at least one, since

$$\bigcap_{q \in \mathcal{Q}} \mathcal{X}^*(h^q) = \emptyset \,.$$

Therefore we could find a point $x' \in L(a,b)$ dominating a, which would be a contradiction. Analogously, there must be a $q_b \in \mathcal{Q}$, $j \neq i$ with $\mathcal{X}^*(h^{q_b}) \subseteq \mathcal{P}(\mathcal{T}_b)$.
So we have q_a, q_b with $h^{q_a}(a) < h^{q_a}(b)$, $h^{q_b}(a) > h^{q_b}(b)$,

$$\mathcal{X}^*(h^{q_a}) \cap L(a,b) = \emptyset \ \text{ and } \ \mathcal{X}^*(h^{q_b}) \cap L(a,b) = \emptyset \,. \tag{5.2}$$

Denote by $\mathcal{T}^a_{x_\lambda}$ the subtree of $\mathcal{T}$ consisting of all $x \in \mathcal{P}(\mathcal{T})$, with $x_\lambda \in L[a,x]$. Analogously, denote by $\mathcal{T}^b_{x_\lambda}$ the subtree of $\mathcal{T}$ consisting of all $x \in \mathcal{P}(\mathcal{T})$, with $x_\lambda \in L[b,x]$. Clearly, by (5.2), and the convexity of the objective functions h^{q_a} and h^{q_b}, we have

$$h^{q_a}(x) \geq h^{q_a}(x_\lambda) \quad \text{for all } \ x \in \mathcal{P}(\mathcal{T}^a_{x_\lambda}) \tag{5.3}$$

and

$$h^{q_b}(x) \geq h^{q_b}(x_\lambda) \quad \text{for all } \ x \in \mathcal{P}(\mathcal{T}^b_{x_\lambda}) \,. \tag{5.4}$$

Combining (5.3) and (5.4), a $y \in \mathcal{P}(\mathcal{T})$ which dominates x_λ has to fulfill

$$y \notin \mathcal{P}(\mathcal{T}^a_{x_\lambda}) \ \text{ and } \ y \notin \mathcal{P}(\mathcal{T}^b_{x_\lambda}) \,,$$

which is not possible since $\mathcal{P}(\mathcal{T}^a_{x_\lambda}) \cup \mathcal{P}(\mathcal{T}^b_{x_\lambda}) = \mathcal{P}(\mathcal{T})$.

□

We know that every lex location is also a Pareto location. From the proof of Theorem 5.4 we can easily see that if a point x is not on a path between two lex locations, this x cannot be Pareto. Therefore we have the following corollary.

Corollary 5.5. *$\mathcal{X}_{par}$ is the convex set in $\mathcal{P}(\mathcal{T})$ generated by all lex locations with respect to $h^1, \ldots, h^Q$.*

The result of Theorem 5.4 is almost equivalent to a characterization given in [Lowe, 1978]. The proof presented here is however shorter and simpler.

We could now use Corollary 5.5 together with Goldman's algorithm ([Goldman, 1971]) and the results about lex locations to solve $1/\mathcal{T}/\bullet/d(\mathcal{V},\mathcal{T})/Q\text{-}\sum_{par}$, but there is a more efficient way.

For $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{V}' \subseteq \mathcal{V}$ let

$$W(\mathcal{V}') := \begin{pmatrix} w^1(\mathcal{V}') \\ w^2(\mathcal{V}') \\ \vdots \\ w^Q(\mathcal{V}') \end{pmatrix},$$

where $w^q(\mathcal{V}') := \sum_{v_i \in \mathcal{V}'} w_i^q$, $\forall q \in \mathcal{Q}$.

Proposition 5.6. *Let $\mathcal{T}$ be partitioned in such a way that $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \{e\}$. Then $W(\mathcal{V}(\mathcal{T}_1))$ dominates $W(\mathcal{V}(\mathcal{T}_2))$ if and only if for all $x \in \mathcal{P}(\mathcal{T}_1)$ there exists some $y \in \mathcal{P}(\mathcal{T}_2)$ which dominates x.*

Proof. Apply Lemmas 1 and 2 in [Goldman, 1971] to every objective functions and use the fact that one inequality has to be strict.

□

Now we can state a multi-criteria version of Goldman's dominance algorithm. We start with a subtree containing only a leaf of the tree and enlarge this subtree until we get a Pareto location by the criterion of Proposition 5.6. This procedure is then repeated for all leaves and we end up with a subtree of all Pareto locations by using Theorem 5.4.

Algorithm 5.1. *Solving $1/\mathcal{T}/\bullet/d(\mathcal{V},\mathcal{T})/Q\text{-}\sum_{par}$*

Input: *$\mathcal{T} = (\mathcal{V}, \mathcal{E})$, with length function l and node weight vectors w^q, $q \in \mathcal{Q}$.*
Output: $\mathcal{X}_{par}$

1. *Initialization*

   (a) $\mathcal{T}' = (\mathcal{V}', \mathcal{E}') := \mathcal{T}$

   (b) $W := W(\mathcal{V})$

   (c) *set the status of all $v_i \in \mathcal{V}$ to* **not investigated**

2. *Choose a leaf v_k of $\mathcal{T}'$ with status* **not investigated** *and set the status of v_k to* **investigated**.

3. *If $\mathcal{V}' = \{v_k\}$ then go to Step 6.*

4. *Let v_l be the (uniquely defined) adjacent node to v_k.*
   If $\begin{pmatrix} w_k^1 \\ \vdots \\ w_k^Q \end{pmatrix}$ dominates $\frac{1}{2}W$ then

(a) $w_l^q := w_l^q + w_k^q$, $q \in \mathcal{Q}$

(b) $\mathcal{V}' := \mathcal{V}' \setminus \{v_k\}$, $\mathcal{E}' := \mathcal{E}' \setminus [v_k, v_l]$

(c) *Update the set of leaves.*

5. *If there are still leaves in* $\mathcal{T}'$ *with status* **not investigated** *then goto Step* 2.

6. **Output:** $\mathcal{X}_{par} = \mathcal{P}(\mathcal{T}')$.

The algorithm runs in $O(Q|\mathcal{V}|)$.
To illustrate the algorithm consider the following example:

Example 5.1. *(see also Figure 5.1)*

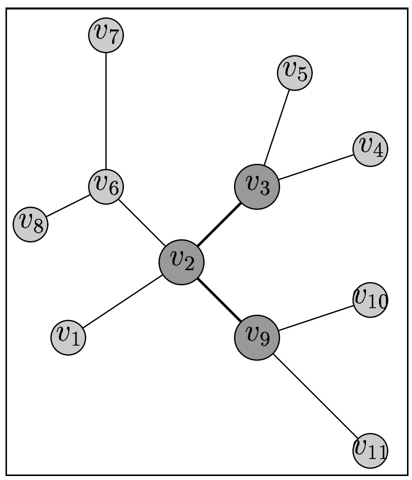

Figure 5.1: Illustration for Example 5.1. The bold edges and nodes indicate the set of Pareto locations.

We solve the following instance of $1/\mathcal{T}/\bullet/d(\mathcal{V},\mathcal{T})/3\text{-}\sum_{par}$. *Let* $l(e) := 1$, $\forall e \in \mathcal{E}$. *The weights of the nodes are given in the following table:*

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
w^1	14	6	8	4	1	2	1	3	2	2	7
w^2	11	3	3	24	5	2	2	3	2	2	5
w^3	16	2	1	1	2	3	3	1	6	4	21

Therefore $W = \begin{pmatrix} 50 \\ 62 \\ 60 \end{pmatrix}$ *and* $\frac{1}{2}W = \begin{pmatrix} 25 \\ 31 \\ 30 \end{pmatrix}$.

The adjacency structure of the tree is given in Figure 5.1. Now we check every leaf till there is no one left with status not investigated*:*

- *take* v_1*:* $w_1 = \begin{pmatrix} 14 \\ 11 \\ 16 \end{pmatrix}$ *dominates* $\frac{W}{2} = \begin{pmatrix} 25 \\ 31 \\ 30 \end{pmatrix}$.
  Therefore $w_2 := \begin{pmatrix} 6+14 \\ 3+11 \\ 2+16 \end{pmatrix} = \begin{pmatrix} 20 \\ 14 \\ 18 \end{pmatrix}$.

By following the algorithm we delete v_8, v_7, v_6, v_5, *and* v_4. *The actual value of* w_3 *is* $\begin{pmatrix} 13 \\ 32 \\ 4 \end{pmatrix}$.

- *take* v_3*:* $w_3 = \begin{pmatrix} 13 \\ 32 \\ 4 \end{pmatrix}$ *does* **not** *dominate* $\frac{W}{2}$.

- *take* v_{11}*:* $w_{11} = \begin{pmatrix} 7 \\ 5 \\ 21 \end{pmatrix}$ *dominates* $\frac{W}{2}$. *Therefore* $w_9 := \begin{pmatrix} 9 \\ 7 \\ 27 \end{pmatrix}$.

- *take* v_{10}*:* $w_{10} = \begin{pmatrix} 2 \\ 2 \\ 4 \end{pmatrix}$ *dominates* $\frac{W}{2}$. *Therefore* $w_9 := \begin{pmatrix} 11 \\ 9 \\ 31 \end{pmatrix}$.

- *take* v_9*:* $w_9 = \begin{pmatrix} 11 \\ 9 \\ 31 \end{pmatrix}$ *does* **not** *dominate* $\frac{W}{2}$.

Since we delete after every domination step the corresponding node from the tree according to Algorithm 5.1 and no leaf with status **not investigated** *is left we end up with*

$$\mathcal{X}_{par} = L[v_9, v_3] \; .$$

6 Conclusions

In this paper we have developed polynomial algorithms for solving lexicographic and Pareto location problems on networks, where all the single-criterion objective functions $f^q(x)$ are of the sum (or median) type. The lexicographical problems are reduced to the problem of finding all lexicographically minimum vectors in a finite set, where all possible permutations of the vector components need to be considered.

Pareto locations can be found with a two-stage strategy. In Stage 1 all Pareto locations with respect to an edge are found, whereas Stage 2 consists in a check which of the candidates found in Stage 1 are dominated. While Stage 1 is similar for the case of $Q = 2$ and $Q > 2$ criteria, Stage 2 is done differently. The resulting polynomial algorithms are considerably improved in the case where the graph is a tree. In particular, Goldman's dominance algorithm is applicable for finding Pareto locations.

Several of the results can immediately be carried over to multi-criteria problems, where each single criterion is of the max (or center) type. The analysis of this paper is based heavily on the partition of the functions f^q into linear subedges — a fact which is also true for center problems, although the convexity property of the overall function f^q on edges e is lost. A paper on this subject is forthcoming.

Another interesting research topic is the mixing of function types. If, for instance, sum and max functions are among $f^1, \ldots, f^Q$, the key concept of linearity on subedges continues to hold and should make it possible to solve the corresponding lex and Pareto location problems. If Euclidean distances with repulsion (i.e. $w_i^1 < 0$) and network distance with attraction (i.e. $w_i^2 > 0$) are considered, different arguments have to be used. Here Pareto locations can be characterized by the solution of restricted network location problems where "circles" centered at the nodes define forbidden parts of the network.

Finally we may investigate orders of the objective space which are different from the lexicographical and componentwise one considered in this paper. For instance, the max ordering, given by

$$(z_1, \ldots, z_Q) \leq_{MO} (y_1, \ldots, y_Q) \;:\Leftrightarrow\; \max\{z_1, \ldots, z_Q\} \leq \max\{y_1, \ldots, y_Q\} \; ,$$

is of interest which was already successfully used in planar location problems (see [Hamacher and Nickel, 1995], [Hamacher, 1995] and [Nickel, 1995]). Other orders can be found in [Ehrgott, 1996].

References

[Ahuja et al., 1993] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows. Theory, Algorithms and Applications.* Prentice-Hall.

[Dearing et al., 1976] Dearing, P. M., Francis, R. L., and Lowe, T. J. (1976). Convex location problems on tree networks. *Operations Research*, 24:628–642.

[Ehrgott, 1996] Ehrgott, M. (1996). A characterization of lexicographic max-ordering solutions. Technical report, University of Kaiserslautern, Department of Mathematics. Report in Wirtschaftsmathematik No. 12. Accepted for Proceedings of the 6th Workshop of the DGOR-Working Group Multicriteria Optimization and Decision Theory.

[Goldman, 1971] Goldman, A. J. (1971). Optimal center location in simple networks. *Transportation Science*, 5:212–221.

[Hamacher, 1995] Hamacher, H. W. (1995). *Mathematische Lösungsverfahren für planare Standortprobleme.* Vieweg, Braunschweig.

[Hamacher and Nickel, 1995] Hamacher, H. W. and Nickel, S. (1995). Restricted planar location problems and applications. *Naval Research Logistics*, 42:967–992.

[Hamacher and Nickel, 1996] Hamacher, H. W. and Nickel, S. (1996). Multicriteria planar location problems. *European Journal of Operational Research*, 94:66–86.

[Hansen et al., 1991] Hansen, P., Labbé, M., and Thisse, J.-F. (1991). From the median to the generalized center. *RAIRO*, 25:73–86.

[Hansen et al., 1986] Hansen, P., Thisse, J.-F., and Wendell, R. E. (1986). Efficient points on a network. *Networks*, 16:358–367.

[Hershberger, 1989] Hershberger, J. (1989). Finding the upper envelope of n line segments in $o(n \log n)$ time. *Information Processing Letters*, 33(4):169–174.

[Labbé et al., 1995] Labbé, M., Peeters, D., and Thisse, J.-F. (1995). Location on networks. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, *Network Routing*, volume 8 of *Handbooks in OR & MS*. Elsevier.

[Levy, 1967] Levy, J. (1967). An extended theorem for location on a network. *Operational Research Quarterly*, 18:433–442.

[Lowe, 1978] Lowe, T. J. (1978). Efficient solutions in multi-objective tree network location problems. *Transportation Science*, 12:298–316.

[Megiddo, 1982] Megiddo, N. (1982). Linear-time alogrithms for linear programming in r^3 and related problems. *SIAM Journal on Computing*, 12:759–776.

[Nickel, 1995] Nickel, S. (1995). *Discretization of Planar Location Problems*. Shaker.