

Compromise Solutions in Multicriteria Combinatorial Optimization

Christina Büsing Kai-Simon Goetzmann Jannik Matuschke

November 15, 2011

Abstract

In multicriteria optimization, a compromise solution is a feasible solution whose cost vector minimizes the distance to the *ideal point* w.r.t. a given norm. The coordinates of the ideal point are given by the optimal values for the single optimization problem for each criterion.

We show that the concept of compromise solutions fits nicely into the existing notion of Pareto optimality: For a huge class of norms, every compromise solution is Pareto optimal, and under certain conditions on the norm *all* Pareto optimal solutions are also a compromise solution, for an appropriate weighting of the criteria. Furthermore, under similar conditions on the norm, the existence of an FPTAS for compromise solutions guarantees the approximability of the Pareto set.

These general results are completed by applications to classical combinatorial optimization problems. In particular, we study approximation algorithms for the multicriteria shortest path problem and the multicriteria minimum spanning tree problem. On the one hand, we derive approximation schemes for both problems, on the other hand we show that for the latter problem simple approaches like local search and greedy techniques do not guarantee good approximation factors.

Keywords: Multicriteria Combinatorial Optimization, Compromise Solutions, Pareto Optimality, Approximation

1 Introduction

1.1 Motivation

In multicriteria optimization, the probably most popular solution concept is that of Pareto optimality. Loosely speaking, a feasible solution is Pareto optimal if there is no other solution that is strictly better in one objective without being worse in another. This concept, however, has the drawback that in general a lot of solutions have this property. It might even be that *all* solutions are Pareto optimal, resulting in an exponential size of the Pareto set.

One way to choose a single, in some sense optimal solution is scalarization, where the (possibly weighted) sum of all objectives is maximized (or minimized). Since the criteria are merged into one objective function, scalarization yields a single optimal solution. This solution is also guaranteed to be Pareto optimal. The concept solves the problem of choosing a single solution from a huge set of candidates, but it has a disadvantage as well: The optimum of the scalarization can be very unbalanced, even if very balanced solutions with an only slightly worse scalarization value exist. This was also observed by Chen et al. [3] in a computational study.

The two aspects discussed above motivate the concept of *compromise solutions*, introduced by Yu [8] in 1973. It yields a single solution which is, in most cases, Pareto optimal, and also guarantees a certain balancing of the objectives. The idea is to find a solution that minimizes the distance to the *ideal point*, which is the point in the objective space obtained by optimizing each objective individually. The distance between a solution and the ideal point can be measured by any norm in \mathbb{R}^k (where k is the number of criteria).

1.2 Definitions and Notations

A multicriteria optimization problem is given by a set of feasible solutions \mathcal{X} and a vector of k objective functions (f_1, \dots, f_k) . For simplicity we restrict to maximization problems. All the definitions can be transferred to minimization problems, however, and also the results carry over. The intuition behind this is that the multicriteria optimization problem, no matter whether it is a minimization or maximization problem, is turned into a single-objective minimization problem – we minimize the distance to the ideal point.

For $y, y' \in \mathbb{R}^k$, we write $y \leq y'$ if $y_i \leq y'_i$ for all $i = 1, \dots, k$, and $y < y'$ if $y_i < y'_i$ for all $i = 1, \dots, k$. By $\mathbb{1}$ we denote the vector of ones of corresponding size, and $[k]$ denotes the set $\{1, 2, \dots, k\}$.

In the context of compromise solutions, we identify a solution with its cost vector, and therefore define the set of all feasible cost vectors as

$$\mathcal{Y} := \{(f_1(x), \dots, f_k(x)) : x \in \mathcal{X}\}.$$

The set of cost vectors that correspond to Pareto optimal solutions (i.e. the set of *non-dominated* cost vectors) is

$$\mathcal{Y}_P := \{y \in \mathcal{Y} : \nexists y' \in \mathcal{Y} \text{ such that } y' \geq y \text{ and } y'_j > y_j \text{ for some } j \in [k]\}.$$

From now on, we will mainly restrict our attention to elements in \mathcal{Y} and implicitly assume that we can always reconstruct the corresponding solution in \mathcal{X} .

Assumptions: We consider integral problems, therefore we can assume w.l.o.g. that all cost vectors are integral, i.e. $\mathcal{Y} \subseteq \mathbb{Z}^k$. We also assume, similar to the assumptions by Papadimitriou and Yannakakis in [5], that the costs are bounded by $|y_i| \leq 2^{\pi(|I|)}$ for all

$y \in \mathcal{Y}$ and all $i \in [k]$, where π is some polynomial and $|I|$ is the encoding length of the input. This is no restriction to any of the classical combinatorial optimization problems, where the objective is the sum of the values of individual elements. We will later denote this bound by M . The assumption $M = 2^{\pi(|I|)}$ means that M has an encoding length which is polynomial in the input.

We now define the central notions of this paper.

Definition 1.1 (Ideal Point). *For a multicriteria maximization problem with set of cost vectors $\mathcal{Y} \subseteq \mathbb{Z}^k$, the ideal point $y^* \in \mathbb{Z}^k$ is defined by*

$$y_j^* = \max_{y \in \mathcal{Y}} y_j .$$

Definition 1.2 (Compromise Solution). *For a multicriteria maximization problem with set of cost vectors $\mathcal{Y} \subseteq \mathbb{Z}^k$ and ideal point $y^* \in \mathbb{Z}^k$, a compromise solution w.r.t. to a norm $\|\cdot\|$ on \mathbb{R}^k is a vector $y^{cs} \in \mathcal{Y}$ with*

$$\|y^* - y^{cs}\| = \min_{y \in \mathcal{Y}} \|y^* - y\| .$$

1.3 Considered Norms

In this paper, we discuss two different families of norms, both parametrized by some number $p \in [1, \infty]$. The first one is the standard ℓ^p -norm:

$$\|y\|_p := \left(\sum_{i=1}^k y_i^p \right)^{1/p} \quad \text{for } p \in [1, \infty),$$

$$\|y\|_\infty := \max_{i=1, \dots, k} y_i .$$

Since computing compromise solutions w.r.t. this norm in general involves arbitrary exponents, leading to computational and possibly numerical problems, we also consider a parametrized sum of the ℓ^1 - and the ℓ^∞ -norm:

$$\|y\|_p := \|y\|_\infty + \frac{1}{p} \|y\|_1 \quad \text{for } p \in [1, \infty),$$

$$\|y\|_\infty := \|y\|_\infty .$$

We call this norm the *cornered p -norm* due to the shape of its unit spheres. This family of norms has also been considered by Gearhardt [4], and a similar one by Steuer and Choo [7].

We also consider weighted versions of norms: For any norm $\|\cdot\|$ on \mathbb{R}^k and any $\lambda \in \mathbb{R}^k$ with $\lambda \geq 0$, $\lambda \neq 0$, set

$$\|(y_1, \dots, y_k)\|^\lambda := \|(\lambda_1 y_1, \dots, \lambda_k y_k)\| .$$

On the one hand, the weights can be used by a decision maker to reflect the importance of the criteria or to scale them in order to get similar magnitudes in all of them. On the other hand, we will use varying weights to get all Pareto optimal solution as compromise solutions.

1.4 Basic Properties

In a given multicriteria maximization problem, for $\lambda \in \mathbb{R}_{\geq 0}^k \setminus \{0\}$ and $p \in [1, \infty]$, let us denote by $\text{CS}_\ell(\lambda, p)$ and $\text{CS}_c(\lambda, p)$ the set of all compromise solutions w.r.t. $\|\cdot\|_p^\lambda$ and $\|\cdot\|_p^\lambda$, respectively. Except from degenerated instances, these sets will consist of a single element, because finding a compromise solution is in fact a single-objective optimization problem, and the distance to the ideal point induces a total ordering on the set of feasible solutions. Further, denote by $\text{CS}_\ell(p)$ and $\text{CS}_c(p)$ the set of all compromise solutions w.r.t. $\|\cdot\|^\lambda$ and $\|\cdot\|^\lambda$ for any feasible weight vector, i.e.

$$\text{CS}_\ell(p) = \bigcup_{\lambda \in \mathbb{R}_{\geq 0}^k \setminus \{0\}} \text{CS}_\ell(\lambda, p), \quad \text{CS}_c(p) = \bigcup_{\lambda \in \mathbb{R}_{\geq 0}^k \setminus \{0\}} \text{CS}_c(\lambda, p).$$

We use $\text{CS}(\lambda, p)$ and $\text{CS}(p)$ in statements that hold for both norms.

Gearhardt [4] considered compromise solutions for the norms defined above in 1979, and observed several interesting properties. We will only point out two of them, which are of particular interest to us.

- (i) For $p \in [1, \infty)$, all compromise solutions are Pareto optimal, i.e.

$$\text{CS}(p) \subseteq \mathcal{Y}_P.$$

- (ii) If \mathcal{Y}_P is bounded, the distance between $\text{CS}(p)$ and \mathcal{Y}_P tends to 0 as p approaches infinity:

$$\sup_{y \in \mathcal{Y}_P} \text{dist}_\infty(y, \text{CS}(p)) \xrightarrow{p \rightarrow \infty} 0, \quad \text{where } \text{dist}_\infty(y, A) = \inf_{y' \in A} \|y - y'\|_\infty.$$

Since in our case all cost vectors are integral, \mathcal{Y}_P has only finitely many elements if it is bounded. Consequently, it follows from (ii) that there is a finite number p such that $\text{CS}(p) = \mathcal{Y}_P$, i.e. if the norm parameter is chosen sufficiently large, any Pareto optimal solution is also a compromise solution, for an appropriate choice of the weight vector λ .

Structure of the paper. In Section 2 we will investigate which value of p suffices for equality of the Pareto set and the set of all compromise solutions. In Section 3 we show that, for sufficiently large but finite p , an FPTAS for compromise solutions suffices to find an ε -approximation of the Pareto set. We then turn to some results on approximating compromise solutions: In 4.1 we prove that scalarization always yields

a constant factor approximation; in 4.2 we extend a result by Aissi et al. [2] on min-max regret robust optimization to show the existence of approximation schemes for the compromise solutions of the multicriteria minimum spanning tree problem (M-MST) and the multicriteria shortest path problem (M-SP); finally in 4.3 we analyze some local search and greedy algorithms for M-MST.

2 How to Choose the Norm Parameter

In this section, we study the size of the norm parameter that is necessary such that all Pareto optimal solutions are compromise solutions. We show that under the assumed bound $M = 2^{\pi(|I|)}$ on the cost vectors the parameter can be chosen such that it has polynomial encoding length.

Theorem 2.1. *Let $\mathcal{Y} \subseteq \mathbb{Z}^k \cap [-M, M]^k$ be the cost vector set of a multicriteria maximization problem. Then $\mathcal{Y}_P = \text{CS}_\ell(p)$ for any $p \in (\log k / \log(1 + \frac{1}{2M}), \infty)$, and $\mathcal{Y}_P = \text{CS}_c(p)$ for any $p \in (2kM, \infty)$.*

Proof. We first prove the statement for the cornered norm. Let y^* be the ideal point, and let $y \in \mathcal{Y}_P$. We will choose a weight vector $\lambda(y) \in \mathbb{R}_{\geq 0}^k \setminus \{0\}$ such that $\text{CS}_c(\lambda(y), p) = \{y\}$ for $p > 2kM$.

Case 1: $y < y^*$. Then we choose $\lambda(y)$ such that the maximum $\max_i \{\lambda_i(y)(y_i^* - y_i)\}$ is obtained in all components at the same time, i.e.

$$\lambda_i(y)(y_i^* - y_i) = C \quad \forall i \in [k]$$

for some constant C . For reasons of simplicity, we choose $C = 1$ and get

$$\lambda_i(y) = \frac{1}{y_i^* - y_i} .$$

Intuitively, this means that y is the “bottom-left corner” of a ball around y^* w.r.t. $\|\cdot\|_p^{\lambda(y)}$.

Now let $y' \in \mathcal{Y}$ with $y' \neq y$. Since $y \in \mathcal{Y}_P$, there exists $j \in [k]$ with $y_j > y'_j$. Integrality implies that $y_j \geq y'_j + 1$. We get

$$y_j - y'_j \geq 1 , \tag{2.1}$$

$$y'_i - y_i \leq 2M \quad \forall i \neq j . \tag{2.2}$$

We want to show that $\|y^* - y\|_p^{\lambda(y)} < \|y^* - y'\|_p^{\lambda(y)}$ for $p > 2kM$, i.e.

$$\max_i \underbrace{\{\lambda_i(y)(y_i^* - y_i)\}}_{=1} + \frac{1}{p} \cdot \underbrace{\lambda(y)^T(y^* - y)}_{=k} < \underbrace{\max_i \{\lambda_i(y)(y_i^* - y'_i)\} + \frac{1}{p} \cdot \lambda(y)^T(y^* - y')}_{(*)} . \tag{2.3}$$

We have

$$\frac{y_j^* - y'_j}{y_j^* - y_j} = \frac{y_j^* - y_j + y_j - y'_j}{y_j^* - y_j} = 1 + \frac{y_j - y'_j}{y_j^* - y_j} \geq 1 + \frac{1}{y_j^* - y_j} \geq 1 + \frac{1}{2M} ,$$

thus we can bound (*) as follows:

$$(*) = \underbrace{\max_i \left\{ \frac{y_i^* - y_i'}{y_i^* - y_i} \right\}}_{\geq 1 + \frac{1}{2M}} + \frac{1}{p} \underbrace{\sum_i \frac{y_i^* - y_i'}{y_i^* - y_i}}_{\geq 0} \geq 1 + \frac{1}{2M} .$$

It follows that (2.3) is fulfilled if

$$1 + \frac{k}{p} < 1 + \frac{1}{2M} \quad \Leftrightarrow \quad p > 2kM .$$

Case 2: $\exists j : y_j = y_j^*$. Let $J := \{j \in [k] : y_j = y_j^*\}$. The idea is to choose big values λ_j for all $j \in J$. Then only solutions y' with $y_j' = y_j^*$ for all $j \in J$ are competitive to y . In particular, we set

$$\lambda_j(y) = \begin{cases} k + 1 & \text{if } j \in J , \\ \frac{1}{y_j^* - y_j} & \text{otherwise .} \end{cases}$$

We have

$$\begin{aligned} \|y^* - y\|_p^{\lambda(y)} &= \max_{i \notin J} \underbrace{\{\lambda_i(y)(y_i^* - y_i)\}}_{=1} + \frac{1}{p} \cdot \sum_{i \notin J} \underbrace{\lambda_i(y)(y_i^* - y_i)}_{=1} \\ &= 1 + \frac{1}{p}(k - |J|) < 1 + \frac{k}{p} \leq 1 + k . \end{aligned}$$

Let $y' \in \mathcal{Y} \setminus \{y\}$.

Case 2.1: $\exists j \in J : y_j' \neq y_j^*$. Then $y_j' \leq y_j^* - 1$ and hence

$$\begin{aligned} \|y^* - y'\|_p^{\lambda(y)} &\geq \lambda_j(y)(y_j^* - y_j') + \frac{1}{p} \cdot \lambda_j(y)(y_j^* - y_j) \\ &\geq \lambda_j(y) \cdot \left(1 + \frac{1}{p}\right) \\ &> k + 1 \\ &> \|y^* - y\|_p^{\lambda(y)} . \end{aligned}$$

Case 2.2: $\forall j \in J : y_j' = y_j^*$. Similar calculations as in Case 1 show that

$$\|y^* - y\|_p^{\lambda(y)} < \|y^* - y'\|_p^{\lambda(y)} \quad \text{if } p > 2kM .$$

These considerations show that it suffices to choose $p > 2kM$ in order to obtain $\mathcal{Y}_P = \text{CS}_c(p)$.

We now turn to the ℓ^p -norm. For a given $y \in \mathcal{Y}_P, y < y^*$, set the weight vector $\lambda(y)$ as before:

$$\lambda_i(y) := \frac{1}{y_i^* - y_i} .$$

Then

$$\left(\|y^* - y\|_p^{\lambda(y)}\right)^p = \sum_{i=1}^k \left(\frac{y_i^* - y_i}{y_i^* - y_i}\right)^p = k .$$

For any $y' \in \mathcal{Y}_P, y' \neq y$, as above we have some j with $y'_j \leq y_j - 1$. It follows that

$$\left(\|y^* - y'\|_p^{\lambda(y)}\right)^p = \underbrace{\left(\frac{y_j^* - y'_j}{y_j^* - y_j}\right)^p}_{\geq 1 + \frac{1}{2M}} + \sum_{\substack{i \in [k] \\ i \neq j}} \underbrace{\left(\frac{y_i^* - y'_i}{y_i^* - y_i}\right)^p}_{\geq 0} \geq \left(1 + \frac{1}{2M}\right)^p,$$

so

$$\|y^* - y\|_p^{\lambda(y)} < \|y^* - y'\|_p^{\lambda(y)} \quad \text{if} \quad k < \left(1 + \frac{1}{2M}\right)^p \Leftrightarrow p > \frac{\log k}{\log\left(1 + \frac{1}{2M}\right)}.$$

If $J := \{j : y_j = y_j^*\} \neq \emptyset$, then we set $\lambda_j(y) = k$ for all $j \in J$.

If $y'_j = y_j^*$ for all $j \in J$, the calculations from above hold and $\|y^* - y\|_p^{\lambda(y)} < \|y^* - y'\|_p^{\lambda(y)}$ for $p > \frac{\log k}{\log\left(1 + \frac{1}{2M}\right)}$. Otherwise there is some $j \in J : y'_j \leq y_j^* - 1$ and thus

$$\left(\|y^* - y'\|_p^{\lambda(y)}\right)^p \geq (\lambda_j(y) \cdot (y_j^* - y'_j))^p \geq \lambda_j(y)^p \geq k,$$

while

$$\left(\|y^* - y'\|_p^{\lambda(y)}\right)^p = k - |J| < k$$

and hence

$$\|y^* - y\|_p^{\lambda(y)} < \|y^* - y'\|_p^{\lambda(y)}.$$

This completes the proof. \square

Remark. The case $y_j = y_j^*$ for some j never occurs, if instead of y^* we use $y^* + \varepsilon \cdot \mathbb{1}$ for some $\varepsilon > 0$. This is a fairly common approach.

To maintain integrality, we have to choose $\varepsilon \geq 1$. A smaller value, such as $\varepsilon = 1/K$, can be modeled by scaling all cost vectors by K and setting $\varepsilon = 1$.

Conclusion. Since we assume $|y_i| \leq 2^{\pi(|I|)}$, Theorem 2.1 shows that for both norms, p can be chosen such that it has an encoding length polynomial in $|I|$, and still all non-dominated solutions are compromise solutions for an appropriate weight vector. Also the weights we choose have a polynomial encoding.

3 Approximation of the Pareto Set

In this section, we will establish a connection between the approximability of compromise solutions and the approximability of the Pareto set. Since we deal with approximation factors here, we restrict to non-negative cost vectors, i.e. we assume $\mathcal{Y} \subseteq \mathbb{N}^k$ (where \mathbb{N} includes the zero).

The approximation of the Pareto set is defined as follows:

Definition 3.1 (ε -approximate Pareto Set). *For a multicriteria maximization problem with cost vector set $\mathcal{Y} \subseteq \mathbb{N}^k$ and Pareto set \mathcal{Y}_P , an ε -approximate Pareto set is a set $\mathcal{Y}_{\varepsilon P} \subseteq \mathcal{Y}$ such that for all $y \in \mathcal{Y}_P$ there is some $y' \in \mathcal{Y}_{\varepsilon P}$ with $y' \geq \frac{1}{1+\varepsilon} \cdot y$.*

A fully polynomial time approximation scheme (FPTAS) for the Pareto set is an algorithm that, for any $\varepsilon > 0$, computes an ε -approximate Pareto set in time polynomial in the encoding size of the input and $1/\varepsilon$.

In 2000, Papadimitriou and Yannakakis [5] showed that there is an FPTAS for the Pareto set if and only if the GAP problem is solvable in polynomial time:

Definition 3.2 (GAP Problem). *For a multicriteria maximization problem with cost vector set $\mathcal{Y} \subseteq \mathbb{N}^k$, the GAP problem is the following: Given a vector $y \in \mathbb{R}^k$ and a number $\varepsilon > 0$, either return $y' \in \mathcal{Y}$ with $y' \geq y$, or decide that there is no $y'' \in \mathcal{Y}$ such that $y'' \geq (1 + \varepsilon)y$.*

We show that, for sufficiently large values of p , an FPTAS for compromise solutions enables us to solve the GAP problem. Thus, if we can approximate compromise solutions, we can also approximate the Pareto set. The notion of approximation algorithms for compromise solutions is used in the classical sense, i.e. the approximation factor refers to the objective value, which is in our case the distance to the ideal point. It does *not* refer to the actual costs of a solution in the different criteria, as it is the case for the approximation of the Pareto set.

Theorem 3.3. *Let $\mathcal{Y} \subseteq \mathbb{N}^k \cap [0, M]^k$ be the cost vector set of a multicriteria maximization problem. If for all $\lambda \in \mathbb{R}_{\geq 0}^k \setminus \{0\}$ there is an FPTAS for $\text{CS}_\ell(\lambda, p)$ for some $p > M \log k / \varepsilon \log(1 + \frac{1}{M})$ or an FPTAS for $\text{CS}_c(\lambda, p)$ for some $p > \frac{kM^2}{\varepsilon}$, then the GAP problem is solvable in polynomial time.*

Proof. Again, we consider the case of the cornered norm first. Let y^* be the ideal point. By $\text{val}_p^\lambda(y)$ we denote the *value* of y in the problem $\text{CS}_c(\lambda, p)$, i.e.

$$\text{val}_p^\lambda(y) := \|y^* - y\|_p^\lambda.$$

For $y \in \mathbb{R}^k, 0 \leq y \leq y^*$, define the corresponding weight $\lambda(y)$ by

$$\lambda_i(y) = \begin{cases} k + 1 & \text{if } y_i = y_i^*, \\ 0 & \text{if } y_i = 0, \\ 1/y_i^* - y_i & \text{otherwise.} \end{cases}$$

With these slightly modified weights all considerations in Section 2 still hold. By choosing $\lambda_i(y) = 0$ if $y_i = 0$ we simply restrict to those dimensions where y has a positive (and thus relevant) component.

Let $y \in \mathbb{R}_{\geq 0}^k$ and $\varepsilon > 0$ be given as input to GAP, and suppose we have an FPTAS for $\text{CS}_c(\lambda(y), p)$ for some $p > \frac{kM^2}{\varepsilon}$. W.l.o.g. $y < y^*$, otherwise the answer to GAP is obviously negative:

$$\exists j : y_j \geq y_j^* \quad \Rightarrow \quad \nexists y' \in \mathcal{Y} : y'_j \geq (1 + \varepsilon)y_j > y_j^* \quad \Rightarrow \quad \nexists y' \in \mathcal{Y} : y' \geq (1 + \varepsilon)y .$$

We also assume $y \neq 0$, since otherwise GAP reduces to finding a feasible solution:

$$\exists y' \in \mathcal{Y} : y' \geq y = 0 \quad \Leftrightarrow \quad \mathcal{Y} \neq \emptyset .$$

Finally, w.l.o.g. we assume that $\varepsilon < 1/y_i$ for all i .

We want to compute a $(1 + \varepsilon')$ -approximate solution to $\text{CS}(\lambda(y), p)$ for some ε' that depends on ε and M (the bound on the objective values) such that we can answer GAP.

We first consider the case where y is integral. This is a restriction, since for the result from [5] we need arbitrary $y \in \mathbb{Q}^k$. However, we can modify the considerations in such a way that they work for fractional y as well. We present these modifications after establishing the case $y \in \mathbb{Z}^k$.

Positive Case. Suppose we find $y' \in \mathcal{Y}$ with $\text{val}_p^{\lambda(y)}(y') \leq \text{val}_p^{\lambda(y)}(y)$. Then by the considerations from Section 2 it follows that $y' \geq y$, so y' is a positive answer to GAP.

Negative Case. Let y' be a $(1 + \varepsilon')$ -approximation to $\text{CS}(\lambda(y), p)$ with $\text{val}_p^{\lambda(y)}(y') > \text{val}_p^{\lambda(y)}(y)$. Set $\alpha := 1 + \varepsilon'$. Then we know

$$\nexists y'' \in \mathcal{Y} : \quad \|y^* - y''\|_p^{\lambda(y)} \leq \frac{1}{\alpha} \|y^* - y\|_p^{\lambda(y)} =: r . \quad (3.1)$$

Set $\tilde{y} := y^* - \frac{1}{\alpha}(y^* - y)$. The distance between \tilde{y} and y^* w.r.t. $\|\cdot\|_p^{\lambda(y)}$ is r :

$$\|y^* - \tilde{y}\|_p^{\lambda(y)} = \frac{1}{\alpha} \|y^* - y\|_p^{\lambda(y)} = r .$$

Also, the maximum in the norm is obtained in all $i \in I := \{i \in [k] : y_i > 0\}$ at the same time:

$$\lambda_i(y)(y_i^* - \tilde{y}_i) = \frac{1}{\alpha} \cdot \frac{y_i^* - y_i}{y_i^* - y_i} = \frac{1}{\alpha} \quad \forall i \in I .$$

Intuitively, \tilde{y} is the “bottom-left corner” of the ball around y^* with radius r . By (3.1) that ball contains no solution (see Figure 3.1).

Since the ball w.r.t. $\|\cdot\|_\infty^{\lambda(y)}$ with corner \tilde{y} is contained in the ball w.r.t. $\|\cdot\|_p^{\lambda(y)}$ with the same corner, it follows that

$$\begin{aligned} \nexists y'' \in \mathcal{Y} : \quad & \|y^* - y''\|_\infty^{\lambda(y)} \leq \|y^* - \tilde{y}\|_\infty^{\lambda(y)} \\ & = \max_i \left\{ \lambda_i(y)(y_i^* - (y_i^* - \frac{1}{\alpha}(y_i^* - y_i))) \right\} \\ & = \frac{1}{\alpha} \max_i \{ \lambda_i(y)(y_i^* - y_i) \} \\ & = \frac{1}{\alpha} \|y^* - y\|_\infty^{\lambda(y)} . \end{aligned} \quad (3.2)$$

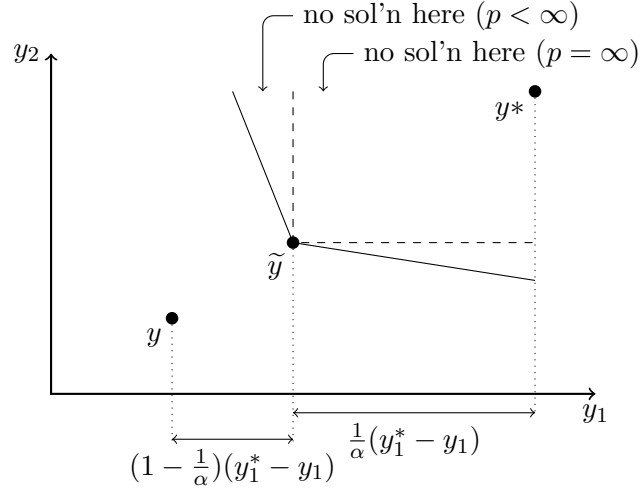


Figure 3.1: Approximation of the compromise solution

We want to have a guarantee that there is no $y'' \in \mathcal{Y}$ with $y'' \geq (1 + \varepsilon)y$, i.e.

$$\begin{aligned} & \nexists y'' \in \mathcal{Y} : \quad y_i'' \geq (1 + \varepsilon)y_i \quad \forall i = 1, \dots, k \\ \Leftrightarrow & \quad \forall y'' \in \mathcal{Y} \exists i \in \{1, \dots, k\} : \quad y_i'' < (1 + \varepsilon)y_i . \end{aligned} \quad (3.3)$$

Now (3.2) is equivalent to

$$\begin{aligned} & \forall y'' \in \mathcal{Y} : \quad \max_i \{\lambda_i(y)(y_i^* - y_i'')\} > \frac{1}{\alpha} \max_i \{\lambda_i(y)(y_i^* - y_i)\} = \frac{1}{\alpha} \\ \Leftrightarrow & \quad \forall y'' \in \mathcal{Y} \exists i \in I : \quad \lambda_i(y)(y_i^* - y_i'') > \frac{1}{\alpha} \\ \Leftrightarrow & \quad \forall y'' \in \mathcal{Y} \exists i \in I : \quad y_i^* - y_i'' > \frac{1}{\alpha}(y_i^* - y_i) \\ \Leftrightarrow & \quad \forall y'' \in \mathcal{Y} \exists i \in I : \quad y_i'' < y_i^* - \frac{1}{\alpha}(y_i^* - y_i) = \tilde{y}_i . \end{aligned} \quad (3.4)$$

Consequently, if we can choose α such that $\tilde{y}_i \leq (1 + \varepsilon)y_i$, this would give us (3.3). We need this inequality for all $i \in I$, since the index whose existence is guaranteed by (3.4) might vary with y'' .

For $i \in I$ we have

$$\tilde{y}_i = y_i^* - \frac{1}{\alpha}(y_i^* - y_i) \leq (1 + \varepsilon)y_i \quad (3.5)$$

$$\begin{aligned} \Leftrightarrow & \quad y_i^* - y_i - \varepsilon y_i \leq \frac{1}{\alpha}(y_i^* - y_i) \\ \Leftrightarrow & \quad \alpha \leq \frac{y_i^* - y_i}{y_i^* - y_i - \varepsilon y_i} \quad \text{since } y_i^* \geq y_i + 1 \text{ and } \varepsilon < \frac{1}{y_i} . \end{aligned} \quad (3.6)$$

Since $y_i \geq 1$ for $i \in I$, we have

$$\frac{y_i^* - y_i - \varepsilon y_i}{y_i^* - y_i} = 1 - \frac{\varepsilon y_i}{y_i^* - y_i} \leq 1 - \frac{\varepsilon}{M} , \quad (3.7)$$

With this, we can bound C from below. Recall that $y'_j = \lfloor y_j \rfloor$ for some j . Then for this j ,

$$C = \frac{y_j^* - \bar{y}_j}{y_j^* - \lfloor y_j \rfloor} \geq \frac{y_j^* - y'_j - \frac{1}{\beta}}{y_j^* - \lfloor y_j \rfloor} = \frac{y_j^* - \lfloor y_j \rfloor - \frac{1}{\beta}}{y_j^* - \lfloor y_j \rfloor} = 1 - \frac{1}{\beta(y_j^* - \lfloor y_j \rfloor)} \geq 1 - \frac{1}{\beta}.$$

As above, we know

$$\begin{aligned} \forall y'' \in \mathcal{Y} \exists i \in I : \quad y_i'' &< \tilde{y}_i := y_i^* - \frac{1}{\alpha}(y_i^* - \bar{y}_i) \\ &= y_i^* - \frac{1}{\alpha}(y_i^* - (y_i^* - C(y_i^* - \lfloor y_i \rfloor))) \\ &= y_i^* - \frac{C}{\alpha}(y_i^* - \lfloor y_i \rfloor) \\ &\leq y_i^* - \frac{C}{\alpha}(y_i^* - y_i) \\ &\leq y_i^* - \frac{1 - 1/\beta}{\alpha}(y_i^* - y_i). \end{aligned}$$

Thus we now need to choose α such that $y_i^* - \frac{1-1/\beta}{\alpha}(y_i^* - y_i) \leq (1 + \varepsilon)y_i$ for all $i \in I$. Inserting the term $(1 - 1/\beta)$ into the calculations (3.5–3.7), we see that it suffices to choose

$$\alpha = \frac{1 - 1/\beta}{1 - \varepsilon/M}.$$

As an approximation factor, α has to be greater than 1, i.e.

$$1 - \frac{1}{\beta} > 1 - \frac{\varepsilon}{M} \quad \Rightarrow \quad \beta > \frac{M}{\varepsilon}.$$

Thus we have to choose $p > \frac{kM^2}{\varepsilon}$, as claimed in the theorem.

All the arguments above also work if we consider the ℓ^p -norm instead of the cornered norm. The analogon of equation (3.2) is established as follows: First observe that if λ has k_λ non-zero components, then

$$\|x\|_\infty^\lambda \leq \|x\|_p^\lambda \leq k_\lambda^{\frac{1}{p}} \cdot \|x\|_\infty^\lambda. \quad (3.8)$$

Secondly, since y is at the corner of a ball around y^* w.r.t. $\|\cdot\|_\infty^{\lambda(y)}$, it holds that

$$\|y^* - y\|_p^\lambda = \left(\sum_i (\lambda_i (y_i^* - y_i))^p \right)^{\frac{1}{p}} = \left(k_\lambda \cdot \max_i \{ (\lambda_i (y_i^* - y_i))^p \} \right)^{\frac{1}{p}} = k_\lambda^{\frac{1}{p}} \cdot \|y^* - y\|_\infty^\lambda. \quad (3.9)$$

We know that there is no $y'' \in \mathcal{Y}$ with $\|y^* - y''\|_p^{\lambda(y)} \leq \frac{1}{\alpha} \|y^* - y\|_p^{\lambda(y)}$, and hence

$$\forall y'' \in \mathcal{Y} : \quad \|y^* - y''\|_p^{\lambda(y)} > \frac{1}{\alpha} \|y^* - y\|_p^{\lambda(y)} \quad (3.10)$$

$$\begin{aligned} \Rightarrow \forall y'' \in \mathcal{Y} : \quad \|y^* - y''\|_\infty^{\lambda(y)} &\stackrel{(3.8)}{\geq} \frac{1}{k^{\frac{1}{p}}} \cdot \|y^* - y''\|_p^{\lambda(y)} \\ &\stackrel{(3.10)}{>} \frac{1}{k^{\frac{1}{p}}} \cdot \frac{1}{\alpha} \cdot \|y^* - y\|_p^{\lambda(y)} \\ &\stackrel{(3.9)}{=} \frac{1}{k^{\frac{1}{p}}} \cdot \frac{1}{\alpha} \cdot k^{\frac{1}{p}} \|y^* - y\|_p^{\lambda(y)} \\ &= \frac{1}{\alpha} \|y^* - y\|_p^{\lambda(y)}. \end{aligned} \quad (3.11)$$

Equation (3.11) is the analogon to (3.2), the following calculations are similar to the case above.

To be able to solve the GAP problem also for fractional input y , we have to scale p by a factor of $\beta > \frac{M}{\varepsilon}$ again, so we need $p > \frac{M \log k}{\varepsilon \log(1 + \frac{1}{M})} \in \mathcal{O}\left(\frac{M^2 \log k}{\varepsilon}\right)$. The calculations are the same as before. \square

Remark. As in Section 2, in both cases the norm parameter can be chosen such that its encoding length is polynomial in the size of the input.

4 Approximation of CS – Positive and Negative Results

In this section we present some results on the approximation of compromise solutions.

4.1 Trivial Constant Factor Approximation Through Scalarization

As pointed out in the introduction, one way to get a single Pareto optimal solution is scalarization, i.e. maximizing the weighted sum of all criteria. The scalarization that corresponds to $\text{CS}(\lambda, p)$ is $\max_{x \in \mathcal{X}} \lambda^\top x$. It turns out that this yields a constant factor approximation to $\text{CS}(\lambda, p)$.

Theorem 4.1. *Let $\mathcal{Y} \subseteq \mathbb{N}^k$ be the cost vector set of a multicriteria maximization problem. For finite p and any k , the scalarization $\max_{y \in \mathcal{Y}} \lambda^\top y$ yields*

- a $\frac{k+kp}{k+p}$ -approximation to $\text{CS}_c(\lambda, p)$,
- a $k^{1-\frac{1}{p}}$ -approximation to $\text{CS}_\ell(\lambda, p)$,
- a k -approximation to $\text{CS}_c(\lambda, \infty) = \text{CS}_\ell(\lambda, \infty)$,

in polynomial time if the single-objective problem is efficiently solvable.

Remark. The approximation factor can be bounded by $\min\{k, p + 1\}$ for the cornered norm, and by k for the ℓ^p -norm. Since the number of criteria k is usually assumed to be constant, this result gives a constant factor approximation.

Proof. Let y^s be the optimum of the scalarization and let y^{cs} be the compromise solution w.r.t. λ and p . We further define

$$\tilde{y} := \operatorname{argmin} \left\{ \|y^* - y\|_p^\lambda : y \in \mathbb{R}^k, \lambda^\top y \leq \lambda^\top y^s \right\}.$$

Note that \tilde{y} is not necessarily in \mathcal{Y} . Since $\mathcal{Y} \subseteq \{y \in \mathbb{R}^k : \lambda^\top y \leq \lambda^\top y^s\}$, we know that $\operatorname{val}(y^{cs}) \geq \operatorname{val}(\tilde{y})$. We also know that \tilde{y} is at the corner of a ball around y^* w.r.t. $\|\cdot\|_p^\lambda$ (cf. Fig. 4.1) and thus (with $\lambda^\top \tilde{y} = \lambda^\top y^s$)

$$\lambda_i(y_i^* - \tilde{y}_i) = C = \frac{1}{k} \cdot \lambda^\top (y^* - y^s) \quad \forall i \in [k].$$

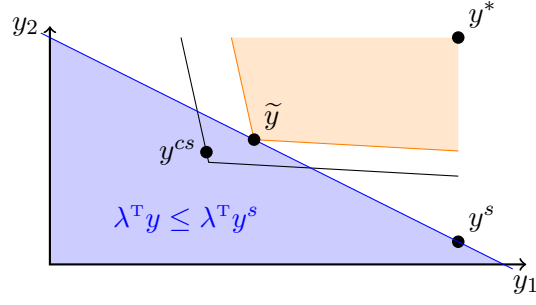


Figure 4.1: Approximation ratio of scalarization

We get

$$\begin{aligned}
\|y^* - y^{cs}\|_p^\lambda &\geq \|y^* - \tilde{y}\|_p^\lambda \\
&= \max_i \{\lambda_i (y_i^* - \tilde{y}_i)\} + \frac{1}{p} \cdot \sum_i \lambda_i (y_i^* - \tilde{y}_i) \\
&= \left(1 + \frac{k}{p}\right) C \\
&= \frac{1}{k} \cdot \left(1 + \frac{k}{p}\right) \lambda^\top (y^* - y^s).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\|y^* - y^s\|_p^\lambda &= \max_i \{\lambda_i (y_i^* - y_i^s)\} + \frac{1}{p} \cdot \sum_i \lambda_i (y_i^* - y_i^s) \\
&\leq \left(1 + \frac{1}{p}\right) \lambda^\top (y^* - y^s).
\end{aligned}$$

Thus,

$$\frac{\|y^* - y^s\|_p^\lambda}{\|y^* - y^{cs}\|_p^\lambda} \leq \frac{k \cdot \left(1 + \frac{1}{p}\right)}{1 + \frac{k}{p}} = \frac{k + kp}{k + p}.$$

This proves the claimed result for $\text{CS}_c(\lambda, p)$ with $p < \infty$.

We get the result for $p = \infty$ if we simply leave out the sum term of the norm in all the calculations above. Note that $\|\cdot\|_\infty = \|\cdot\|_\infty$, so we are left to show the result for the ℓ^p -norm for finite p . Using the notations from above we get

$$\begin{aligned}
\|y^* - y^{cs}\|_p^\lambda &\geq \|y^* - \tilde{y}\|_p^\lambda \\
&= \left(\sum_{i=1}^k (\lambda_i (y_i^* - \tilde{y}_i))^p\right)^{\frac{1}{p}} \\
&= k^{\frac{1}{p}} \cdot C \\
&= k^{\frac{1}{p}-1} \cdot \lambda^\top (y^* - y^s).
\end{aligned}$$

On the other hand,

$$\|y^* - y^s\|_p^\lambda \leq k^{\frac{1}{p}} \|y^* - y^s\|_1^\lambda = \lambda^T(y^* - y^s) .$$

Thus,

$$\frac{\|y^* - y^s\|_p^\lambda}{\|y^* - y^{cs}\|_p^\lambda} \leq k^{1 - \frac{1}{p}} .$$

This completes the proof. \square

4.2 From Pseudopolynomial Algorithms to Approximation Schemes

Multicriteria optimization and in particular the concept of compromise solutions is closely related to robust optimization. If each criterion is considered as one *scenario* in the robust setting, then a compromise solution w.r.t. $\|\cdot\|_\infty$ is exactly the same as a *min-max regret robust* solution.

Let OPT denote the minimal distance of a feasible solution to the ideal point. For min-max regret robust problems, Aissi et al. [1] show that if we can compute lower and upper bounds L and U on OPT in polynomial time, where $U \leq q(|I|)L$ for some polynomial q of the encoding length of the instance $|I|$, and if there is an algorithm for the min-max regret robust problem that runs in time polynomial in $|I|$ and the upper bound U , then there is an FPTAS for the min-max regret robust problem.

In this section we show that this result extends to $\text{CS}(\lambda, p)$ for $p \in [1, \infty)$ if the distance is measured in the cornered norm. For the ℓ^p -norm we get a $(4 + \varepsilon)$ -approximation. The pseudopolynomial algorithms from [1] for the shortest path problem and the minimum spanning tree problem work for $\text{CS}(\lambda, p)$ as well, resulting in FPTAS for these problems (for the cornered norm, resp. $(4 + \varepsilon)$ -approximation algorithms for the ℓ^p -norm).

4.2.1 The General Results

Notation. For a multicriteria maximization problem \mathcal{P} denote by $\text{CS}^{\mathcal{P}}(p)$ the problem of finding a compromise solution for \mathcal{P} . For simplicity, we consider the unweighted case, i.e. $\lambda = \mathbb{1}$, but all arguments carry over to the weighted case.

The results in this section only hold for binary combinatorial optimization problems, which include classical problems such as the shortest path and the minimum spanning tree problem, but not general integer problems, e.g. the minimum cost flow problem.

Instead of the set \mathcal{Y} of cost vectors, we now consider the space \mathcal{X} of feasible solutions. Its dimension is denoted by n , and because we restrict to binary problems we have $\mathcal{X} \subseteq \{0, 1\}^n$. The cost for element i in criterion j is c_{ji} , hence the cost of a solution $x \in \mathcal{X}$ w.r.t. criterion j is $c_j^T x$.

Theorem 4.2. *Let \mathcal{P} be a multicriteria maximization problem and $p \in [1, \infty)$. If*

(i) for any instance I of $\text{CS}_c^{\mathcal{P}}(p)$, a lower and an upper bound L and U on the optimal value of I can be computed in time $\pi_1(|I|)$, such that $U \leq \pi_2(|I|)L$, where π_1 and π_2 are non-decreasing polynomials,

(ii) and there exists an algorithm that finds for any instance I an optimal solution in time $\pi_3(|I|, U)$, where π_3 is a non-decreasing polynomial,

then there is an FPTAS for $\text{CS}_c^{\mathcal{P}}(p)$.

Proof. The result in [1] is proven by applying the given algorithm to an instance \bar{I} with modified cost coefficients. We modify their proof slightly: As in [1], let $t \leq n$ be an upper bound on the size (i.e. the number of non-zero entries) of any feasible solution. Set $\varepsilon' := \frac{\varepsilon}{1+k/p}$ and let $\bar{c}_{ji} := \left\lfloor \frac{2tc_{ji}}{\varepsilon'L} \right\rfloor$ be the cost coefficients of the modified instance \bar{I} . Let x^* and \bar{x}^* be optimal solutions to $\text{CS}_c^{\mathcal{P}}(p)$ for I and \bar{I} , respectively, denote the respective ideal points by y^* and \bar{y}^* and let $x^{(j)}, \bar{x}^{(j)}$ for $j \in [k]$ be optimal solutions of \mathcal{P} for c_j and \bar{c}_j .

We start with some auxiliary calculations: It holds that

$$\frac{\varepsilon'L}{2t} \cdot \bar{c}_{ji} \leq c_{ji} < \frac{\varepsilon'L}{2t} (\bar{c}_{ji} + 1), \quad (4.1)$$

$$y_j^* = c_j^{\text{T}} x^{(j)} \geq c_j^{\text{T}} \bar{x}^{(j)} \geq \frac{\varepsilon'L}{2t} \bar{c}_j^{\text{T}} \bar{x}^{(j)} = \frac{\varepsilon'L}{2t} \bar{y}_j^*, \quad (4.2)$$

$$y_j^* = c_j^{\text{T}} x^{(j)} \leq \frac{\varepsilon'L}{2t} (\bar{c}_j + \mathbb{1})^{\text{T}} x^{(j)} \leq \frac{\varepsilon'L}{2t} \bar{c}_j^{\text{T}} x^{(j)} + \frac{\varepsilon'L}{2} \leq \frac{\varepsilon'L}{2t} \bar{y}_j^* + \frac{\varepsilon'L}{2}. \quad (4.3)$$

With this we can bound the value of \bar{x}^* w.r.t. the original costs c . Let $\text{val}_p(x)$ and $\overline{\text{val}}_p(x)$ denote the values of a solution x w.r.t. the original and the modified costs, respectively. We get

$$\begin{aligned} \text{val}_p(\bar{x}^*) &= \max_j (y_j^* - c_j^{\text{T}} \bar{x}^*) + \frac{1}{p} \sum_j (y_j^* - c_j^{\text{T}} \bar{x}^*) \\ &\stackrel{(4.1)}{\leq} \max_j \left(\frac{\varepsilon'L}{2t} (\bar{y}_j^* - \bar{c}_j^{\text{T}} \bar{x}^*) \right) + \frac{\varepsilon'L}{2} + \frac{1}{p} \sum_j \left(\frac{\varepsilon'L}{2t} (\bar{y}_j^* - \bar{c}_j^{\text{T}} \bar{x}^*) + \frac{\varepsilon'L}{2} \right) \\ &= \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} + \frac{\varepsilon'L}{2t} \cdot \overline{\text{val}}_p(\bar{x}^*) \\ &\leq \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} + \frac{\varepsilon'L}{2t} \cdot \overline{\text{val}}_p(x^*) \\ &= \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} + \max_j \left(\frac{\varepsilon'L}{2t} (\bar{y}_j^* - \bar{c}_j^{\text{T}} x^*) \right) + \frac{1}{p} \sum_j \left(\frac{\varepsilon'L}{2t} (\bar{y}_j^* - \bar{c}_j^{\text{T}} x^*) \right) \\ &\stackrel{(4.1)}{\leq} \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} + \max_j \left(y_j^* - c_j^{\text{T}} x^* + \frac{\varepsilon'L}{2t} \mathbb{1}^{\text{T}} x^* \right) + \frac{1}{p} \sum_j \left(y_j^* - c_j^{\text{T}} x^* + \frac{\varepsilon'L}{2t} \mathbb{1}^{\text{T}} x^* \right) \\ &\stackrel{(4.2)}{\leq} \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} + \text{val}_p(x^*) + \left(1 + \frac{k}{p} \right) \frac{\varepsilon'L}{2} \\ &= \varepsilon L + \text{val}_p(x^*) \\ &\leq (1 + \varepsilon) \text{OPT}. \end{aligned}$$

It remains to be shown that \bar{x}^* can be computed in time polynomial in $|I|$ and $\frac{1}{\varepsilon}$. For this, denote by \bar{L} and \bar{U} the lower and upper bounds on the optimal value $\overline{\text{OPT}}$ of the modified instance \bar{I} . According to the requirements of the theorem we can compute L and then \bar{x}^* in time

$$\begin{aligned} \pi_1(|I|) + \pi_3(|\bar{I}|, \bar{U}) &\leq \pi_1(|I|) + \pi_3(|\bar{I}|, \pi_2(|\bar{I}|)\bar{L}) \\ &\leq \pi_1(|I|) + \pi_3(|\bar{I}|, \pi_2(|\bar{I}|)\overline{\text{OPT}}) \\ &\leq \pi_1(|I|) + \pi_3\left(|\bar{I}|, \pi_2(|\bar{I}|)\left(\frac{2t}{\varepsilon'}\pi_2(|I|) + t\left(1 + \frac{k}{p}\right)\right)\right), \end{aligned}$$

where the last inequation holds because

$$\begin{aligned} \overline{\text{OPT}} &= \max_j (\bar{y}_j^* - \bar{c}_j^\top \bar{x}^*) + \frac{1}{p} \sum_j (\bar{y}_j^* - \bar{c}_j^\top \bar{x}^*) \\ &\leq \max_j (\bar{y}_j^* - \bar{c}_j^\top x^*) + \frac{1}{p} \sum_j (\bar{y}_j^* - \bar{c}_j^\top x^*) \\ &\stackrel{(4.1)}{\leq} \max_j \left(\frac{2t}{\varepsilon' L} \cdot y_j^* - \frac{2t}{\varepsilon' L} \cdot c_j^\top x^* + t \right) + \frac{1}{p} \sum_j \left(\frac{2t}{\varepsilon' L} \cdot y_j^* - \frac{2t}{\varepsilon' L} \cdot c_j^\top x^* + t \right) \\ &\stackrel{(4.2)}{\leq} \frac{2t}{\varepsilon' L} \cdot \text{OPT} + t \left(1 + \frac{k}{p} \right) \\ &\leq \frac{2tU}{\varepsilon' L} + t \left(1 + \frac{k}{p} \right) \\ &\leq \frac{2t}{\varepsilon'} \cdot \pi_2(|I|) + t \left(1 + \frac{k}{p} \right). \end{aligned}$$

Finally note that $|\bar{I}| \leq \pi_4(|I|, \log \frac{1}{\varepsilon}, \log \frac{k}{p})$ for some polynomial π_4 , thus the above calculations prove that the running time is indeed polynomial. \square

Remark. For the running time it is essential that p is fixed or at least bounded from below by a positive constant (e.g. $p \geq 1$), as the running time is only polynomial in $\frac{1}{p}$. Since for $p \rightarrow 0$ the problem $\text{CS}_c^{\mathcal{P}}(p)$ at some point is equivalent to scalarization this is only a minor restriction.

For the case of the ℓ^p -norm the approximation factor can not be transferred. We get an additional factor of 4 in this case.

Theorem 4.3. *Let \mathcal{P} be a multicriteria maximization problem and $p \in [1, \infty)$. If*

- (i) *for any instance I of $\text{CS}_c^{\mathcal{P}}(p)$, a lower and an upper bound L and U on the optimal value of I can be computed in time $\pi_1(|I|)$, such that $U \leq \pi_2(|I|)L$, where π_1 and π_2 are non-decreasing polynomials,*
- (ii) *and there exists an algorithm that finds for any instance I an optimal solution in time $\pi_3(|I|, U)$, where π_3 is a non-decreasing polynomial,*

then there is a $(4 + \varepsilon)$ -approximation algorithm for $\text{CS}_\ell^{\mathcal{P}}(p)$.

Proof. We use the same notations as above, but this time we set $\bar{c}_{ji} := \left\lfloor \frac{tc_{ji}}{\varepsilon' L} \right\rfloor$ with $\varepsilon' = \varepsilon/6k$. We know that

$$\frac{\varepsilon' L}{t} \cdot \bar{c}_{ji} \leq c_{ji} < \frac{\varepsilon' L}{t} (\bar{c}_{ji} + 1), \quad (4.4)$$

$$\frac{\varepsilon' L}{t} \bar{y}_j^* \leq y_j^* \leq \frac{\varepsilon' L}{t} \bar{y}_j^* + \varepsilon' L, \quad (4.5)$$

$$x^p + y^p \leq (x + y)^p \leq 2^{p-1}(x^p + y^p) \quad \forall x, y \geq 0. \quad (4.6)$$

The second inequality in (4.6) holds because $f(x) = x^p$ is a convex function. We now bound the value of \bar{x}^* w.r.t. the original cost c .

$$\begin{aligned} \text{val}_p(\bar{x}^*)^p &= \sum_j (y_j^* - c_j^{\text{T}} \bar{x}^*)^p \\ &\stackrel{(4.4)}{\leq} \sum_j \left(\frac{\varepsilon' L}{t} (\bar{y}_j^* - \bar{c}_j^{\text{T}} \bar{x}^*) + \varepsilon' L \right)^p \\ &\stackrel{(4.5)}{\leq} 2^{p-1} k (\varepsilon' L)^p + 2^{p-1} \left(\frac{\varepsilon' L}{t} \right)^p \sum_j (\bar{y}_j^* - \bar{c}_j^{\text{T}} \bar{x}^*)^p \\ &\leq 2^{p-1} k (\varepsilon' L)^p + 2^{p-1} \left(\frac{\varepsilon' L}{t} \right)^p \sum_j (\bar{y}_j^* - \bar{c}_j^{\text{T}} \bar{x}^*)^p \\ &\stackrel{(4.4)}{\leq} 2^{p-1} k (\varepsilon' L)^p + 2^{p-1} \left(\frac{\varepsilon' L}{t} \right)^p \sum_j \left(\frac{t}{\varepsilon' L} (y_j^* - c_j^{\text{T}} x^*) + t \right)^p \\ &\stackrel{(4.5)}{=} 2^{p-1} k (\varepsilon' L)^p + 2^{p-1} \sum_j ((y_j^* - c_j^{\text{T}} x^*) + \varepsilon' L)^p \\ &\stackrel{(4.6)}{\leq} 2^{p-1} k (\varepsilon' L)^p + 2^{2p-2} k (\varepsilon' L)^p + 2^{2p-2} \sum_j (y_j^* - c_j^{\text{T}} x^*)^p \\ &\leq \left((2^{(p-1)/p} + 2^{(2p-2)/p}) k^{1/p} \varepsilon' \text{OPT} \right)^p + \left(2^{(2p-2)/p} \text{OPT} \right)^p \\ &\stackrel{(4.6)}{\leq} \left(\underbrace{(2^{(p-1)/p} + 2^{(2p-2)/p})}_{\leq 6} \underbrace{k^{1/p}}_{\leq k} \varepsilon' \text{OPT} + \underbrace{2^{(2p-2)/p}}_{\leq 4} \text{OPT} \right)^p \\ &\leq ((4 + \varepsilon) \text{OPT})^p. \end{aligned}$$

The polynomial running time of the algorithm follows as before:

$$\begin{aligned}
\text{OPT}(\bar{I})^p &\leq \sum_j (\bar{y}_j^* - \bar{c}_j^\top x^*)^p \\
&\leq \sum_j \left(\frac{t}{\varepsilon' L} (y_j^* - c_j^\top x^*) + t \right)^p \\
&\leq 2^{p-1} k t^p + 2^{p-1} \left(\frac{t}{\varepsilon' L} \right)^p \text{OPT}(I)^p \\
&\leq 2^{p-1} k t^p + 2^{p-1} \left(\frac{t}{\varepsilon'} \right)^p \pi_2(|I|)^p \\
&\leq \left(2^{(p-1)/p} k^{1/p} t + 2^{(p-1)/p} \frac{6kt}{\varepsilon} \pi_2(|I|) \right)^p \\
&\leq \left(2kt + 12kt \cdot \frac{1}{\varepsilon} \cdot \pi_2(|I|) \right)^p,
\end{aligned}$$

and thus

$$\begin{aligned}
\pi_1(|I|) + \pi_3(|\bar{I}|, U(\bar{I})) &\leq \pi_1(|I|) + \pi_3(|\bar{I}|, \pi_2(|\bar{I}|) \text{OPT}(\bar{I})) \\
&\leq \pi_1(|I|) + \pi_3(|\bar{I}|, 2kt \cdot \pi_2(|\bar{I}|) (1 + 6/\varepsilon \cdot \pi_2(|I|))) .
\end{aligned}$$

This completes the proof. \square

Similarly to Proposition 1 in [1], we show that the necessary bounds U and L can be computed if the single-objective problem is tractable:

Lemma 4.4. *Let \mathcal{P} be a problem solvable in polynomial time. Then for all $p \in [1, \infty]$ and all instances I of $\text{CS}^{\mathcal{P}}(p)$ (for both the cornered norm and the ℓ^p -norm) we can compute L such that $L \leq \text{OPT}(I) \leq kL$.*

Proof. Let x^s be an optimal solution to the scalarization of $\text{CS}^{\mathcal{P}}(p)$. By Theorem 4.1 we know that

$$\text{val}_p(x^s) \leq k \cdot \text{OPT}(I) \leq k \cdot \text{val}_p(x^s),$$

thus $L := \frac{1}{k} \cdot \text{val}_p(x^s)$ fulfills the claim. \square

4.2.2 Applications

The pseudopolynomial algorithms for the shortest path problem (SP) and the minimum spanning tree problem (MST) presented in [1] can be used to solve $\text{CS}(\lambda, p)$ as well, as they both compute all (non-dominated) regret vectors (that obey the upper bound U), and the compromise solution always has a non-dominated regret vector.

Note, by the way, that for SP no non-negativity of the edge weights is required. It suffices if the weights are conservative. In [1] the triangle inequality for shortest paths is

used to show that no relevant solutions are lost by discarding regret vectors that violate the upper bound.¹ This inequality holds for general conservative edge weights.

The result for MST uses a translation of the ideal to the origin, a transformation that can also be applied in the context of compromise solutions.

Corollary 4.5. *For $\mathcal{P} \in \{\text{SP}, \text{MST}\}$ and any $p \in [1, \infty)$ and $\lambda \in \mathbb{Q}_{\geq 0}^k \setminus \{0\}$, there is an FPTAS for $\text{CS}_c^{\mathcal{P}}(p)$, and a $(4 + \varepsilon)$ -approximation algorithm for $\text{CS}_\ell^{\mathcal{P}}(\lambda, p)$ for any $\varepsilon > 0$.*

4.3 Greedy and Local Search Approaches for Minimum Spanning Tree

In this section we discuss some local search and greedy approaches to find a compromise solution for M-MST and show why they do not give better approximation ratios than the trivial approximation through scalarization. We only consider the cornered norm here and restrict to the unweighted case again, i.e. $\lambda = \mathbb{1}$. For all algorithms we first analyze the case $p = \infty$, and then extend the statements to $p < \infty$.

4.3.1 Local Search

The local search neighborhood we consider is that of a simple edge swap: Out of a given spanning tree, we remove one edge and add another one to reconnect the two resulting connected components. This leads directly to Algorithm 1.

Algorithm 1: Local Search

Input: Graph $G = (V, E)$, edge weights $c : E \rightarrow \mathbb{N}^k$

Output: Locally optimal spanning tree

- 1 Let T be an arbitrary spanning tree (e.g. optimize w.r.t. one of the criteria).
 - 2 While it is possible to improve the value (in terms of $\text{CS}(p)$) of T by a simple edge swap, do so.
 - 3 Return T .
-

The graph in Figure 4.2 shows that this algorithm can be arbitrarily bad.

The ideal point is $(1, 1)$. The tree with the edge set $\{e_1, e_3, e_4\}$ is a local optimum, since its cost vector is (M, M) , so its value in terms of $\text{CS}(\infty)$ is $M - 1$, and any edge swap results in a tree where in at least one criterion the cost is $M + 1$, so the value increases.

The optimal solution on the other hand consists of the edges $\{e_2, e_3, e_5\}$ and has cost $(2, 2)$ and a value of 1. The ratio between the value of a locally and a globally optimal solution can thus be arbitrarily bad. Of course the local optimum discussed here is not an optimum w.r.t. either of the criteria. However, introducing a third criterion with zero costs on $\{e_1, e_3, e_4\}$ and small costs on $\{e_2, e_5\}$ would make it a feasible starting solution of Algorithm 1 without changing the objective values significantly.

¹The argument is that once a regret vector of a certain path γ violates the upper bound, so will all regret vectors of extensions of γ .

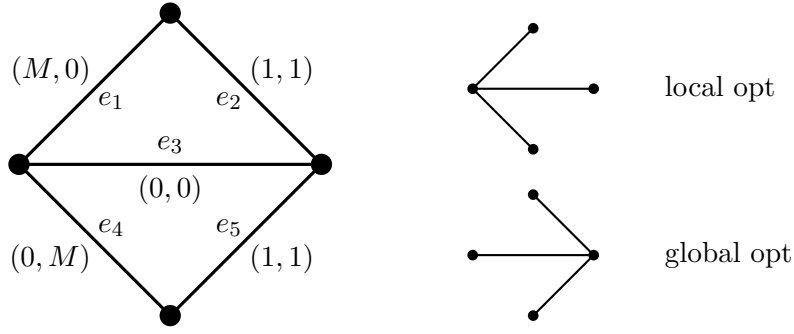


Figure 4.2: Counterexample for local search

To see that local search has no better approximation guarantee than scalarization also for $p < \infty$, choose $M = p + 2$. Then

$$\begin{aligned} \text{val}(\{e_1, e_3, e_4\}) &= \max[(M, M) - (1, 1)] + \frac{1}{p}[(M - 1) + (M - 1)] = p + 3 + \frac{2}{p}, \\ \text{val}(\{e_2, e_3, e_4\}) &= \max[(M + 1, 1) - (1, 1)] + \frac{1}{p}[(M + 1 - 1) + (1 - 1)] = p + 3 + \frac{2}{p}, \\ \text{val}(\{e_1, e_3, e_5\}) &= \max[(1, M + 1) - (1, 1)] + [(1 - 1) + (M + 1 - 1)] = p + 3 + \frac{2}{p}, \\ \text{val}(\{e_2, e_3, e_5\}) &= \max[(2, 2) - (1, 1)] + [(2 - 1) + (2 - 1)] = 1 + 2 = 3, \end{aligned}$$

so $\{e_1, e_3, e_4\}$ is still locally optimal, and simple calculations show that $\frac{p+3+2/p}{3} > \frac{2+2p}{2+p}$, where the latter is the approximation guarantee for scalarization with two criteria.

Improved Local Search. An idea to improve the performance and ensure that it performs at least as good as the scalarization is to start the local search with a scalarization optimum. Empirically, on complete graphs with up to 20 nodes and two (metric) criteria this performs very well. Table 4.1 gives an overview over the performance on random instances that were constructed as follows: For each node, two pairs of coordinates in the square $[0, 1] \times [0, 1]$ were chosen uniformly at random. A complete graph was constructed, where each edge has two edge weights, one for the distance of its two nodes in each of the coordinate pairs.

We used a standard IP formulation to compute the optimum using SCIP 2.0.0 with CPLEX 12 as an LP solver. The optimal value was then compared with the result of the local search. In fact, the local search seems to perform better the bigger the graph becomes. We could not assess the algorithm for greater graphs since solving the exact IP formulation took too long for graphs with more than 19 nodes.

Although the improved local search algorithm empirically performs very well, there are examples showing that no approximation factor better than that for scalarization can be guaranteed. To see this, consider the graph on four nodes as depicted in Figure 4.3.

#nodes	#instances	average gap
10	10000	0.680%
11	10000	0.725%
12	10000	0.771%
13	10000	0.811%
14	10000	0.869%
15	10000	0.860%
16	10000	0.909%
17	10000	0.906%
18	4610	0.921%
19	580	0.937%

Table 4.1: Performance of improved local search on random complete graphs

The ideal point for this instance is $(0, 0)$. Scalarization yields the tree $\{e_1, e_3, e_6\}$, the optimal tree is $\{e_1, e_2, e_6\}$. Of course local search might decide to exchange e_3 for e_2 and terminate with the optimum. But a feasible local move is also to exchange e_1 for e_5 , reaching the locally optimal tree $\{e_3, e_5, e_6\}$. For $p = \infty$ the values are as follows:

$$\begin{aligned}
\text{scalarization opt} &= \text{val}(\{e_1, e_3, e_6\}) = 2M + 2, \\
\text{local opt} &= \text{val}(\{e_3, e_5, e_6\}) = 2M, \\
\text{global opt} &= \text{val}(\{e_1, e_2, e_6\}) = M + 2, \\
&\Rightarrow \frac{\text{local opt}}{\text{global opt}} \xrightarrow{M \rightarrow \infty} 2.
\end{aligned}$$

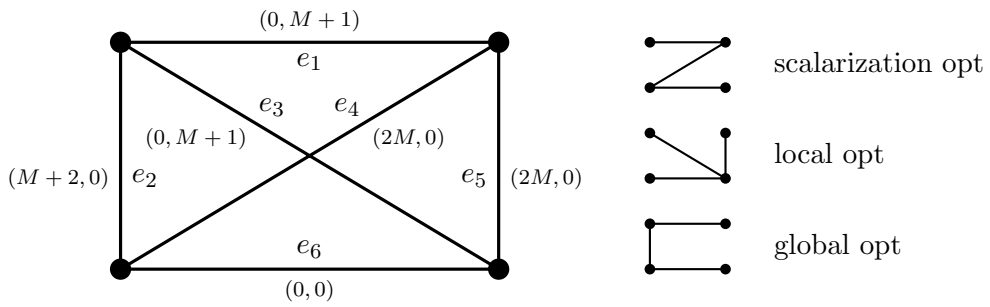


Figure 4.3: Counterexample for improved local search

For $p < \infty$, we modify the edge weights as depicted in Figure 4.4. Then the optimal

trees remain the same, and we get the following values:

$$\begin{aligned}
\text{scalarization opt} &= \text{val}(\{e_1, e_3, e_6\}) = (1 + 1/p)(2 + 2/p)M + 2 + 2/p, \\
\text{local opt} &= \text{val}(\{e_3, e_5, e_6\}) = (1 + 1/p)(2 + 2/p)M + 1/p, \\
\text{global opt} &= \text{val}(\{e_1, e_2, e_6\}) = (1 + 1/p)(1 + 2/p)M + 2 + 3/p, \\
&\Rightarrow \frac{\text{local opt}}{\text{global opt}} \xrightarrow{M \rightarrow \infty} \frac{2 + 2/p}{1 + 2/p} = \frac{2p + 2}{p + 2} = \frac{kp + k}{p + k} \quad (\text{since } k = 2).
\end{aligned}$$

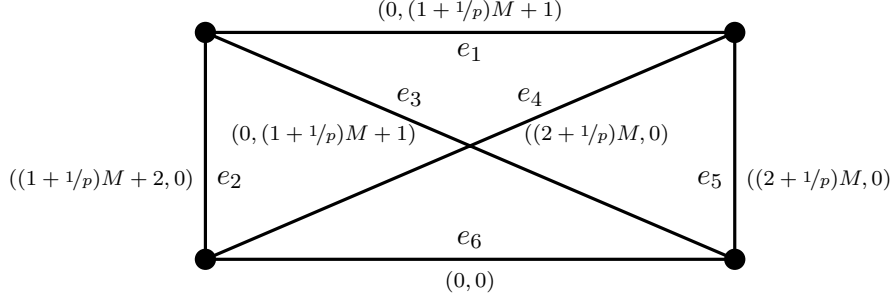


Figure 4.4: Counterexample for improved local search for finite p

In both cases, the locally optimal solution is not better than the approximation factor which is guaranteed for the scalarization.

4.3.2 Moving along a path between two MSTs

For the bicriteria case, an idea closely related to the local search algorithm discussed above is to move from a minimum spanning tree w.r.t. the first criterion to a minimum spanning tree w.r.t. the second criterion by sequentially swapping edges. A similar idea has been used for the constrained minimum spanning tree problem by Ravi and Goemans [6] in 1996. The formal description of the procedure is given in Algorithm 2.

Algorithm 2: Moving along a path between two MSTs

Input: Graph $G = (V, E)$, edge weights $c : E \rightarrow \mathbb{N}^2$

Output: A spanning tree

- 1 Compute MSTs T' and T'' w.r.t. to the two criteria.
 - 2 Compute a sequence $T' = T_1, T_2, \dots, T_\ell = T''$, where T_{i+1} arises from T_i by a single edge swap.
 - 3 Out of the set $\{T_1, \dots, T_\ell\}$, return the tree with the best value.
-

Unfortunately, this algorithm has no constant approximation ratio. To show this, we extend the graph from our first example as depicted in Figure 4.5. In this example, we get from T' to T'' by a single edge swap, so the sequence of trees only consists of the two MSTs. The value (in terms of $\text{CS}(\infty)$) of T' and T'' is M , while that of the compromise

solution T^* is 2, so again the ratio between the value of the computed solution and the optimum is unbounded.

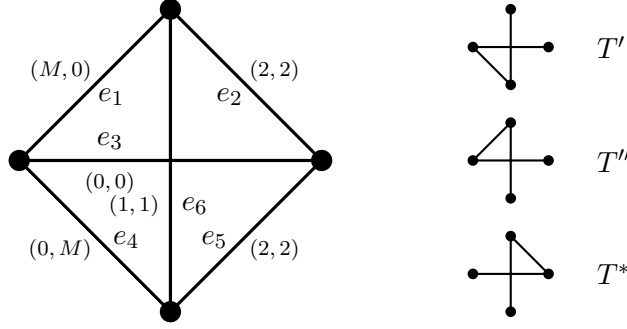


Figure 4.5: Counterexample for the moving along a path between to MSTs

For any constant $p < \infty$, we have

$$\begin{aligned} \text{val}(T') = \text{val}(T'') &= \max[(1, M+1) - (1, 1)] + \frac{1}{p} \cdot [0 + M] = M \left(1 + \frac{1}{p}\right), \\ \text{val}(T^*) &= \max[(3, 3) - (1, 1)] + \frac{1}{p} \cdot [2 + 2] = 2 \left(1 + \frac{2}{p}\right), \end{aligned}$$

so the value of the computed solution is proportional to M , while the optimum is constant, so the approximation ratio is arbitrarily large.

4.3.3 Greedy

Finally we consider a greedy algorithm, which is very similar to Kruskal's algorithm for the single-objective MST. At all times, it aims to minimize the value $\text{val}_p(T)$ of the current partial solution T . The value is defined as

$$\text{val}_p(T) := \max_{i \in [k]} \{c_i(T) - y_i^*\} + \frac{1}{p} \sum_{i \in [k]} (c_i(T) - y_i^*), \quad \text{where } c_i(T) = \sum_{e \in T} c_i(e).$$

Note that $\text{val}_p(T)$ can be negative in the beginning, but in the end it corresponds to the distance of the computed solution to the ideal point y^* . The details of the procedure are listed in Algorithm 3.

Unfortunately also this algorithm performs arbitrarily bad. As an example for this, consider the graph in Figure 4.6. The ideal point is (M, M, M) . For $M \leq \min\{p, \frac{1}{2}(p+1)\}$ it can be shown that the algorithm chooses only dashed edges, resulting in a solution with value $\|(2M, 2M, 2M)\|_p = (1 + \frac{3}{p})2M$. The tree that consists of the solid edges, however, has a value of $\|(1, 1, 1)\|_p = 1 + \frac{3}{p}$. Therefore the greedy solution is at least a factor of $2M$ worse than the optimum.

For $p = \infty$, M can be chosen arbitrarily large, proving that the algorithm has no constant approximation factor. For $p \geq 2$ we get that the algorithm can not have an approximation guarantee better than $p + 1$ and is thus outperformed by scalarization.

Algorithm 3: Greedy algorithm

Input: Graph $G = (V, E)$, edge weights $c : E \rightarrow \mathbb{N}^k$

Output: A spanning tree

- 1 Set $T := \emptyset$.
 - 2 **while** $|T| \neq n - 1$ **do**
 - 3 Choose $e \in E$ such that $\text{val}_p(T \cup \{e\})$ is minimal.
 - 4 Add e to T , remove it from E .
 - 5 Remove all e' from E that close circles in T .
 - 6 **end**
 - 7 Return T .
-

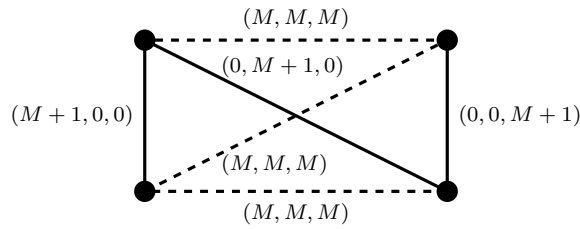


Figure 4.6: Counterexample for the greedy algorithm

For arbitrary p , consider the slightly simpler example from Figure 4.7. The ideal point is $(1, 1)$, and thus we get

$$\begin{aligned} \text{val}(\{e_1\}) &= 0, \\ \text{val}(\{e_2\}) &= \text{val}(\{e_3\}) = 1, \end{aligned}$$

so the greedy algorithm will pick e_1 in the first step, resulting in

$$\begin{aligned} \text{val}(\text{greedy sol'n}) &= \text{val}(\{e_1, e_2\}) = \|(0, 2)\|_p = 2 + 2/p, \\ \text{val}(\text{opt sol'n}) &= \text{val}(\{e_2, e_3\}) = \|(1, 1)\|_p = 1 + 2/p. \end{aligned}$$

Therefore, the greedy algorithm has no approximation ratio better than $\frac{2p+2}{p+2}$, which is the same as that of the trivial approximation in the case of $k = 2$.

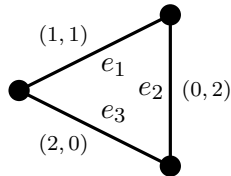


Figure 4.7: Counterexample for the greedy algorithm for arbitrary p

Summing up, in general the greedy algorithm has no better approximation guarantee than scalarization, and for large values of p it is even worse.

Conclusion. The easy approaches presented here do not lead to an approximation ratio better than the trivial one from Section 4.1.

5 Conclusion and Open Questions

The presented concept of compromise solutions fits neatly into the existing concept of Pareto optimality: All compromise solutions are Pareto optimal (for $p < \infty$), and all Pareto optimal solutions are a compromise solution (for sufficiently large values of p). Moreover, an FPTAS for compromise solutions (for sufficiently large values of p) gives an FPTAS for the Pareto set.

These results show that approximating compromise solutions is at least as hard as approximating the Pareto set. An open question is whether it is actually strictly harder.

There are examples for problems with an FPTAS for compromise solutions. However, these algorithms essentially do something very similar to approximating the Pareto set, only to choose an approximate compromise solution from this set in the end.

An open question is whether there is a *direct* algorithm (which does not approximate the Pareto set first) for computing compromise solutions for some problem. Some easy approaches for the minimum spanning tree problem were shown not to have the desired approximation guarantee.

References

- [1] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Approximating min-max (regret) versions of some polynomial problems. In Danny Chen and D. Lee, editors, *Computing and Combinatorics*, volume 4112 of *Lecture Notes in Computer Science*, pages 428–438. Springer Berlin / Heidelberg, 2006.
- [2] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Approximation of min-max and min-max regret versions of some combinatorial optimization problem. *European Journal of Operations Research*, 179(2):281–290, jun 2007.
- [3] W. Chen, M. M. Wiecek, and J. Zhang. Quality utility—a compromise programming approach to robust design. *Journal of Mechanical Design*, 121(2):179–187, 1999.
- [4] W. B. Gearhart. Compromise solutions and estimation of the noninferior set. *Journal of Optimization Theory and Applications*, 28:29–47, 1979. 10.1007/BF00933599.

- [5] C.H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 86–92, 2000.
- [6] R. Ravi and M. Goemans. The constrained minimum spanning tree problem. In Rolf Karlsson and Andrzej Lingas, editors, *Algorithm Theory — SWAT'96*, volume 1097 of *Lecture Notes in Computer Science*, pages 66–75. Springer Berlin / Heidelberg, 1996.
- [7] Ralph Steuer and Eng-Ung Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 1983. 10.1007/BF02591870.
- [8] P. L. Yu. A class of solutions for group decision problems. *Management Science*, 19(8):pp. 936–946, 1973.