

Approximation Algorithms for Combinatorial Multicriteria Optimization Problems

Matthias Ehrgott

Fachbereich Mathematik

University of Kaiserslautern

67663 Kaiserslautern

Germany

e-mail: ehrgott@mathematik.uni-kl.de

fax: (49) 631 29081

Abstract

The computational complexity of combinatorial multiple objective programming problems is investigated. *NP*-completeness and *#P*-completeness results are presented. Using two definitions of approximability, general results are presented, which outline limits for approximation algorithms. The performance of the well known tree and Christofides' heuristics for the TSP is investigated in the multicriteria case with respect to the two definitions of approximability.

Keywords: Multicriteria Optimization, Combinatorial Optimization, *NP*-completeness, Approximation Algorithms

AMS subject classification: 90C29, 90C27

1 Introduction

In this paper we will study combinatorial optimization problems with multiple criteria. Interest in multicriteria optimization (or multiple objective programming, MOP) with respect to theory and applications has been growing in recent years, as can be seen from the literature reviews in [32, 39]. The reason for this is certainly that in real world problems almost always multiple criteria are more appropriate than a single one. For example consider the simultaneous minimization of travel cost, time, and distance in route planning. Although multiple criteria optimization in combinatorial problems has not gained the same attention as in linear or nonlinear problems, the classical problems have been studied, see [36] for a survey. However, many decision problems are combinatorial in nature, so

the necessity of a thorough understanding of combinatorial optimization problems in the presence of conflicting criteria is evident.

This is the direction which we will pursue in this paper. We will study the computational complexity of such problems, and see that multicriteria counterparts of polynomially solvable problems such as the shortest path or the minimum spanning tree problem are NP -hard (Section 2). Due to this fact heuristics have to be investigated. We shall present some general results on the quality of approximation algorithms for combinatorial multiple objective programming problems (CMOP) in Section 3.

Then in Section 4 we especially address the travelling salesman problem (TSP). Since the TSP is NP -complete with one objective, the multicriteria version has the difficulty of the TSP itself plus the difficulty of multiple objectives. In Section 4 we will investigate the performance of the well known approximation heuristics for the symmetric TSP with triangle inequality in the multicriteria case. Results show that in some cases the same performance ratio can be guaranteed.

The rest of the introduction is devoted to a formal definition of combinatorial multiple objective programming problems (CMOP) and a review of the basic notions of computational complexity.

1.1 Combinatorial Multicriteria Optimization Problems

The feasible set of a combinatorial problem is defined as a subset $\mathcal{F} \subseteq 2^E$ of the power set of a finite set $E = \{e_1, \dots, e_m\}$. For example, consider the minimum spanning tree problem. $G = (V, E)$ is a graph with node set V and the edge set E , the feasible set is the set of spanning trees of G , i.e. $\mathcal{F} = \{T \subseteq E : T \text{ is a spanning tree of } G\}$. A combinatorial optimization problem is formulated as follows:

$$\min_{S \in \mathcal{F}} f(S) \tag{P}$$

Typically, in combinatorial optimization only two types of objective functions are considered, namely the sum and the bottleneck objective:

$$f(S) = \sum_{e \in S} w(e), \text{ or}$$

$$f(S) = \max_{e \in S} w(e),$$

where $S \in \mathcal{F}$ and $w : E \rightarrow \mathbb{Z}$ is some weight function. We, too, will only consider these two types of objectives in this paper.

A combinatorial problem can also be formulated in terms of binary variables. For this purpose we introduce a variable x_i for each element $e_i \in E$. Then, a feasible solution $S \in \mathcal{F}$ can be represented by a binary vector $x \in \{0, 1\}^m$ if we define

$$x_i = \begin{cases} 1 & e_i \in S \\ 0 & \text{else.} \end{cases}$$

With this definition $S = \{e_i : x_i = 1\}$. It is therefore equivalent to speak about feasible solutions as subsets of E or about their representations by binary vectors. Accordingly \mathcal{F} will be represented by a subset of $\{0, 1\}^m$. We will make use of the possibility of formulating a combinatorial multicriteria problem in terms of binary variables throughout the paper.

In a multicriteria combinatorial problem several weight functions $w_q : E \rightarrow \mathbb{Z}$ are given, yielding several objective functions f_q , $q = 1, \dots, Q$ of the sum or bottleneck type. The problem is then to solve

$$\text{"} \min_{S \in \mathcal{F}} (f_1(S), \dots, f_Q(S)) \text{"} \quad (\text{CMOP})$$

in the sense of Pareto optimality (or efficiency). A subset $S \in \mathcal{F}$ is called Pareto optimal if there does not exist another feasible solution $S' \in \mathcal{F}$ such that $f_q(S') \leq f_q(S)$ for all $q = 1, \dots, Q$ with strict inequality for at least one of the objectives. The corresponding vector $f(S) = (f_1(S), \dots, f_Q(S))$ is called efficient or non-dominated. The set of Pareto optimal solutions of (CMOP) will be denoted by \mathcal{P} , the set of efficient values by \mathcal{E} throughout the paper. We will denote combinatorial multicriteria problems of the sum and bottleneck type by

$$Q\text{-}\sum P \text{ and } Q\text{-}\max P,$$

respectively. E.g. 3-max *TSP* denotes a travelling salesman problem with three bottleneck objectives. We remark that it is also possible to consider mixed objectives. However, these will not be considered in the current paper. The topic is under research, and for preliminary results we refer to [6]. Below, we will always assume that $w_q(e) \geq 0$ for all $e \in E$ and $q = 1, \dots, Q$.

1.2 Computational Complexity

This section is intended as a brief and informal review of the basic concepts of computational complexity. For a detailed and more formal presentation of the subject we refer to the seminal book of [12].

The theory of computational complexity is formulated for decision problems $D(P)$. Let us consider a single objective combinatorial optimization problem (P). Then the associated decision problem is the following:

Given a constant $k \in \mathbb{Z}$, does there exist a feasible solution $S \in \mathcal{F}$ such that $f(S) \leq k$? D(P)

E.g. given a graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbb{Z}_+$, and a constant $k \in \mathbb{Z}$, does there exist a spanning tree T of G of total weight at most k , i.e. such that $\sum_{e \in T} w(e) \leq k$?

A decision problem $D(P)$ belongs to the **class \mathbf{IP}** of problems if there exists a (deterministic) Turing machine programme requiring a number of steps bounded by a polynomial in

the size of the input which solves $D(P)$. In this sense the class \mathcal{IP} is the class of ‘easy’ problems. In this class decision problems derived from optimization problems usually have the property that there exist algorithms which solve the optimization problem with a number of elementary operations bounded by a polynomial in the input size, usually the cardinality of E . Therefore the associated decision problem can also be solved in polynomial time for every constant k . Members of this class are e.g. (the decision versions of) the minimum spanning tree problem (or more generally, the minimum matroid basis problem), the shortest path problem, and the assignment problem.

A decision problem $D(P)$ belongs to the **class \mathcal{INP}** if there exists a nondeterministic polynomial time Turing machine programme which solves $D(P)$. Loosely speaking this means that it is possible to check in polynomial time whether a solution yields a ‘yes’ answer for $D(P)$. Obviously the class \mathcal{IP} is a subclass of the class \mathcal{INP} . The reverse inclusion is a fundamental open problem.

To understand the relation between \mathcal{IP} and \mathcal{INP} it is necessary to have the notion of transformation. A **polynomial time transformation** of problem $D(P_1)$ to problem $D(P_2)$ is an algorithm A which for every instance of $D(P_1)$ constructs an instance of $D(P_2)$ such that a feasible solution of the instance of $D(P_1)$ yields a ‘yes’ answer if and only if the corresponding feasible solution of $D(P_2)$ constructed by algorithm A yields a ‘yes’ answer. Polynomial transformability is denoted by

$$D(P_1) \propto D(P_2).$$

$D(P_1)$ and $D(P_2)$ are **equivalent** if both $D(P_1) \propto D(P_1)$ and $D(P_2) \propto D(P_1)$ hold. A problem $D(P)$ is called **\mathcal{INP} -complete** if $D(P)$ is in \mathcal{INP} and for all other problems $D(P')$ in \mathcal{INP} $D(P') \propto D(P)$. Finally, $D(P)$ (or a problem which is not a decision problem) is called **\mathcal{INP} -hard** if the latter property is fulfilled, but containment in \mathcal{INP} is not known. Actually, due to the transitivity of \propto , it is sufficient for \mathcal{INP} -completeness or \mathcal{INP} -hardness of a problem $D(P)$ to show transformability to $D(P)$ of only one problem known to be \mathcal{INP} -complete. Among the class of \mathcal{INP} -complete problems are the satisfiability problem, the first problem ever shown to be \mathcal{INP} -complete, and the (decision versions) of the travelling salesman problem, the set covering problem, etc..

Closely related to decision problems are counting problems. The question ‘Does there exist a feasible solution that yields the answer “yes”?’ is replaced by ‘How many feasible solutions yielding “yes” answers do exist?’ The counting problem associated with a combinatorial optimization problem can be formulated as follows:

Given a constant $k \in \mathbb{Z}$, how many feasible solutions $S \in \mathcal{F}$ do exist such that $f(S) \leq k$ do exist? $\#(P)$

If we consider again the example of the spanning tree problem, the question is: how many spanning trees of total weight less than or equal to k does G contain?

As the number of such solutions may be exponential in the size of the input (i.e. in $m = |E|$), it would not be possible to enumerate all of them in polynomial time. However,

writing down the answer to $\#(P)$ may nevertheless be possible in polynomial time. (This is the case in the spanning tree problem). It is immediately clear that counting problems related to NP -complete problems are ‘difficult’. To have a measure for this difficulty of counting problems, the class $\#P$ (read: number P) has been introduced in [37].

A counting problem $\#(P)$ belongs to the class $\#P$ if there exists a nondeterministic algorithm such that for each problem instance the number of ‘guesses’ which yield a ‘yes’ answer is equal to the number of solutions x for which $f(x) \leq k$. Furthermore the longest computation that confirms the ‘yes’ answer for a certain ‘guess’ is supposed to be bounded by a polynomial in the size of the input. A counting problem is **$\#P$ -complete** if it belongs to the class $\#P$ and for all other counting problems $\#(P')$ in $\#P$ there exists a **parsimonious transformation** such that

$$\#(P') \propto_p \#(P).$$

A parsimonious transformation is a polynomial time transformation A such that the number of ‘yes’ solutions of every instance of $\#(P')$ is the same as the number of ‘yes’ solutions of the instance of $\#(P)$ resulting from the transformed problem $A\#(P)$. As an example we mention the problem of counting the number of perfect matchings in a bipartite graph, which is equivalent to computing the permanent of a matrix, and has been shown to be $\#P$ -complete in [37].

2 Computational Complexity of Combinatorial Multicriteria Problems

2.1 (CMOP) with sum objectives

To be able to prove NP - and $\#P$ -completeness results we will refer to two well known problems, the knapsack and the partition problem, which we present here in the formulation of [18].

- **KNAPSACK**: Given $(a_1, \dots, a_n, b) \in \mathbb{Z}^{n+1}$ does there exist an $x \in \{0, 1\}^n$ such that $ax = b$?
- **PARTITION**: Given $(c_1, \dots, c_n) \in \mathbb{Z}^n$, $\sum c_i = 2C$, does there exist a subset $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} c_i = \sum_{i \notin S} c_i$?

From now on we will denote the counting problem associated with a specific decision problem $D(P)$ by using $\#$ as prefix: $\#(P)$.

Lemma 1 1. *PARTITION and KNAPSACK are NP -complete.*

2. *$\#PARTITION$ and $\#KNAPSACK$ are $\#P$ -complete.*

The first part is well known, see [18] and [12]. For the second one we remark that, as indicated in [38], the transformations used in [18] to show *NP*-completeness of *KNAPSACK* and *PARTITION* are parsimonious.

It is easy to see that the following version of *KNAPSACK*, termed 0-1-*KNAPSACK* for distinction, is also *NP* complete, see e.g. [12].

- 0-1-*KNAPSACK*: Given (c_1, \dots, c_n, r) and $(p_1, \dots, p_n, d) \in \mathbb{Z}^{n+1}$, does there exist an $x \in \{0, 1\}^n$ such that $cx \leq r$ and $px \geq d$?

Lemma 2 *0-1-KNAPSACK is NP-complete, #0-1-KNAPSACK is #P-complete.*

Proof:

Let $c_i = p_i = a_i$ $i = 1, \dots, n$ and $r = d = b$. This definition provides a transformation $\text{KNAPSACK} \propto \text{0-1-KNAPSACK}$ which is parsimonious and the result follows. \square

In order to transfer the notions of *IP*, *NP* and *#IP* to (CMOP) we first introduce a decision problem related to (CMOP) in a straightforward manner:

Given constants $k_1, \dots, k_Q \in \mathbb{Z}$, does there exist a feasible solution $S \in \mathcal{F}$ such that $f_q(S) \leq k_q$, $q = 1, \dots, Q$? D(CMOP)

The corresponding counting problem is:

How many feasible solutions $S \in \mathcal{F}$ do satisfy $f_q(S) \leq k_q$, $q = 1, \dots, Q$? #(CMOP)

In the remaining part of this section we will summarize some complexity results for combinatorial multicriteria optimization problems. We start with the following trivial observation.

Remark 1 *If a combinatorial optimization problem 1- $\sum P$ or 1-max P is NP-hard, then the same is true for the multicriteria counterparts Q - $\sum P$, resp. Q -max P , for all $Q \geq 2$.*

An important question concerning complexity is, of course, whether problems which are in class *IP* if a single objective is considered remain within that class in the presence of multiple objectives. The answer to this question is negative except for special cases.

Let us consider a problem without any constraints, i.e. $E = \{e_1, \dots, e_m\}$ and the feasible set is the power set of E , $\mathcal{F} = 2^E$. Therefore any subset of E is a feasible solution and the problem is to find a subset of E of minimal weight. This problem is called the unconstrained combinatorial optimization problem (UCP). With only one criterion this problem is trivial. What about the bicriteria version?

The decision version of the bicriteria unconstrained combinatorial optimization problem 2 - $\sum UCP$

$$\min_{S \in 2^E} \left(\sum_{e \in S} w_1(e), \sum_{e \in S} w_2(e) \right)$$

is as follows: given w_1 and $w_2 \in \mathbb{Z}^m$, two constants k_1 and $k_2 \in \mathbb{Z}$, does there exist a subset $S \subseteq \{e_1, \dots, e_m\}$ such that $\sum_{e \in S} w_1(e) \leq k_1$ and $\sum_{e \in S} w_2(e) \leq k_2$?

If we formulate 2- ΣUCP in terms of binary variables we have $\mathcal{F} = \{0, 1\}^m$. Then the decision problem is as follows: does there exist an $x \in \{0, 1\}^m$ such that $w_1 x \leq k_1$ and $w_2 x \leq k_2$? We define a parsimonious transformation from 0-1-KNAPSACK by letting $w_1 = c$, $k_1 = r$, $w_2 = -p$, and $k_2 = -d$. Thus

$$0\text{-}1\text{-KNAPSACK} \propto_p 2\text{-}\Sigma UCP$$

and hence we have proven:

Proposition 1 $D(2\text{-}\Sigma UCP)$ is NP -complete, $\#(2\text{-}\Sigma UCP)$ is $\#P$ -complete.

From this result, we may argue that NP - and $\#P$ -completeness are a feature of multicriteria optimization problems which occur very often. That this is indeed the case is illustrated by some examples from the literature.

The second example is the shortest path problem, S-T SHORTEST PATH. Given a graph $G = (V, E)$ or a digraph $D = (V, A)$ and two nodes s and t in the respective node set, the feasible set \mathcal{F} is the set of paths from s to t in G or D . The bicriteria shortest path problem is

$$2\text{-}\Sigma \text{ S-T SHORTEST PATH.}$$

Proposition 2 ([30]) $D(2\text{-}\Sigma \text{ S-T SHORTEST PATH})$ is NP -complete, the corresponding counting problem $\#(2\text{-}\Sigma \text{ S-T SHORTEST PATH})$ is $\#P$ -complete.

Proof:

The decision problem clearly is in NP . Therefore NP -completeness is shown by the following transformation 0-1-KNAPSACK \propto 2- Σ S-T SHORTEST PATH given in [30].

From (c_1, \dots, c_n) , (p_1, \dots, p_n) , r , and d we construct the following bicriteria shortest path problem on a digraph $D = (V, A)$ defined as follows:

$$\begin{aligned} V &= \{v_0, \dots, v_n\} \\ A &= \{(v_{i-1}, v_i) : i = 1, \dots, n\} \cup \{(v_{i-1}, v_i)' : i = 1, \dots, n-1\} \\ w_1((v_{i-1}, v_i)) &= c_i \\ w_2((v_{i-1}, v_i)) &= 0 \\ w_1((v_{i-1}, v_i)') &= 0 \\ w_2((v_{i-1}, v_i)') &= p_i. \end{aligned}$$

Then there exists a path P from v_0 to v_n in D satisfying both $\sum_{e \in P} w_1(e) \leq k_1$ and $\sum_{e \in P} w_2(e) \leq \sum_{i=1}^n p_i - k_2$ if and only if there exists a subset $S \subseteq \{1, \dots, n\}$ such that $\sum_{i \in S} c_i \leq k_1$ and $\sum_{i \in S} p_i \geq k_2$ ($i \in S$ corresponds to selecting (v_{i-1}, v_i) and $i \notin S$ to selecting $(v_{i-1}, v_i)'$ in D). Finally we remark that the transformation is parsimonious and therefore the $\#P$ -completeness is also proven. \square

The third problem we consider is the assignment problem. Given a complete bipartite graph $K_{n,n}$ we have the set of all perfect matchings of $K_{n,n}$ as feasible set \mathcal{F} . Again we consider the bicriteria case, i.e. $w_1, w_2 \in \mathbb{R}^{n^2}$ are two costs on the edges of $K_{n,n}$, and we consider 2- Σ ASSIGNMENT.

Proposition 3 ([30, 25]) *The decision problem $D(2\text{-}\Sigma \text{ ASSIGNMENT})$ is NP -complete and the corresponding counting problem $\#(2\text{-}\Sigma \text{ ASSIGNMENT})$ is $\#\text{P}$ -complete.*

Proof:

We present the reduction of a version of PARTITION which is also NP -complete according to [12] to 2- Σ ASSIGNMENT.

- **EQUI-PARTITION:** Let $(c_1, \dots, c_{2n}) \in \mathbb{Z}^{2n}$ be such that $\sum_{i=1}^{2n} c_i = 2C$. Does there exist a subset $S \subseteq \{1, \dots, 2n\}$, $|S| = n$ such that $\sum_{i \in S} c_i = \sum_{i \notin S} c_i = C$.

The reduction $\text{EQUI-PARTITION} \propto 2\text{-}\Sigma \text{ ASSIGNMENT}$ is taken from [30]. We use c_i , $i = 1, \dots, 2n$ to construct the following bicriteria assignment problem. Let \hat{c} be a real number such that $\hat{c} > \max_i c_i$. Let the node set V of $K_{2n,2n}$ be $V = U \cup R$. Define the weights on the edges of $K_{2n,2n}$ as follows:

$$w(u_i, r_i) = \begin{cases} (\hat{c} + c_i, \hat{c} - c_i) & r_i \text{ odd} \\ (\hat{c}, \hat{c}) & r_i \text{ even.} \end{cases}$$

Then there exists a subset $S \subset \{1, \dots, 2n\}$, $|S| = n$ such that $\sum_{i \in S} c_i = C$ if and only if $K_{2n,2n}$ with the above weights contains a perfect matching M with $w_1(M) \leq 2n\hat{c} + C$ and $w_2(M) \leq 2n\hat{c} - C$.

The main result of [37] mentioned in the previous section has been used in [25] to show that $\#(2\text{-}\Sigma \text{ ASSIGNMENT})$ is $\#\text{P}$ -complete. \square

As the single criterion counterparts of all three problems presented here, 2- Σ UCP, 2- Σ S-T SHORTEST PATH, and 2- Σ ASSIGNMENT are solvable in polynomial time, Propositions 1 - 3 are rather disappointing. They show that even bicriteria combinatorial problems are among the hardest problems. All results above have been proved for the case of 2 criteria. However, we remark that the results remain valid in the general case of Q criteria. Just let $w_q(e) := 0$ for all $e \in E$ and $q \geq 3$. Other results highlighting the intractability of the $Q\text{-}\Sigma P$ type concern the number of Pareto optimal solutions. We cite one from [6], see also [16].

Proposition 4 *Let (CMOP) be a multiple criteria combinatorial optimization problem with the properties*

- $|S| = n$ for all $S \in \mathcal{F}$ and
- $w(e_i) = (2^{i-1}, 2^m - 2^{i-1})$.

Then the set of all Pareto optimal solutions \mathcal{P} is equal to \mathcal{F} and $|\mathcal{E}| = |\mathcal{P}| = |\mathcal{F}|$.

Proof:

Let $S \in \mathcal{F}$ be any feasible solution. Then

$$f_1(S) + f_2(S) = \sum_{l=1}^n (w_1(e_{il}) + w_2(e_{il})) = \sum_{l=1}^n (2^{i_l-1} + 2^m - 2^{i_l-1}) = n2^m.$$

Therefore, by the uniqueness of binary representations of numbers, all objective values $f(S_1)$ and $f(S_2)$ are pairwise non-comparable. \square

In Proposition 4 all feasible solutions are also Pareto optimal and they have different objective value vectors. Other results in this respect are known in the literature, see e.g. [17].

2.2 (CMOP) with bottleneck objectives

All the preceding results are for (CMOP) with sum objectives. The situation for bottleneck objectives is completely different. Here, a threshold approach can be used, the consequence being that a problem 2-max P can be solved in polynomial time by Algorithm 2.1, whenever the single criterion bottleneck problem

$$\min_{S \in \mathcal{F}} \max_{e \in S} w(e)$$

can be solved in polynomial time.

The set \mathcal{P} will finally contain Pareto optimal solutions representing the whole efficient set \mathcal{E} . Sometimes such a set is called a complete set of alternatives, see [6] and references therein. The running time of Algorithm 2.1 is bounded by $O(mB(m))$, where $B(m)$ is the time needed to solve the single criterion bottleneck problem 1-max P .

The algorithm can be generalized to the case of Q bottleneck objectives, yielding a bound of $O(m^Q B(m))$ and thus confirming polynomial solvability for any fixed Q . The question whether the problem is also solvable in polynomial time if the number Q of objectives is part of the input is open yet.

3 General Approximability Results

The concept of approximability and performance ratios of algorithms is thoroughly studied in combinatorial optimization, see e.g. [4] for a list of results. The performance ratio $R(S, S^*)$ of a feasible solution S of an instance of a (minimization) problem (P) with respect to an optimal solution S^* is defined as

$$R(S, S^*) := \frac{f(S)}{f(S^*)}.$$

Accordingly a (polynomial time) algorithm A for problem (P) is called an $r(n)$ -approximate algorithm if $R(A(P), S^*) \leq r(|(P)|)$ for all instances of problem (P), where $A(P)$

Algorithm 2.1 : Solving 2-max P

Input: E , weights $w_1(e), w_2(e)$ for all $e \in E$

Output: Set of Pareto optimal solutions \mathcal{P}

$i := 1$

let S_1 be an optimal solution of $\min_{S \in \mathcal{F}} \max_{e \in S} w_1(e)$

$w_{11}^* := f_1(S_1), w_{12}^* := f_2(S_1)$

$\mathcal{P} := \{S_1\}$

1: $E_i := \{e \in E : w_2(e) < w_{i2}^*\}$

if there is no $S \in \mathcal{F}$ such that $S \subset E_i$

then goto 2:

else begin

$\mathcal{F}^{i+1} := \{S \in \mathcal{F} : S \subseteq E_i\}$

let S_{i+1} be an optimal solution of $\min_{S \in \mathcal{F}^{i+1}} \max_{e \in S} w_1(e)$

$\mathcal{P} := \mathcal{P} \cup \{S_{i+1}\}$

$w_{i+1,1}^* := f_1(S_{i+1}), w_{i+1,2}^* := f_2(S_{i+1})$

end

$i := i + 1$, **goto** 1:

2: Delete all dominated solutions from \mathcal{P}

is the solution found by algorithm A and $|(P)|$ denotes the size of a problem instance. Furthermore, $r : \mathbb{N} \rightarrow [1, \infty]$ is an arbitrary function. Note that $r(n) \equiv 1$ means that problem P can be solved (in polynomial time) by Algorithm A . We also remark that $R(S, S^*) = q$ can be equivalently stated as $\frac{f(S) - f(S^*)}{f(S^*)} = q - 1$.

For multicriteria problems we cannot directly transfer this definition, since we have objective value vectors. So we have to decide how to compare these vectors, i.e. a norm is necessary. Throughout the paper we will assume that \mathbb{R}^Q is equipped with a monotonous norm $\|\cdot\|$. A norm is monotonous if, whenever $|a_q| \leq |b_q|, q = 1, \dots, Q$ holds for two elements $a, b \in \mathbb{R}^Q$, then $\|a\| \leq \|b\|$.

Furthermore, since efficient vectors are not unique, we have the options to define approximability with respect to one or to all Pareto optimal solutions. We will use the second approach. We will now provide two possible definitions of performance ratios in multiple criteria optimization. We shall consider algorithms which find **one** solution only.

In the first, we compare the norms of the vectors. Note that the norm of a heuristic solution may be larger or smaller than that of Pareto optimal solutions, wherefore the absolute value is needed.

Definition 1 1. Let $S \in \mathcal{F}$ be a feasible solution of (CMOP) let $S^* \in \mathcal{P}$ be a Pareto

optimal solution. The performance ratio R_1 of S with respect to S^* is defined as

$$R_1(S, S^*) := \frac{|||f(S)|| - ||f(S^*)|||}{||f(S^*)||}$$

2. An algorithm A for (CMOP) is an $r_1(n)$ -approximate algorithm if the solution S found by the algorithm satisfies

$$R_1(A(\text{CMOP}), S^*) \leq r_1(|\text{CMOP}|)$$

for all Pareto optimal solutions $S^* \in \mathcal{P}$.

The second option of comparing the vectors directly, i.e. measuring the norm of the difference between heuristic and Pareto optimal solutions, is taken care of in the following definition.

Definition 2 1. Let $S \in \mathcal{F}$ be a feasible solution of (CMOP), let $S^* \in \mathcal{P}$ be a Pareto optimal solution of (CMOP). The performance ratio R_2 of S with respect to S^* is defined as

$$R_2(S, S^*) := \frac{||f(S) - f(S^*)||}{||f(S^*)||}$$

2. An algorithm A for (CMOP) is an $r_2(n)$ -approximate algorithm for (CMOP) if the solution S found by the algorithm satisfies

$$R_2(A(\text{CMOP}), S^*) \leq r_2(|\text{CMOP}|)$$

for all Pareto optimal solutions $S^* \in \mathcal{P}$.

Note that if any feasible solution S satisfies $R_2(S, S^*) \leq \rho$ for some Pareto solution S^* then also $R_1(S, S^*) \leq \rho$. Therefore we can observe:

Remark 2 An $r(n)$ -approximate algorithm for (CMOP) in the sense of Definition 2 is also an $r(n)$ -approximate algorithm in the sense of Definition 1.

The first definition hints to the idea of using a solution with minimal norm as an approximate heuristic solution for the multicriteria problem. That this is indeed possible is shown in Theorem 1.

Theorem 1 Let S_n^* be a feasible solution of (CMOP) with minimal norm, i.e.

$$||S_n^*|| = \min_{S \in \mathcal{F}} ||S||.$$

Then the performance ratio $R_1(S_n^*, S^*) \leq 1$ for all $S^* \in \mathcal{P}$.

Proof:

$$\frac{||f(S_n^*)|| - ||f(S^*)||}{||f(S^*)||} = \frac{||f(S^*)|| - ||f(S_n^*)||}{||f(S^*)||} = 1 - \frac{||f(S_n^*)||}{||f(S^*)||} \leq 1$$

□

Note that there always exists an S_n^* that is itself a Pareto optimal solution. So two questions arise: The first, whether the ratio 1 is tight, i.e. whether instances can be constructed which achieve this ratio. The second is whether S_n^* can be found.

Example 1 We consider a very simple example of a (CMOP). Let $E = \{a, b, c, d\}$ and let \mathcal{F} consist of all two element subsets of E . Weights are given in the table below, M representing a large number.

e	a	b	c	d
$(w_1(e), w_2(e))$	$(M, 0)$	$(0, M)$	$(1, 1)$	$(1, 1)$

Then the five solutions $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, and $\{c, d\}$ are Pareto optimal, with the efficient vectors $(M+1, 1)$, $(1, M+1)$, and $(2, 2)$. It is easy to see that both $R_1(\{c, d\}, \{a, c\})$ and $R_2(\{c, d\}, \{a, c\})$ approach 1 as M approaches infinity. This observation is true for all l_p norms, $||x||_p = \left(\sum_{q=1}^Q |x_q|^p\right)^{\frac{1}{p}}$ with $1 \leq p \leq \infty$.

For most problems it will be possible to construct examples with a behaviour like this simple one. For specific problems, however, improved results may be possible.

Addressing the second question of whether S_n^* can be found the answer is: it depends on the specific (CMOP) and on the norm chosen. The problem to find S_n^* is

$$\min_{S \in \mathcal{F}} ||f(S)||. \tag{NMP}$$

When $||\cdot||$ is the maximum norm l_∞ the problem is the so called max-ordering problem. We refer to [6] and references therein for results on this type of problems. Here, we will restrict ourselves to a result on a condition when the problem (NMP) is indeed solvable in polynomial time.

Proposition 5 *In the following cases problem (NMP) is solvable in polynomial time.*

- 1- $\sum P$ is solvable in polynomial time and $||\cdot||$ is l_1 , i.e. $||x|| = \sum_{q=1}^Q |x_q|$.
- 1-max P is solvable in polynomial time and $||\cdot||$ is l_∞ , i.e. $||x|| = \max_{q=1}^Q |x_q|$.

Proof:

In both cases the respective problem (NMP) is equivalent to a single criterion problem, which by assumption can be solved in polynomial time.

- The following holds.

$$\begin{aligned}
\min_{S \in \mathcal{F}} \|f(S)\| &= \min_{S \in \mathcal{F}} \left(\sum_{q=1}^Q f_q(S) \right) \\
&= \min_{S \in \mathcal{F}} \sum_{q=1}^Q \left(\sum_{e \in S} w_q(e) \right) \\
&= \min_{S \in \mathcal{F}} \sum_{e \in S} \hat{w}(e),
\end{aligned}$$

where $\hat{w}(e) = \sum_{q=1}^Q w_q(e)$.

- For all $S \in \mathcal{F}$ it holds that

$$\begin{aligned}
\|f(S)\| = \max_{q=1}^Q f_q(S) &= \max_{q=1}^Q \left(\max_{e \in S} w_q(e) \right) \\
&= \max_{e \in S} \left(\max_{q=1}^Q w_q(e) \right) \\
&= \max_{e \in S} \hat{w}(e),
\end{aligned}$$

with $\hat{w}(e) = \max_{q=1}^Q w_q(e)$.

□

Proposition 5 has immediate consequences for such polynomially solvable problems as the shortest path problem, the spanning tree problem and the assignment problem. The well known algorithms that solve these (single objective) problems are 1-approximate algorithms for the multicriteria counterparts.

Whenever problem (NMP) is not solvable in polynomial time, other approximation algorithms have to be considered. Then we may turn to the question on how to approximate S_n^* . But recall that there exists a Pareto optimal solution S_n^* of (NMP) which hence is included in the definition of an $r(n)$ -approximate algorithm anyway. As an important example of such a problem we consider the travelling salesman problem.

4 The Multicriteria TSP

The travelling salesman problem (TSP) can be formulated on the complete graph with n nodes, K_n , as follows. If $w(i, j)$ is the weight of edge $[i, j]$ find a Hamiltonian cycle C of K_n such that either the sum of the weights of the edges in C is as small as possible. In this and the following sections \mathcal{F} will always denote the set of Hamiltonian cycles of a graph. The multiple criteria TSP is defined with $Q > 1$ weights on the edges of K_n . Let $w_q(i, j) \geq 0$, $q = 1, \dots, Q$, $i, j \in \{1, \dots, n\}$ be Q nonnegative weights. Then the problem is

$$\text{“min” } (f_1(C), \dots, f_q(C))$$

subject to $C \in \mathcal{F}$.

where $f_q(C)$ is $\sum_{e \in C} w_q(e)$. The set of Pareto optimal tours will be denoted by \mathcal{P} , the set of efficient weight vectors by \mathcal{E} . The bottleneck TSP will not be considered in this paper. We restrict ourselves to the symmetric TSP with triangle inequality, that is we assume $w_q(i, j) = w_q(j, i)$ and $w_q(i, k) \leq w_q(i, j) + w_q(j, k)$ for all i, j and k in $\{1, \dots, n\}$ and all q in $\{1, \dots, Q\}$.

Before we begin the presentation of the results we will review previous results in the context of TSP. The travelling salesman problem is the most widely studied combinatorial optimization problem, confirmed by a vast amount of literature. For classical topics we refer to [21] and the references therein. Research on the TSP in recent years was mainly focussed on four fields.

1. Polyhedral combinatorics with the aim of solving large scale TSPs exactly by branch and cut methods, see [8, 11, 14, 26],
2. generalized TSPs or TSPs with additional constraints, e.g. [1, 5, 20, 27],
3. polynomially solvable special cases (see the surveys in [2, 13]), and
4. investigation of meta heuristics such as Tabu Search, Simulated Annealing or Genetic Algorithms for the TSP, stimulated by the growing interest in these methods in general. We refer to [9, 19, 23, 24, 33, 34].

Very few papers have treated the multicriteria TSP. Below we list some of the problems investigated and some important results. In [31] a heuristic method for the solution of the TSP with one sum and one bottleneck objective, i.e. 1- \sum 1-maxTSP, where both objectives are obtained from the same cost matrix (w), is presented. No performance ratios are given for this algorithm.

In [15] 2-opt and 3-opt exchange heuristics for the max-ordering TSP (sum objectives) are analyzed. The max-ordering TSP is

$$\min_{C \in \mathcal{T}} \max_{q=1, \dots, Q} (f_1(C), \dots, f_Q(C)).$$

TSPs with two (sum) criteria are solved in [10] by a branch and bound approach. A similar method is suggested in [35] for any number of objectives. A multiple labelling scheme is used to keep track of possible Pareto optimal tours. However, these algorithms are enumerative and exhibit exponential time complexity in the worst case for two reasons: The complexity of the TSP itself and the (possibly) large number of Pareto optimal tours. The cardinality of \mathcal{P} is investigated in [7]. There are two main results on the maximal number of Pareto optimal TSP-tours. For each number n and each number $Q > 2$ there exist distance matrices ($w^q(i, j)$), $q = 1, \dots, Q$ such that all possible TSP-tours are also Pareto optimal, i.e.

$$|\mathcal{P}| = |\mathcal{E}| = \frac{(n-1)!}{2}.$$

For the second result we note that a property holds for almost all graphs on n nodes if the ratio of the number of graphs on n nodes fulfilling that property and the number of all graphs on n nodes approaches 1 as n approaches infinity.

Theorem 2 *For the multiple criteria TSP with $Q \geq 2$ and for almost all graphs on n nodes the following asymptotic lower bound on the maximum number of Pareto optimal tours and efficient vectors holds.*

$$\max |\mathcal{P}| \geq \max |\mathcal{E}| \geq \frac{(n-1)!}{e^{\phi} 2^n}$$

Here $\phi = \phi(n)$ is an arbitrary function such that $\phi(n) \rightarrow \infty$, $\phi(n)/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$.

Due to Theorem 2 we expect an exponential number of Pareto optimal tours for an instance of the multicriteria TSP. Since the problem of finding one of them is already *NP*-hard, that would imply solution of a possibly exponential number of *NP*-hard problems to determine \mathcal{E} . In the following sections we will therefore investigate approximation algorithms for the TSP. We prove that the two well known approximation algorithms for the symmetric TSP with triangle inequality, the tree algorithm and Christofides' algorithm, can be used in the multiple criteria case, too.

The rest of this section is organized as follows. In Section 4.1 we present the two algorithms. Their performance is investigated in Section 4.2, with respect to the two definitions of approximation ratios. In Section 4.3 we give an example.

4.1 Algorithms

There are two well known heuristics for the TSP, which guarantee worst case performance ratios for their solutions, namely the tree and the Christofides' algorithm.

The idea of the tree algorithm in the single criterion TSP is rather simple. First a minimum spanning tree (MST) of K_n is constructed, then each edge of the MST is taken twice, yielding a Eulerian graph G . An Euler tour of G then defines an embedded TSP-tour. The TSP-tour can easily be constructed from the Euler tour in $O(n)$ time. However, in general there are several Euler tours which can be used to find TSP-tours. In the multiple criteria case the algorithm is as follows.

Christofides' algorithm also begins with the construction of a minimum spanning tree. But instead of duplicating the edges of the MST the edges of a minimal weight perfect matching of the complete graph on the odd degree nodes (of which there is always an even number) of the MST are added. Again the resulting graph is Eulerian and a TSP-tour embedded in an Euler tour can be found in $O(n)$ time. The method was first published in [3] and is also analyzed in [28] and [21].

We remark that validity of both algorithms, i.e. the fact that they produce a TSP-tour, can be shown exactly as in the single criterion case. Therefore we refer to the sources above which include these proofs and omit them here.

Algorithm 4.1 : Tree Algorithm for Q - Σ TSP

Input: Distance matrix $(w_q(i, j))$ $q = 1, \dots, Q$

Output: A TSP tour C

- Step 1 Find a spanning tree with minimal normed weight,
 $ST \in \operatorname{argmin} \{ \|f(T)\| : T \text{ is a spanning tree of } K_n \}$
- Step 2 Define G by the nodes of K_n and two copies of each edge of ST ,
 $G := (V(K_n), E)$, $E := E(ST) \cup \{e' : e \in E(ST)\}$
- Step 3 Find an Euler tour EU in G and a TSP-tour C embedded in EU
-

Algorithm 4.2 : Christofides' Algorithm for Q - Σ TSP

Input: Distance matrix $(w_q(i, j))$, $q = 1, \dots, Q$

Output: A TSP-tour C

- Step 1 Find a spanning tree ST with minimal normed weight,
 $ST \in \operatorname{argmin} \{ \|f(T)\| : T \text{ is a spanning tree of } K_n \}$
- Step 2 Define G^* as the complete graph on the odd degree nodes of ST ,
 $G^* := (V^*, E^*)$, $V^* = \{v \in V(G) : v \text{ has odd degree in } ST\}$, $E^* = \{[u, v] : u, v \in V^*\}$
- Step 3 Find a perfect matching PM of G^* with minimal normed weight
 $PM \in \operatorname{argmin} \{ \|f(M)\| : M \text{ is a perfect matching of } G^* \}$
- Step 4 Define G as the union of ST and PM , $G := (V(K_n), E(PM) \cup E(ST))$
- Step 5 Find an Euler tour EU in G and a TSP-tour C embedded in EU
-

Concerning computational complexity it is obvious that the construction of an Euler tour and the embedded TSP tour can be achieved in $O(n)$ time. However, the overall complexity of Algorithms 4.1 and 4.2 depends on the complexity of the problem of finding norm-minimal spanning trees respectively norm-minimal perfect matchings, i.e again problems of type (NMP). In the case of the l_1 -norm both of these problems are polynomially solvable, since they reduce to the same problem with only one criterion (see Proposition 5). In this case the well known algorithms for minimum weight perfect matching and minimum weight spanning tree can be used and both tree and Christofides' algorithm have polynomial time worst case complexity, in fact $O(n^2)$ and $O(n^3)$, respectively.

In case of the l_∞ -norm, however, both problems are known to be *NP*-hard, see [16] and [22]. These problems are known as the max-ordering spanning tree and max-ordering perfect matching problem. Therefore enumeration methods will be needed to solve these subproblems in the two algorithms and exponential running times will occur.

4.2 Approximation Results for the Multicriteria TSP

In view of Theorem 1 there are two problems concerning the approximation algorithms

1. Do Algorithms 4.1 and 4.2 guarantee the theoretical bound?
2. What is their performance for problem (NMP)?

We will address these questions below. The next theorem provides an affirmative answer to the first question.

Theorem 3 *The tree algorithm is a 1-approximate algorithm for the multicriteria TSP, Q - Σ TSP, in the sense of Definition 1, i.e. $r_1(n) \equiv 1$.*

Proof:

Let C be the TSP-tour found by the tree algorithm and let C^* be in \mathcal{P} . We have to show that

$$-||f(C^*)|| \leq ||f(C)|| - ||f(C^*)|| \leq ||f(C^*)|| \quad (1)$$

Since the first inequality is trivial we look at the second. From the algorithm and the triangle inequality it follows that $f(C) \leq 2f(ST) = f(G)$ which by monotonicity of the norm implies

$$||f(C)|| \leq 2||f(ST)|| \quad (2)$$

Due to the choice of ST and because removing one edge from \bar{C} yields a spanning tree we have

$$||f(ST)|| \leq ||f(C^*)|| \quad (3)$$

Combining (2) and (3) implies

$$||f(C)|| \leq 2||f(C^*)|| \quad (4)$$

and the claim holds. □

Theorem 3 shows that the performance ratio guaranteed for the tree algorithm is the same for multiple and single criteria problems. From Section 3 it is obvious that for Christofides' algorithm the $\frac{1}{2}$ approximation ratio cannot be valid for the multicriteria TSP. However, it has the best possible ratio of 1, too. In Section 4.3 we will see that in practice it yields better results than the Tree algorithm.

Theorem 4 *Christofides' algorithm is a 1-approximate algorithm for the multicriteria TSP, Q - Σ TSP, in the sense of Definition 1, i.e. $r_1(n) \equiv 1$.*

Proof:

Let C be the TSP-tour constructed by Christofides' algorithm and let C^* be a Pareto optimal TSP-tour. Again we have to show (1). We denote by $\{i_1, \dots, i_{2m}\}$ the odd-degree nodes of the spanning tree ST in the algorithm as they appear in C^* , i.e.

$$C^* = \alpha_0, i_1, \alpha_1, i_2, \dots, \alpha_{2m-1}, i_{2m}, \alpha_{2m}$$

where $\alpha_i, i = 1, \dots, 2m$ are possibly empty sequences of nodes. We define two perfect matchings on $\{i_1, \dots, i_{2m}\}$. Let

$$M_1 = \{[i_1, i_2], [i_3, i_4], \dots, [i_{2m-1}, i_{2m}]\}$$

and

$$M_2 = \{[i_2, i_3], [i_4, i_5], \dots, [i_{2m}, i_1]\}.$$

Then by the triangle-inequality it follows that $f(C^*) \geq f(M_1) + f(M_2)$. By the choice of PM in the algorithm it is clear that $\|f(M_1)\| \geq \|f(PM)\|$ and $\|f(M_2)\| \geq \|f(PM)\|$ and hence with monotonicity of the norm:

$$\|f(C^*)\| \geq \|f(M_1) + f(M_2)\| \geq \max\{\|f(M_1)\|, \|f(M_2)\|\} \geq \|f(PM)\| \quad (5)$$

The construction of C and the triangle inequality imply that

$$\|f(C)\| \leq \|f(G)\| = \|f(ST) + f(PM)\| \leq \|f(ST)\| + \|f(PM)\|. \quad (6)$$

We observe that (3) holds. Then substituting (3) and (5) into (6) we conclude

$$\|f(C)\| \leq 2\|f(C^*)\|. \quad (7)$$

□

The result of Theorem 4 is weaker than in the single criterion case because we cannot prove that $\|f(C^*)\| \geq 2\|f(PM)\|$ in general. An important special case in which we can indeed prove a better performance ratio than in Theorem 4 occurs when the norm is the l_1 -norm, $\|x\|_1 = \sum_{q=1, \dots, Q} |x_q|$. In this case we have $\|f(M_1) + f(M_2)\|_1 = \|f(M_1)\|_1 + \|f(M_2)\|_1 \geq 2\|f(PM)\|_1$. Using this instead of (5) in the proof of Theorem 4 we have:

Corollary 1 *If the norm of \mathbb{R}^Q is the l_1 -norm Christofides' algorithm provides a TSP-tour C such that*

$$-1 \leq \frac{\|f(C)\|_1 - \|f(C^*)\|_1}{\|f(C^*)\|_1} \leq \frac{1}{2}$$

for all $C^* \in \mathcal{P}$.

Another problem in proving a better worst case bound is that it may happen that there are Pareto optimal TSP-tours, which have a larger norm than the TSP-tour found by Christofides' algorithm. Hence there may occur lower deviations from Pareto optimal tours. However, for the norm minimizing TSP (NMP) this situation is impossible. We state this as a second Corollary.

Corollary 2 *For the problem $\min_{C \in \mathcal{F}} \|f(C)\|$ Christofides' algorithm provides a TSP-tour C such that*

$$0 \leq \frac{\|f(C)\| - \|f(C_n^*)\|}{\|f(C_n^*)\|} \leq 1,$$

where C_n^* is the optimal solution of (NMP). If furthermore the norm of \mathbb{R}^Q is l_1 then Christofides' algorithm is a $\frac{1}{2}$ -approximate algorithm for the norm minimizing TSP (NMP).

For the rest of this section we will restrict ourselves to l_p -norms, i.e. $\|x\|_p = \left(\sum_{q=1}^Q |x_q|^p\right)^{\frac{1}{p}}$ for $x \in \mathbb{R}^Q$, which clearly are monotonous norms. The following lemma will be used in the proof of the main result on approximation algorithms.

Lemma 3 *Let $a, b \in \mathbb{R}_+^Q$ and let $p \geq 1$ then*

$$\left(\sum_{q=1}^Q |a_q - b_q|^p\right)^{\frac{1}{p}} \leq \left(\sum_{q=1}^Q (a_q^p + b_q^p)\right)^{\frac{1}{p}}$$

Proof:

$|a_q - b_q|^p \leq (\max\{a_q, b_q\})^p \leq a_q^p + b_q^p$ holds for all $q = 1, \dots, Q$. \square

Theorem 5 *Let the norm of \mathbb{R}^Q be $\|\cdot\|_p, 1 \leq p \leq \infty$. Then both the tree and Christofides' algorithm are $(2^p + 1)^{\frac{1}{p}}$ -approximate algorithms for the multicriteria TSP, Q - Σ TSP, in the sense of Definition 2. That is, for both algorithms $r_2(n) \equiv (2^p + 1)^{\frac{1}{p}}$*

Proof:

Let C be the TSP-tour found by either the tree or Christofides' algorithm and let $C^* \in \mathcal{P}$ be a Pareto optimal TSP-tour. Then

$$\begin{aligned} \frac{\|f(C) - f(C^*)\|_p}{\|f(C^*)\|_p} &= \frac{\left(\sum_{q=1}^Q |w_q(C) - w_q(C^*)|^p\right)^{\frac{1}{p}}}{\left(\sum_{q=1}^Q (w_q(C^*))^p\right)^{\frac{1}{p}}} \\ &\leq \frac{\left(\sum_{q=1}^Q ((w_q(C))^p + (w_q(C^*))^p)\right)^{\frac{1}{p}}}{\left(\sum_{q=1}^Q (w_q(C^*))^p\right)^{\frac{1}{p}}} \\ &= \left(\frac{\|f(C)\|_p^p + \|f(C^*)\|_p^p}{\|f(C^*)\|_p^p}\right)^{\frac{1}{p}} \\ &\leq \left(\frac{2^p \|f(C^*)\|_p^p + \|f(\bar{C}^*)\|_p^p}{\|f(C^*)\|_p^p}\right)^{\frac{1}{p}} \\ &= (2^p + 1)^{\frac{1}{p}} \end{aligned}$$

where the first inequality follows from Lemma 3 and the second from (4) and (7), respectively. \square

If we let p approach infinity we see that both algorithms are 2-approximate algorithms if $\|\cdot\| = \|\cdot\|_\infty$. This result can be verified directly using $\|x\|_\infty = \max_{q=1, \dots, Q} |x_q|$ in the proof of Theorem 5.

The important special case of $p = 1$ allows again to prove a better result for Christofides' algorithm. We can proceed as in Corollaries 1 and 2 after Theorem 4. This observation allows to replace 2 by $\frac{3}{2}$ in Theorem 5. Setting $p = 1$ we can state Corollary 3.

Corollary 3 *If the norm of \mathbb{R}^Q is the l_1 -norm Christofides' algorithm is a $\frac{5}{2}$ -approximate algorithm (in the sense of Definition 2) for the multicriteria TSP, Q - Σ TSP, i.e. $r_2(n) \equiv 5/2$.*

Let $\rho(p)$ denote the approximation ratios for l_p -norms. Then according to Theorem 5 Figure 1 shows the values of $\rho(p) = (2^p + 1)^{\frac{1}{p}}$.

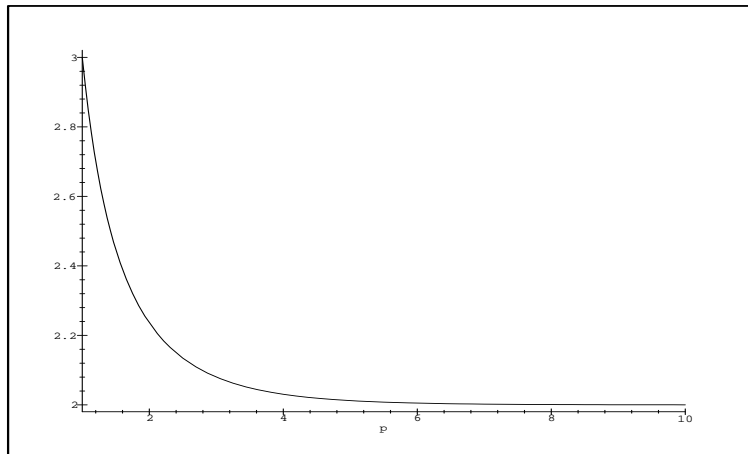


Figure 1: Approximation Ratios for l_p -norms

4.3 An Example

As an example let us consider the complete graph K_6 with $Q = 3$ weights on the edges (see Table 1). In the example we will calculate the exact maximal deviations of all possible TSP-tours the tree and Christofides' algorithm may find to the seven Pareto optimal TSP-tours. We consider l_p -norms with $p \in \{1, 2, \infty\}$. The seven Pareto optimal TSP-tours are listed in Table 2 with their weights and the corresponding normed weights, according to the three norms we consider in this example.

	1	2	3	4	5	6
1	–	(1,57,39)	(6,55,173)	(5,24,6)	(5,19,249)	(5,46,45)
2	(1,57,39)	–	(5,151,354)	(4,126,348)	(6,121,430)	(4,137,25)
3	(6,55,173)	(5,151,354)	–	(5,121,511)	(6,90,76)	(1,117,404)
4	(5,24,6)	(4,126,348)	(5,121,511)	–	(6,34,251)	(8,61,39)
5	(5,19,249)	(6,121,430)	(6,90,76)	(6,34,251)	–	(2,27,328)
6	(5,46,45)	(4,137,25)	(1,117,404)	(8,61,39)	(2,27,328)	–

Table 1: Weights of K_6 in the Example

No.	Pareto Optimal Tour						l_1	l_2	l_∞	Weight
1	1	4	6	5	3	2	1279	936.906	842	(27,410,842)
2	1	4	5	6	3	2	1812	1441.674	1382	(20,410,1382)
3	1	5	6	3	4	2	2364	1936.247	1879	(18,467,1879)
4	1	4	5	3	6	2	1283	923.478	801	(23,459,801)
5	1	3	5	4	6	2	1068	743.590	603	(31,434,603)
6	1	4	3	5	6	2	1464	1085.675	985	(23,456,985)
7	1	3	4	5	6	2	1782	1395.445	1327	(24,431,1327)

Table 2: Pareto Optimal TSP-Tours

For both algorithms we need spanning trees with minimal norm. These are unique and listed for the three norms in Table 3. In Christofides' algorithm it is also necessary to find perfect matchings with minimal norm on the complete graph consisting of the odd degree nodes of the trees listed in Table 3. These matchings are also unique and shown in Table 4.

	Spanning Tree	Norm	Weight
l_1	1 – 2, 1 – 3, 1 – 4, 1 – 6, 3 – 5	634	(23,272,339)
l_2	1 – 2, 1 – 3, 1 – 4, 1 – 6, 3 – 5	435.240	(23,272,339)
l_∞	1 – 2, 1 – 3, 1 – 4, 3 – 5, 4 – 6	333	(26,287,333)

Table 3: Spanning Trees with Minimal Norm

	Perfect Matching	Norm	Weight
l_1	2 – 6, 4 – 5	457	(10,171,276)
l_2	2 – 6, 4 – 5	324.69	(10,171,276)
l_∞	1 – 5, 2 – 6	274	(9,156,274)

Table 4: Perfect Matchings with Minimal Norm

Tables 3 and 4 provide the necessary data to apply the two algorithms. For the tree algorithm the data for the l_1 - and l_2 -norms allow the construction of 24 TSP-tours (derived from all possible Euler tours by short-cutting nodes which are visited twice), 4 of which are Pareto optimal. If the l_∞ -norm is used only 6 TSP-tours can be constructed, one of which is Pareto optimal. Calculating the maximal deviations of these tours from Pareto optimal tours according to Definitions 1 and 2, we get the results of Table 5. Investigating Christofides' algorithm we see that for all three norms 4 TSP-tours can be constructed, one of which is Pareto optimal. Here for approximation in the sense of Definition 1 we distinguish between upper (norm of the heuristic TSP-tour is larger than that of the Pareto

optimal tour) and lower (norm of the heuristic TSP-tour is smaller than that of the Pareto optimal TSP-tour) deviations. Results are summarized in Table 6.

	l_1	l_2	l_∞
Definition 1	74.16	91.75	91.5
Definition 2	74.91	100.16	91.5

Table 5: Deviations Using the Tree Algorithm

		l_1	l_2	l_∞
Definition 1	Upper Deviation	34.55	39.87	55.72
	Lower Deviation	54.82	61.60	67.91
Definition 2	Deviation	55.92	65.93	67.91

Table 6: Deviations Using Christofides' Algorithm

The example shows that although we cannot prove better worst case approximation bounds for Christofides' algorithm in general this method usually will produce better results. This is due to the more sophisticated idea of adding a perfect matching on the odd degree nodes to the spanning tree instead of duplicating it to make the graph G Eulerian. For the l_2 -norm we also see that a worst case error of 1 in the sense of Definition 2 norm sense cannot be proved for the tree algorithm.

5 Conclusions

In this paper we have first studied the computational complexity of combinatorial multicriteria optimization problems. The results obtained clearly emphasize the intrinsic difficulty of (CMOP). Due to these complexity results we argued that heuristics should be investigated to solve (CMOP).

In our paper we have focused attention on algorithms with a guaranteed worst case performance ratio. We provided two possible definitions of approximate algorithms with performance ratio with respect to the set of all Pareto optimal solutions. Some general results were presented showing the limitations of such an approach. The TSP was considered as an example of a problem which is NP -complete in the single objective case, but which, nevertheless, admits approximate solutions up to a certain bound. The results obtained indicate that the performance of the tree and Christofides' algorithm in some cases are as good as in the single objective case.

As pointed out at several points, this direction of research, to the author's knowledge, has not gained much attention. Some further directions of research emanate quite obviously. We will briefly indicate these below.

Another – component-wise – definition of a performance ratio is possible if we consider

$$R_q(S, S^*) := \frac{|f_q(S) - f_q(S^*)|}{f_q(S^*)}, \quad q = 1, \dots, Q$$

The question is: Do there exist problems (CMOP) that admit $(r(n))$ -approximate algorithms in the sense that

$$R_q(A(\text{CMOP}), S^*) \leq r_q(n), \quad q = 1, \dots, Q.$$

either for some or for all Pareto optimal solutions S^* ?

Very little is known about such algorithm up to now. The definition has been used in [29] for bicriteria linear programs and minimum cost flow problems. As an indication of the problems related to such an approach, e.g. for the TSP, note that equations (3) and (5) do not hold component-wise in general.

Other directions of research are the investigation of approximate algorithms for multiple objective versions of polynomially solvable combinatorial optimization problems. Results for approximation algorithms which deliver good approximations to (at least) one Pareto optimal solution have to be considered, too.

The final conclusion is that this study may serve as an incentive to study approximate solutions of (CMOP) problems in the future. Due to the importance of (CMOP) in real world problems, efficient methods which yield (provably) good results will be needed in the future.

Acknowledgement

The author would like to thank Alexander Feldmann for ideas which influenced this study and for carrying out the computations of the example in Section 4.3.

References

- [1] E. Balas. The prize collecting traveling salesman problem II: Polyhedral results. *Networks*, 25(4):199–216, 1995.
- [2] R.E. Burkard, V.G. Deineko, R. van Dal, J.A.A. van der Veen, and G.J. Woeginger. Well-solvable special cases of the TSP: A survey. Technical report, Institut für Matheamtik, TU Graz, 1995. SFB Report 52.
- [3] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon University, 1976.
- [4] P. Crescenzi and V. Kann. A compendium of NP optimization problems. <http://www.nada.kth.se/theory/problemlist.html>, 1997.

- [5] Y. Dumas, J. Desrosiers, E. Gelinas, and M.M. Solomon. An optimal algorithm for the travelling salesman problem with time windows. *Operations Research*, 43(2):367–371, 1995.
- [6] M. Ehrgott. *Multiple Criteria Optimization – Classification and Methodology*. Shaker Verlag, Aachen, 1997.
- [7] V.A. Emelichev and V.A. Perepelitsa. On cardinality of the set of alternatives in discrete many-criterion problems. *Discrete Mathematics and Applications*, 2(5):461–471, 1992.
- [8] R. Euler and H. Le Verge. Complete linear descriptions of small asymmetric traveling salesman polytopes. *Discrete Applied Mathematics*, 62:193–208, 1995.
- [9] C.N. Fiechter. A parallel tabu search algorithm for large traveling salesman problems. *Discrete Applied Mathematics*, 51(3):243–267, 1994.
- [10] R. Fischer and K. Richter. Solving a multiobjective traveling salesman problem by dynamic programming. *Mathematische Operationsforschung und Statistik, Series Optimization*, 13(2):247–252, 1982.
- [11] M. Fischetti. Clique tree inequalities define facets of the asymmetric traveling salesman polytope. *Discrete Applied Mathematics*, 56(1):9–18, 1995.
- [12] M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [13] P.C. Gilmore, E.L. Lawler, and D.B. Shmoys. *The traveling salesman problem*, chapter 4 Well-solved special cases. Wiley Interscience Series in Pure and Applied Mathematics. John Wiley and Sons, Chichester, 1985.
- [14] M. Groetschel and O. Holland. Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming*, A51(2):141–202, 1991.
- [15] A. Gupta and A. Warburton. Approximation methods for multiple criteria travelling salesman problems. In Y. Sawaragi, editor, *Towards Interactive and Intelligent Decision Support Systems: Proceedings of the 7. International Conference on Multiple Criteria Decision Making, Kyoto*, pages 211–217. Springer-Verlag, Berlin, 1986.
- [16] H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52:209–230, 1994.
- [17] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, number 177 in Lecture Notes in Economics and Mathematical Systems, pages 109–127, 1979.

- [18] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- [19] J. Knox. Tabu search performance on the symmetric travelling salesman problem. *Computers and Operations Research*, 21(8):867–876, 1994.
- [20] A.V. Kryazhimskiy and V.B. Savinov. The travelling-salesman problem with moving objects. *Journal of Computer Systems Sciences International*, 33(3):144–148, 1995.
- [21] E.L. Lawler, J.K. Lenstra, A.G. Rinnooy Kan, and D.B. Shmooy, editors. *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley Interscience Series in Pure and Applied Mathematics. John Wiley and Sons, Chichester, 1985.
- [22] U. Lebrecht. Max-linear assignment problems in robot optimization (in German). Master’s thesis, University of Kaiserslautern, Department of Mathematics, 1991.
- [23] W. Lin, J.G. Delgado-Frias, D.C. Gause, and S. Vassiliadis. Hybrid Newton-Raphson genetic algorithm for the traveling salesman problem. *Cybernetic Systems*, 26(4):387–412, 1995.
- [24] R. Mundigl. *Artificial neural nets approaches for the solution of travelling salesman problems (in German)*. Peter Lang, Frankfurt a. M., 1995.
- [25] P. Neumayer. Complexity of optimization on vectorweighted graphs. In A. Bachem, U. Derigs, M. Jünger, and R. Schrader, editors, *Operations Research 93*, pages 359–361. Physica Verlag, Heidelberg, 1994.
- [26] M. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, 6:1–7, 1987.
- [27] M. Padberg and G. Rinaldi. A branch-and-cut approach to a traveling salesman problem with side constraints. *Management Science*, 35:1393–1412, 1989.
- [28] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice Hall, Englewood Cliffs, 1982.
- [29] G. Ruhe and B. Fruhwirth. ϵ –optimality for bicriteria programs and its application to minimum cost flows. *Computing*, 44(1):21–34, 1990.
- [30] P. Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In *Recent advances and historical development of vector optimization*, number 294 in Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, 1986.
- [31] I.K. Sigal. Algorithms for solving the two-criterion large-scale travelling salesman problem. *Computational Mathematics and Mathematical Physics*, 34(1):33–43, 1994.

- [32] R.E. Steuer, L.R. Gardiner, and J. Gray. A bibliographic survey of the activities and international nature of multiple criteria decision making. *Journal of Multi-Criteria Decision Analysis*, 5:195–217, 1996.
- [33] D.M. Tate, C. Tunasar, and A.E. Smith. Genetically improved presequences for euclidian traveling salesman problems. *Mathematics and Computer Modelling*, 20(2):135–143, 1994.
- [34] P. Tian and Z. Yang. An improved simulated annealing algorithm with genetic characteristics and the traveling salesman problem. *Journal of Information and Optimization Science*, 14(3):241–255, 1993.
- [35] C.T. Tung. A multicriteria pareto-optimal algorithm for the traveling salesman problem. *Asia-Pacific Journal of Operational Research*, 11:103–115, 1994.
- [36] E.L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3:83–104, 1994.
- [37] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
- [38] D.J.A. Welsh. *Complexity: Knots, Colourings and Counting*. Cambridge University Press, Cambridge, 1993.
- [39] D.J. White. A bibliography on the application of mathematical programming multiple-objective methods. *Journal of the Operational Research Society*, 41(8):669–691, 1990.