Connectedness of Efficient Solutions in Multiple Criteria Combinatorial Optimization

Matthias Ehrgott and Kathrin Klamroth

Preprint Nr. 265

ISSN 0943-8874
Connectedness of Efficient Solutions in Multiple Criteria Combinatorial Optimization

Matthias Ehrgott\*\  
Universität Kaiserslautern

Kathrin Klamroth\t\  
Universität Kaiserslautern

Abstract

In multiple criteria optimization an important research topic is the topological structure of the set $X_e$ of efficient solutions. Of major interest is the connectedness of $X_e$, since it would allow the determination of $X_e$ without considering non-efficient solutions in the process. We review general results on the subject, including the connectedness result for efficient solutions in multiple criteria linear programming. This result can be used to derive a definition of connectedness for discrete optimization problems. We present a counterexample to a previously stated result in this area, namely that the set of efficient solutions of the shortest path problem is connected. We will also show that connectedness does not hold for another important problem in discrete multiple criteria optimization: the spanning tree problem.

1 General Results

The general multicriteria optimization problem is

$$\min f(x)$$

s.t. $x \in X$

where $f : \mathbb{R}^n \to \mathbb{R}^Q$ and $X \subseteq \mathbb{R}^n$. In the general case $\mathbb{R}^Q$ is ordered by a cone $K$ (see [11] for general results on orders defined by cones). In multiple criteria optimization the notion of optimality is usually replaced by efficiency, since in general different solution values in $\mathbb{R}^Q$ exist which can be considered as "best" solutions of the problem in the sense that they cannot be improved. $x_e \in X$ is called efficient solution if $(f(x_e) - K) \cap f(X) = \{ f(x_e) \}$. The set of all efficient solutions is denoted by $X_e$. Most of the research in this area has been devoted to the case where $K = \mathbb{R}^Q_+$. Then the ordering defined by $K$ is the componentwise order and $x_e \in X$ is efficient if there is no $x \in X$ such that $f_q(x) \leq f_q(x_e), q = 1, \ldots, Q$ where strict inequality holds in at least one case.

To state a general result on the connectedness of $X_e$ we have to introduce the concept of $K$-compactness. A set $Y \subseteq \mathbb{R}^Q$ is said to be $K$-compact if $(y - K) \cap Y$ is compact for all $y \in Y$.

**Theorem 1** ([9]) If $K$ is a closed, convex, pointed (i.e. $z \in K \Rightarrow -z \not\in K$) cone such that $\text{int}(K) \neq \emptyset$ and $Y = f(X)$ is closed, convex and $K$-compact, then $X_e$ is connected.

This result has been generalized in [5] to the case where $\mathbb{R}^n$ and $\mathbb{R}^Q$ are replaced by locally convex spaces. Several authors proved connectedness of $X_e$ for special types of functions [10, 1]. Also several results on the connectedness of the set of weakly efficient solutions are known [10], where $x_{we} \in X$ is said to be weakly efficient if $(f(x_e) - \text{int}(K)) \cap f(X) = \emptyset$.

\*Partially supported by a grant of the Deutsche Forschung Gemeinschaft and the European HCrM Programme

\tSupported by a grant of the Deutsche Forschungsgemeinschaft
In the following we will only consider the case $K = R^d_+$. In this case efficient solutions are often called pareto optimal solutions. We now review the connectedness result for multiple criteria linear programming (MCLP), i.e. the case where $X = \{ x \in R^n | x \geq 0, a^k x \leq b_k, k = 1, \ldots, n \}$ and $f(x) = (c^1 x, \ldots, c^q x), c^k \in R^n$. Obviously Theorem 1 immediately implies that $X_\ast$ is connected in this special case. We will later use MCLP to define connectedness in discrete multiple criteria optimization. Before the general result of Theorem 1 was known the connectedness result for MCLP had been proved by various authors [2, 4, 12].

The most important solutions in linear programming are basic solutions which correspond to vertices of the polyhedral feasible set $X$, and fundamental solutions which correspond to extreme rays of $X$, if $X$ is unbounded. Let $B$ and $F$ denote the sets of basic feasible and fundamental solutions, respectively. Then $z_1, z_2 \in X_\ast \cap B$ are said to be adjacent if they have $m - 1$ basic variables in common and $\alpha z_1 + (1 - \alpha) z_2$ is efficient for all $\alpha \in [0, 1]$. Furthermore $z_\ast \in B \cap X$ and $z_f \in F \cap X$ are said to be adjacent if $z_\ast + \beta z_f$ is efficient for all $\beta \geq 0$. Now let $B$ be the index set of all efficient basic feasible solutions and $F$ be the index set of all fundamental solutions which are adjacent to an efficient basic feasible solution. The main result in [6] is the following

Theorem 2 ([6]) Define a graph $G = (V, E)$ by introducing a node for each index in $B \cup F$ and an edge between two nodes if the corresponding solutions are adjacent. Then $G$ is connected.

Theorem 2 provides the relation between the connectedness of $X_\ast$ in the usual topological sense of Theorem 1 and connectedness of the graph of efficient basic solutions, which will be used for discrete problems.

2 Combinatorial Problems

2.1 The Shortest Path Problem

The connectedness result for multiple criteria linear programming was used in [8] to derive an algorithm for finding all efficient paths from node $s$ to node $t$ in a given directed graph $G = (V, A)$. This graph-theoretical problem can be formulated as a linear program by

$$\min (c^1 x, \ldots, c^q x)$$

s.t. $\sum_j z_{ij} - \sum_j z_{ji} = \begin{cases} 1 & i = s \\ 0 & i \notin \{s, t\} \\ -1 & i = t \end{cases}$

The author defines two paths from $s$ to $t$ to be adjacent if they are contained in two adjacent basic feasible solutions of the above LP. These basic feasible solutions represent spanning trees of the underlying directed graph $G$. From the algorithm it is concluded that the following result holds:

Theorem 3 ([8]) Let $p, p'$ be two efficient $s - t$-paths. Then there exists a sequence of adjacent efficient $s - t$-paths $(p, p_1, \ldots, p_k, p')$.

In the sense of Theorem 2 this means that the graph defined by adjacency of efficient $s - t$-paths is connected. Although the algorithm is correct the conclusion is not true in general, see Theorem 4.

2.2 The Spanning Tree Problem

Another important discrete optimization problem is the spanning tree problem: Given an undirected graph $G = (V, E)$, find $\min (c^1(T), \ldots, c^q(T))$ such that $T$ is a spanning tree of $G$. The linear programming formulation of this problem is:
\[
\begin{align*}
\text{min}(c^1 x, \ldots, c^Q x) \\
\text{s.t.} \sum_{e \in E} x_e &= n - 1 \\
\sum_{e \in E(S)} x_e &\leq |S| - 1 \quad \forall S \subseteq V \quad \text{where } E(S) = \{e = [i, j] \in E | i, j \in S\} \\
x_e &\geq 0
\end{align*}
\]

We define two (efficient) spanning trees to be adjacent if they have \(n - 2\) edges in common. This definition corresponds exactly to the adjacency of efficient basic solutions defined above. We will now formally introduce the efficiency graph corresponding to a spanning tree problem and a shortest path problem.

**Definition 1** Let \(G = (V, E)\) with edge costs \(c^1, \ldots, c^Q : E \rightarrow \mathbb{R}\) be a given graph. The efficiency graph \(E^T(G)\) for the spanning tree problem on \(G\) is defined as follows: Its node set is the set of efficient spanning trees of \(G\). Two nodes are joined by an edge if the corresponding spanning trees are adjacent. Analogously the efficiency graph \(E^{P(s, t)}(G)\) for the shortest path problem on \(G\) with end nodes \(s\) and \(t\) is defined: Its node set is the set of efficient paths from \(s\) to \(t\). Two nodes are joined by an edge if the corresponding paths are adjacent, where adjacency is defined as in Section 2.1.

The question now is: Is the graph defined by adjacency of efficient spanning trees connected?

### 2.3 A Common Counterexample

**Theorem 4** \(E^T(G)\) and \(E^{P(s, t)}(G)\) are not connected in general.

The proof is provided by Example 1 and Lemma 1.

**Example 1** Consider the graph \(G_1 = (V, E)\) given in Figure 1. There are 12 efficient spanning trees of \(G_1\), listed in Table 1. Obviously each efficient spanning tree contains all edges with cost \((0, 0)\). Therefore in Table 1 we only list edges with positive costs.
It is easy to see that \( T_h \) is not adjacent to any other efficient spanning tree.

We will now look at the problem of finding efficient paths from \( s_1 \) to \( s_4 \) in the same graph of Figure 1. Clearly if \( \mathcal{G}^{P(s_1,s_4)}(G_1) \) is not connected the same holds in the directed case: \( G_1 \) can be directed by just orienting each arc from left to right in Figure 1. Lemma 1 then provides the counterexample to Theorem 2.

**Lemma 1** In Example 1, \( \mathcal{G}^{P(s_1,s_4)}(G) \) and \( \mathcal{G}^T(G) \) are isomorphic.

**Proof:**

Every tree \( T_i \in V(\mathcal{G}^T(G)) \) must contain exactly one of the edges \([s_j, s_{j+1}],[s_{j+1}, s_{j+2}],[s_{j+3}, s_{i+4}]\) for each \( j = 1, 2, 3 \). Analogously every path \( P_i \in V(\mathcal{G}^{P(s_1,s_4)}(G)) \) contains exactly one of these three edges for each \( j = 1, 2, 3 \). Now let \( f : V(\mathcal{G}^T(G)) \rightarrow V(\mathcal{G}^{P(s_1,s_4)}(G)) \) be defined by \( f(T_i) = P_i \) if and only if \( \forall j, k \in \{1, 2, 3\} [s_j, s_{j+1}] \in T_i \Rightarrow [s_j, s_{j+1}] \in P_i \). It is easy to see that \( f \) is bijective. Thus it remains to check whether \( T_i \) is adjacent to \( T_j \) if and only if \( f(T_i) \) is adjacent to \( f(T_j) \). By the definition of adjacency of paths it follows immediately that if \( T_i \) is adjacent to \( f(T_j) \). On the other hand, if \( T_i \) is not adjacent to \( T_j \) then \( f(T_i) \) and \( f(T_j) \) differ in at least two of the three subpaths \((s_1, s_2), (s_2, s_3) \) and \((s_3, s_4)\). Thus there can't exist two spanning trees \( T_1 \) and \( T_2 \) with \( f(T_1) \subset T_1, f(T_2) \subset T_2 \) and \( T_1, T_2 \) being adjacent, i.e. having 11 edges in common.

\[ \square \]

### 2.4 Generalization

We will now generalize Example 1: Starting from any graph \( G \) it is possible to construct a graph \( \tilde{G} \) containing \( G \) as a subgraph such that \( \mathcal{G}^T(\tilde{G}) \) is not connected.

Let \( G = (V,E) \) be a graph and \( c : E \rightarrow \mathbb{R}^Q \) the cost-function on the edges. Some definitions are needed. For simplicity we restrict ourselves to the case \( Q = 2 \). Note that the results also hold for \( Q > 2 \).

**Definition 2** 1. A spanning tree \( T^* \) of \( G \) is an extremal efficient spanning tree if there exist \( \lambda > 0 \) such that \( T^* \in \arg\min \{ \sum_{e \in E(T)} \lambda c_1(e) + (1 - \lambda)c_2(e) | T \text{ is spanning tree of } G \} \).

2. A spanning tree \( T \) of \( G \) is a lexicographic minimal spanning tree w.r.t. \( (c^1,c^2) \) if there exists no other tree \( T' \) such that \((c^1(T'), c^2(T')) <_L (c^1(T), c^2(T)) \) where \( <_L \) denotes the "lexicographical smaller" relation on \( \mathbb{R}^2 \). The set of all such trees is denoted by \( T_1 \). Analogously \( T_
is lexicographically minimal w.r.t. $(c^2, c^1)$ if there is no tree $T'$ such that $(c^2(T'), c^1(T')) \prec_L (c^2(T), c^1(T))$. These trees are denoted by $T_2$.

3. Let $T_1, T_2$ be spanning trees of $G$. $T_1$ dominates $T_2$ if $c_i(T_1) \leq c_i(T_2), i = 1, 2$ with strict inequality in at least one case.

4. For a graph $G$ and $v \in V(G)$ let $d_G(v)$ denote the degree of $v$, i.e. the number of edges incident to $v$. Let $V^* \subset V$. Then $\delta_{V^*}(G) := \min_{v \in V^*} d_G(v)$ denotes the minimal degree of nodes not in $V^*$.

Notice that despite Theorem 4 it is known that the set of extremal efficient spanning trees is connected in the sense of adjacency, see [3]. In the following we will construct a graph $G'$ containing $G$ as a subgraph such that, in the corresponding efficiency graph, $\delta_{T_1 \cup T_2}(E^T(G')) < \delta_{T_1 \cup T_2}(E^T(G))$.

Let $T_1, \ldots, T_m$ be the set of efficient spanning trees of $G$. Furthermore let $T_k \in \{T_1, \ldots, T_m\} \setminus T_1 \cup T_2$ such that $\delta_{E^T(G)}(T_k) = \delta_{T_1 \cup T_2}(E^T(G))$. We denote the costs of $T_i$ by $(a_i, b_i)$ and will use $(a_k, b_k) = (x, y)$ for easier distinction. We assume that the numbering of efficient trees is such that the costs are ordered lexicographically, i.e. $a_1 \leq \cdots \leq x \leq \cdots \leq a_1 \leq \cdots a_m$ and $b_1 \geq \cdots \geq y \geq \cdots \geq b_1 \geq \cdots \geq b_m$. Furthermore let $T_i \in \{T_{k+1}, \ldots, T_m\}$ such that $T_k$ and $T_i$ are connected by an edge in $E^T(G)$.

We distinguish two cases and extend $G$ in two different ways:

**Extension 1:**
First let us assume that there exist $n \in \mathbb{N}$ and $0 < \epsilon < \min\{x - a_1, a_1 - x\}$ such that

$$x > \frac{1}{n}a_i + \frac{n-1}{n}(a_1 + \epsilon) \quad (1)$$

$$y > \frac{1}{n}b_i + \frac{n-1}{n}(b_1 + \epsilon) \quad (2)$$

Then $G' = (V(G) \cup \{v'\} \cup \{v_0, \ldots, v_n\}, E(G) \cup \{(v, v_0), \ldots, (v, v_n)\} \cup \{(v_0, v'), \ldots, (v_n, v')\}$ (where $v$ is an arbitrary node of $V(G)$).

Let $C = -(n - 1)(y - b_1 - \epsilon)$ and assign the following costs to the additional edges:

$$c(v, v_i) = (0, 0); \quad i \in \{1, \ldots, n\}$$

$$c(v_0, v') = (a_i - x - \epsilon, C + b_i - b_1 + \epsilon)$$

$$c(v_i, v') = ((i-1)(x - a_1 - \epsilon), C + (i-1)(y - b_1 - \epsilon)); \quad i \in \{1, \ldots, n\}$$

**Lemma 2** If conditions (1) and (2) hold, then $\delta_{T_1 \cup T_2}(E^T(G')) < \delta_{T_1 \cup T_2}(E^T(G))$.

**Proof**
It is obvious that any efficient spanning tree of $G'$ must contain exactly one edge $[v_i, v']$ and all of the edges $[v, v_i]$ together with an efficient spanning tree of $G$. Therefore we consider the set $\{T_{ij} | i \in \{0, \ldots, n\}, j \in \{1, \ldots, m\}\}$ of spanning trees, where $E(T_{ij}) = E(T_j) \cup [v_i, v'] \cup \{[v, v_i] | i = 0, \ldots, n\}$.

Below we list their costs:

$$c(T_{01}) = (2a_1 - x - \epsilon, C + 2b_1 - b_1 + \epsilon)$$

$$c(T_{02}) = (a_1 - \epsilon, C + y + b_1 - b_1 + \epsilon)$$

$$c(T_{0i}) = (a_1 + a_i - x - \epsilon, C + b_i + \epsilon)$$

$$c(T_{ni}) = (a_1 + a_m - x - \epsilon, C + b_1 + b_m - b_1 + \epsilon)$$

$$c(T_{11}) = (a_1, C + b_1)$$
We omitted all trees \( T_{ij}, i \in \{1, \ldots, n-1\}, j \not\in \{1, k, l, m\} \). Then we observe that \( [T_{i1}, T_{i2}, j] \in E(E(G^T(G'))) \) for \( i_1, i_2 \in \{0, \ldots, n\}, j \in \{1, \ldots, m\} \) if \( i_1 \neq i_2 \) and that \( [T_{i1}, T_{i2}, j] \in E(E(G^T(G'))) \) if \( [T_{i1}, T_{i2}, j] \in E(E(G^T(G))). \) It is easy to see that:

- \( T_{ik} \) is dominated by \( T_{i+1,1}, i \in \{1, \ldots, n-1\} \)
- \( T_{nk} \) is dominated by \( T_{11} \) due to (1) and (2)
- \( T_{0k} \) is efficient, since \( \epsilon < x - a_l \)
- \( T_{nl} \) is dominated by \( T_{11} \) since \( \epsilon < x - a_l \)

It follows that \( T_{0k} \) is not connected to any \( T_{ik}, i \in \{1, \ldots, n\} \) and thus there are only edges \( [T_{0k}, T_{0j}] \in E(G^T(G')) \) if \( [T_{i1}, T_{i2}, j] \in E(E(G^T(G))) \) and \( T_{0j} \) is efficient. Therefore \( d_{E^G(G')(G')} \) of \( T_{0k} \) is at least one less than \( d_{E^G(G')(G')} \).

\[ \square \]

**Extension 2:**

In the second case we consider the situation that (1) and (2) do not hold. Then let \( G^* = (V(G) \cup \{v^*, v_1, v_2\}, E(G) \cup \{[v, v_1], [v, v_2], [v_1, v^*], [v_2, v^*]\}) \) where \( v \) is an arbitrary node of \( V(G) \).

We assign the following costs to the additional edges:

\[
c(v, v_i) = (0, 0); \quad i = 1, 2
\]

\[
c(v_1, v^*) = (0, \beta)
\]

\[
c(v_2, v^*) = (a_1 - a_l - \delta, 0)
\]

where \( \beta \geq \max \{\frac{x + y - 2a_l - \delta}{2}, b_1 - b_l\} \) and \( \delta > 0 \) is sufficiently small. Then we can argue as before that all efficient spanning trees of \( G^* \) must be contained in \( \{T_{ij} | i \in \{1, 2\}, j \in \{1, \ldots, m\}\} \) where \( E(T_{ij}) = E(T_{ij}) \cup \{[v, v_1], [v, v_2], [v_1, v^*], [v_2, v^*]\} \). The costs of these trees are listed below:

\[
c(T_{11}) = (a_1, b_1 + \beta) \quad c(T_{21}) = (a_1 - \delta, b_1)
\]

\[
\cdots \quad \cdots
\]

\[
c(T_{1k}) = (x, y + \beta) \quad c(T_{2k}) = (x + a_l - a_1 - \delta, y)
\]

\[
\cdots \quad \cdots
\]

\[
c(T_{1l}) = (a_1, b_1 + \beta) \quad c(T_{2l}) = (2a_l - a_1 - \delta, b_l)
\]

\[
\cdots \quad \cdots
\]

\[
c(T_{1m}) = (a_m, b_m + \beta) \quad c(T_{2m}) = (a_m + (a_1 - a_1 - \delta), b_m)
\]
We observe that:

- By the choice of $\beta$, $T_{11}$ is dominated by $T_{31}$
- $T_{1k}$ and $T_{11}$ are efficient.

Then if $T_{3k}$ is dominated by some other efficient spanning tree of $G^*$ we have the same result as in the first case: $d_{G^*}(G^*)^T(T_{3k})$ is at least one less than the $d_{G^*}(G^*)^T(T_k)$. Otherwise we consider the edge $[T_{1k}, T_{3k}]$ and check conditions (1) and (2):

\[
\begin{align*}
x &> \frac{1}{n} (x + (a_1 - a_1 - \delta)) + \frac{n-1}{n} (a_1 + \epsilon) \\
x &> \frac{x + a_1 - 2a_1 - \delta - \epsilon}{x - a_1 - \epsilon} \\
y + \beta &> \frac{1}{n} y + \frac{n-1}{n} (b_1 + \beta + \epsilon) \\
\beta &> (n - 1)(b_1 - y + \epsilon)
\end{align*}
\]

If we choose $n = \frac{x + a_1 - 2a_1 - \delta - \epsilon}{x - a_1 - \epsilon} + 1$ and $\epsilon > 0$ small enough, conditions (1) and (2) hold. Hence after appropriate renumbering of the efficient trees we have exactly the situation of the first case with $T_{1k}$ in the place of $T_k$ and $T_{3k}$ in the place of $T_i$.

Analogously to Example 1 Extension 1 and Extension 2 can be easily transfered to the shortest path problem by replacing "spanning tree" by "$s-v$-path" and $T$ by $P$ in Definition 2 and in Extension 1 and 2. Then Lemma 2 can be reformulated as follows:

**Lemma 3** If conditions (1) and (2) hold, then $\delta_{P\cup P_2}(E(G^P(s,v')(G'))) < \delta_{P\cup P_2}(E(G^P(s,v)(G)))$.

**Theorem 5** For a given graph $G = (V, E)$ and costs $c^1, \ldots, c^Q : E \rightarrow \mathbb{R}_+$ there exists a graph $\hat{G}_T$ and costs $\hat{c}^1, \ldots, \hat{c}^Q : E(\hat{G}_T) \rightarrow \mathbb{R}_+$ containing $G$ as a subgraph such that $E(G)^T(\hat{G}_T)$ is not connected.

Analogously for a given graph $G = (V, E)$, vertices $s, v \in V(G)$ and costs $c^1, \ldots, c^Q : E \rightarrow \mathbb{R}_+$ there exists a graph $\hat{G}_P$ and costs $\hat{c}^1, \ldots, \hat{c}^Q : E(\hat{G}_P) \rightarrow \mathbb{R}_+$ containing $G$ as a subgraph such that $E(G^P(s,v))(\hat{G}_P)$ is not connected.

**Proof:**

We apply Extension 1 and, if necessary, Extension 2 iteratively. By Lemma 2 (Lemma 3) it is clear that $\delta_{T_1 \cup T_2}(E(G^T(G'))) < \delta_{P\cup P_2}(E(G^P(s,v)(G'))))$ decreases at least in every second step. After finitely many steps we have constructed a graph $\hat{G}$ such that $E(G^T(\hat{G})) < (E(G^P(s,v)(\hat{G})))$ is disconnected.

It should be noted that after application of Extension 1 or 2 $T_{0k}$ is still not lexicographically minimal, i.e. not contained in $T_1 \cup T_2$. The ordering of the spanning trees is without loss generality, since it is always possible to interchange the first and the second cost function. Thus the assumptions of Extensions 1 and 2 are still fulfilled after each iteration.

\[\square\]

### 3 Conclusions

First let us note that shortest path and spanning tree problems are not the only discrete multiple criteria problems for which the set of efficient solutions is not connected in general. The method described in Section 2.4 can also be applied to construct examples of nonconnected efficiency graphs for multiple criteria matroid optimization problems, where the matroid is either a partition or a transversal matroid.
Despite the negative results of Theorem 4 and Theorem 5 we remark that according to our experience a disconnected graph $E^T(G)$ appears only very rarely. We carried out computational tests together with M. Lind from Aarhus University, Denmark [7]. He implemented a program for finding efficient spanning trees based on the connectedness hypothesis. The approach is as follows:

First all extremal efficient spanning trees are found. Then a neighbourhood search is used to find non-extremal efficient spanning trees.

A total of 50 randomly generated graphs with 10 to 50 nodes were tested and no example of a disconnected efficiency graph was found. In these tests we compared the efficient solutions found under the hypothesis of connectedness were compared to all efficient solutions calculated by an enumeration approach.

Therefore we conclude that, although the efficiency graph is not connected in general, a procedure based on connectedness hypothesis yields a very good approximation of the set of efficient spanning trees. In many cases all efficient spanning trees will be found and in many others only few will be missing. On the other hand the approach implemented in [7] is much faster than an enumeration approach to find all efficient solutions. Running times were a within minutes of CPU-time even for larger graphs of 50 nodes, whereas for some graphs with 50 nodes and even for very dense graphs with 20 nodes we were not able to find the set of all efficient spanning trees within 10 hours of computing time.

References


