Minimal paths on ordered graphs

Ulrike Bossong
Dietmar Schweigert

August 1996

Abstract

To present the decision maker's (DM) preferences in multicriteria decision problems as a partially ordered set is an effective method to catch the DM's purpose and avoid misleading results. Since our paper is focused on minimal path problems, we regard the ordered set of edges (E,$\leq$). Minimal paths are defined in respect to power-ordered sets which provides an essential tool to solve such problems. An algorithm to detect minimal paths on a multicriteria minimal path problem is presented.

Keywords

minimal paths, bicriterion path problems, multicriteria analysis, power ordered sets, efficient solutions

1 Introduction

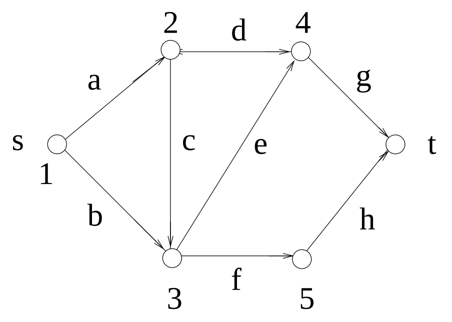
Weighted graphs play an important role in most combinatorial optimization problems. For a longest path on an acyclic directed graph the weight of an edge may symbolize the duration of time for a step in the schedule of a project. For a shortest path in a network the weights may be distances between locations. In general the weights are values in some scales (km, hours) which measure distances or time.
In a lot of decision problems the scaling causes many difficulties and sometimes one comes even to misleading results. If a decision maker presents his choices and preferences then it is often better to describe the preferences in

form of a partially ordered set (see [3])than to measure them in a scale. For instance a tourist may have difficulties to compare some points of interest in a city and cannot give a complete ranking.
In the paper we assume throughout that a decision maker presents his preferences in form of a partially ordered set. Hence we consider ordered graphs $G = (V; E)$ where the set $(V; \leq)$ of vertices and/or the set $(E; \leq)$ of edges is partially ordered. In location problems one needs partially ordered sets $(V; \leq)$. As we will only consider path problems we confine us to $(E; \leq)$ and call such a graph an ordered graph.
In the papers of Schweigert [5], [6] various methods have been developed to solve such problems. The essential tool was the power-ordered set $(P(E); \leq)$. This concept is necessary for the theory to define minimal paths but our approach contains a decision procedure which works without this tools. Besides an algorithm for generating all minimal paths we show that every shortest path problem in graph-theoretical as well as every efficient path problem in multicriteria optimization can be analyzed within our frame work.

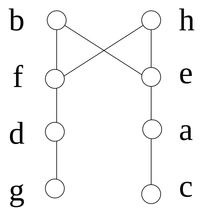
2 Fundamental concepts

For the convenience of the reader we will explain most of our definitions with the help of the following example.

$G = (V, E)$



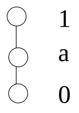
$(E; \leq)$



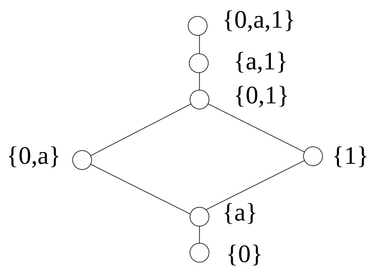
$(E; \leq)$ is presented by a Hasse diagram. For instance g, c are minimal elements in this order which are not comparable. For instance f is smaller than b and h.

Definition 2.1 *Let $\mathcal{E} = (E; \leq)$ be finite ordered set. $\mathcal{P}(\mathcal{E}) = (P(E); \leq_p)$ is called the power ordered set of E if $P(E)$ is the power set of E without the empty set and if the relation $\leq_P$ is defined on $P(E)$ in the following way. $\{a_1, ..., a_n\} \leq_p \{b_1, ..., b_m\}$ if and only if there exists an injective mapping $\pi : \{a_1, ..., a_n\} \to \{b_1, ..., b_m\}$ such that $a_i \leq \pi(a_i)$ for $i = 1, ..., n$ in $(E; \leq)$ and $m \geq n$.*

Example 2.2 *We consider the ordered set $(E; \leq)$ with the Hassediagram*



$(P(E); \leq_p)$ has the following Hassediagram



Lemma 2.3 *The relation $\leq_p$defined on $P(E)$ is an order relation.*

Proof. Obviously $\leq_p$ is reflexive. By the composition of maps it is easy to see that $\leq_p$ is transitive.Let $A \leq_p B$ and $B \leq_p A$. Then there exist injective maps $\pi : A \to B$ and $\varphi : B \to A$ such that $a \leq \pi(a) \leq \varphi(\pi(a)) \cdot \varphi \circ \pi$ is increasing on a finite set and therefore the identity.

Remark 2.4 *As supremas and infimas not always exists in $(P(E); \leq_p)$ the power ordered set of E is usually no lattice.*

Remark 2.5 *The concept of a power ordered set can also be defined for infinite ordered sets. We call the subsets $A, B \in P(E)$ order-equipotent if there exists an injective map $\pi : A \to B$ with $a \leq \pi(a)$ and an injective map $\varphi : B \to A$ with $b \leq \varphi(b)$. In the following we write $\leq$ instead of $\leq_p$.*

3 Properties of minimal paths

All of the graph theoretical optimization problems like shortest paths, minimal matchings and optimal travelling-salesman-tours can be formulated for ordered graphs. We confine us to the problem of finding minimal paths on ordered graphs. Let $G = (V, E)$ be an acyclic directed graph with a source s and a sink t. We present a directed path P by the set $\{a_1, ..., a_k\}$ of its edges.

Definition 3.1 *A directed path $P = \{a_1, ..., a_k\}$ of G from s to t is called a minimal if there exists no path $T = \{b_1, ..., b_m\}$, $m \leq k$ from s to t such that $\{a_1, ..., a_k\} > \{b_1, ..., b_m\}$.*

Remark 3.2 *The principle of dynamic optimization is of course valid. A minimal path P from s to t consists of minimal (a, b)-paths where a, b are vertices of P.*

Example 3.3 *In the example above $P = \{a, d, g\}$ and $Q = \{a, c, e, g\}$ are minimal paths. The path $T = \{a, c, f, h\}$ is not minimal because we have $a \leq a$, $d < f$, $g < h$.*

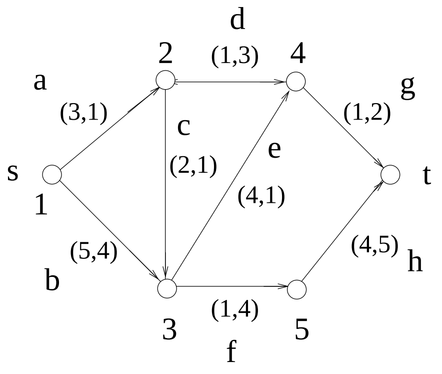
4 Efficient and minimal paths

In multicriteria optimization one considers several objective functions. For instance we may consider for every edge e a vector $(w_1(e), w_2(e))$ where the first component is the distance and the second component is the travel time between the two vertices a, b of the edge e. In a region with mountains the components need not to be in proportion. Let us consider the following example If we have a path $P = \{e_1, ..., e_k\}$ from s to t then the weigth $w(P)$

is defined as

$$w(P) = (w_1(P), w_2(P)) = \left(\sum_{i=1}^{k} w_1(e_i), \sum_{i=1}^{k} w_2(e_i)\right).$$

In our example the path $P = \{a, d, g\}$ has the weight $w(P) = (5, 6)$.
The order relation in vector optimization is defined componentwise [4]. Let P_1, P_2 be two paths with $w(P_1) = (w_1(P_1), ..., w_n(P_1))$ and
$w(P_2) = (w_1(P_2), ..., w_2(P_n))$. We have $w(P_1) \leq w(P_2)$ if and only if



$w_i(P_1) \leq w_i(P_2)$ for every $i = 1, ..., n$ and we have $w(P_1) < w(P_2)$ if and only if $w(P_1) \leq w(P_2)$ and there is some $j \in \{1, ..., n\}$ with $w_j(P_1) < w_j(P_2)$.

Definition 4.1 *A path P from s to t of the vector-weighted graph G is called efficient if there exist no other path Q from s to t such that $w(Q) < w(P)$ (compare [7]p.213 or [7]).*

The aim of this section is to show that to every such vector optimization problem one can present an ordered graph such that every efficient path is a minimal path on this ordered graph $G = (V, E)$.
We define the poset $(E; \leq)$ of the edges of the graph G in the following way. $e \leq h$ if and only if $w(e) \leq w(h)$.
For the graph G from above with vectors as weights we recognize that we get the same poset as in our example in section 2.

Theorem 4.2 *Let G be a vector-weighted acyclic directed graph with a source s and a sink t. Let $w(e) > 0$ for every edge $e \in E$. Let $(E; \leq)$ be the poset which is given by the vector-weights. Every efficient path from s to t is a minimal path from s to t.*

Proof. Let $P = \{e_1, ..., e_m\}$ be an efficient path from s to t. Assume that P is not minimal. Then there exists a path $Q = \{a_1, ..., a_k\}$ $k \leq m$ with $Q < P$. Then we have $a_1 \leq b_{\pi(1)}, ..., a_k \leq b_{\pi(k)}$ for an injective mapping $\pi : \{1, ..., k\} \to \{1, ..., n\}$ and furthermore there exist an index $j \in \{1, ..., k\}$ with $w(a_j) < w\left(b_{\pi(j)}\right)$ or $k < m$. In every case we have $\sum_{i=1}^{k} w(a_i) < \sum_{s=1}^{m} w(b_s)$. Hence we have $w(Q) < w(P)$ which is a contradiction to P is efficient.

Remark 4.3 *The hypothesis $w(e) > 0$ for $e \in E$ does not seriously restrict our statement. We may always add a constant vector to every weight $w(e)$ of an edge without changing the problem essentially.*

5 Comparisons in the power ordered set

We like to avoid the computation of the power ordered set $(P(E); \leq)$ of a partially ordered set $(E; \leq)$. Instead of this we need a procedure which decides wether $A \leq B$ for $A, B \in P(E)$ or $A = \{a_1, ..., a_n\}$ and $B = \{b_1, ..., b_m\}$. It is necessary that $n \leq m$ otherwise $A \leq B$ will not hold.
We construct the following pairs with $B_i \subseteq B$, $i = 1, ..., n$

$$\begin{array}{llll} (a_1, B_1) & \text{where} & B_1 = \{b_{11}, ..., b_{1r}\} & \text{suchthat} \\ \vdots & & a_1 \leq b_{11}, ...\ a_1 \leq b_{1r} & \\ (a_i, B_i) & \text{where} & B_i = \{b_{i1}, ..., b_{is}\} & \text{suchthat} \\ \vdots & & a_i \leq b_{i1}, ...\ a_i \leq b_{1s} & \\ (a_n, B_n) & \text{where} & B_n = \{b_{n1}, ..., b_{nt}\} & \text{suchthat} \\ & & a_n \leq b_{n1}, ...\ a_n \leq b_n & \end{array}$$

For every i we select the largest subset B_i of B which is possible. If $B_i = \emptyset$ for some i then $A \leq B$ does not hold.
But if there exists sets E_i with $|E_i| = 1$, $E_i \subseteq B_i$ for every $i = 1, ..., n$ and $e_k \cap e_l = \emptyset$ for k, l $\in$ { 1,..,n}then we have $A \leq B$.

5.1 Algorithm for comparisons in the power ordered set.

Input. $A = \{a_1, ..., a_n\}$, $B = \{b_1, ..., b_m\}$
Is $A \leq B$?

step 0 **a) if** m > n **then** A $\neq$ B

b) Computation of the $B_i, i = 1 \ldots, n$ **c** Sort A, B $B_i, i, \ldots, n$ such that

step 1 **a)** f.a. i $\leq j; i, j = 1, \ldots, n$ holds: $a_i \not\geq a_j$

b) f.a. i $\leq j; i, j = 1, \ldots, n$ holds: $| B_i | \geq | B_j |$

(Analogous for the b_i)

```
Step 1 (Initialization)
    a) for $i = 1, ..., n$ do
    $k_i \leftarrow |B_i|$
    $x_i \leftarrow 1$ (elementindex of the set $B_i$)
    $e_i \leftarrow B_i[x_i]$
    (the $x_i$th element of $B_i$ is assigned to $e_i$) od

    b) $x_0 \leftarrow 0$,
    $k_0 \leftarrow 0$,
    $B_0[1] \leftarrow 0$,
    $e_0 \leftarrow B_0[1]$,
    $l \leftarrow 0$,
    $L \leftarrow \emptyset$

Step 2 if $B_i = \emptyset$ then stop; $A \not\leq B$ fi

Step 3 a) if f.a. i,j $\in \{1, \ldots, n\}; i \neq j$ gilt $e_i \neq e_j$ then stop; A $\leq$ B fi

    b) l $\leftarrow$ min $\{j \in \{1, \ldots, n\} \mid \exists i \in \{1, \ldots, n\} : e_j = e_i\}$
    L $\leftarrow \{i \in \{1, \ldots, n\} \mid e_i = e_l\}$
    $i_{max} \leftarrow$ max(L)
    L $\leftarrow$ L $\setminus \{i_{max}\}$
    index $\leftarrow$ 0

    c) for i $\in$ L do                                                    d)
       c1) if $x_i = k_i$ then
       index $\leftarrow$ index + 1
       L $\leftarrow$ L $\cup$ { $i_{max}$ } $\setminus$ { i }
       fi
       if $x_i \neq k_i$ then
       $x_i \leftarrow x_i + 1$
       $e_i \leftarrow B_i[x_i]$
       if $i = i_{max}$ then index $\leftarrow$ index - 1 fi
       fi
       od
```

```
if  index $\geq 1$ then
L $\leftarrow$ L $\setminus$ max(L); index $\leftarrow 0$
for j = min(L) to max(L) do
d1) if  index $\leq 1$ then
if $x_j \leq 1$ then
index $\leftarrow$ index + 1
L $\leftarrow$ L $\cup$ { max(L) }
else
d2) $x_j \leftarrow 1$
k $\leftarrow$ element{ k $\in \{1, \ldots, n\}$ , $k \neq j \mid e_k = B_j[x_j]$ }
if $k < j$ then index $\leftarrow$ index + 1 fi
if $k \neq k_k$
then
$x_k \leftarrow x_k + 1$; $e_k \leftarrow B_k[x_k]$; $e_j \leftarrow B_j[x_j]$
else  if $x_j \neq k_j - 1$
then $x_j \leftarrow x_j + 1$; goto d2)
else j $\leftarrow$ k; goto d1)
fi
fi
else A $\not\leq$ B
fi od fi
goto a)
```

(This algorithm is used as a subprocedure in the following section.)

Example 5.1 *If we take the partially ordered set $(E, \leq)$ of section 2, the algorithm stops for $A = \{a, d, g\}$, $B = \{a, c, f, h\}$ with $e_1 = a$, $e_2 = f$, $e_3 = h$ and the positive result $A \leq B$. For $A = \{a, d, g\}$, $B = \{a, c, e, h\}$ it stops with the result "$A \leq B$ does not hold".*

6 Generating all minimal path

In the following algorithm we use labels for every vertex of the graph. Every label of a vertex i is of the form

$$[(e_\mu, \ldots, e_\nu), (j, h_k)]_k$$

where $(e_\mu, ..., e_\nu)$ is the series of edges of a path from 1 to i. The preceding vertex j of the veretex i in this path is stored in the pointer (j, h_k). h_k denotes the label of j which is used for this path. k is the index of the label. The vertices may have temporary or permanent labels

6.1 Algorithm.

Step 0 Assign to the vertex 1 the temporary label $[(-), (-,-)]_1$

Step 1 If the set of all temporary labels is empty goto 3. Otherwise determine the labels with the least number i of the vertices. Among them determine the label with the least index. Let this be the label of the vertex i with the index k. Put this as a permanent label.

Step 2 While there exists a vertex $j \in V$ with $e_\eta = (i, j)$, $e_\eta \in E$, do

- **a)** Assign to the vertex j the temporary label $[(e_\mu, ..., e_\nu, e_\eta), (j, h_k)]_k$
- **b)** Eliminate all labels of the vertex j which presents a non minimal path goto 1

Step 3 Stop. All permanent labels of the sink t present a minimal path from s to t.

In our example the algorithm stops with the permanent labels

$$\left[(a,d,g), \left(4, \left[(a,d), \left(2, [(a), (1, [(-), (-,-)]_1)]_1\right)\right]_1\right)\right]_1$$

and

$$\left[(a,c,e,g), \left(4, \left[(a,c,e), \left(3, \left[(a,c), \left(2, [(a), (1, [(-), (-,-)]_1)]_1\right)\right]_2\right)\right]_2\right)\right]_2$$

These are both efficient (in usual they are not).

References

[1] Hansen, P.: Bicriterion path problem, Lecture Notes in Economics Math. Syst. 177, Springer Heidelberg 1987 109-127

[2] Martins, E.Q.V.: On a multicriterion shortest path problem, Eur.J.O.R. 16 (1984) 236-245

[3] Roubens, M., Vincke, P.: Preference Modelling, Lect. Notes in Economics Math. Sys. Springer Heidelberg 1985

[4] Schweigert, D.:Vector-weighted matchings, Combinatoric Advances (ed. C. J. Colbourn, E. S. Mahmoodian) Kluwer 1995, 267-276

[5] Schweigert, D.: On trees and clusters on ordered graphs, to appear

[6] Schweigert, D.: Ordered graphs and minimal spanning trees, to appear

[7] Steuer, R.E.: Multiple criteria optimization: Theory, Computation and Application, Wiley, New York 1986

Prof. Dr. D. Schweigert
Fachbereich Mathematik
Universität Kaiserslautern
Postfach 3049
D 67653 Kaiserslautern

U. Bossong
Fachbereich Mathematik
Universität Kaiserslautern
Postfach 3049
D 67653 Kaiserslautern