Yugoslav Journal of Operations Research
5 (1995), Number 5, 79-93

# MULTICRITERIA OPTIMUM PATH PROBLEMS

Dimiter IVANCHEV

*Institute of Applied Mathematics and Computer Science, Technical University of Sofia, P.O. Box 384, 1000 Sofia, Bulgaria*

Dimitris KYDROS

*Department of Applied Informatics, University of Macedonia Egnatias 156, P.O. Box 1591, 54006 Thessaloniki, Greece*

**Abstract:** We consider the problem of finding paths in a network between two certain nodes, when some criteria like *cheap, fast, wide, reliable* etc. have been posed. The problems stated below, are multicriteria and are solved by means of lexicographical ordering of the criteria. The proposed algorithms are polynomial ones.

**Keywords:** Multicriteria optimization, path problems, lexicographical order.

## 1. INTRODUCTION AND MOTIVATION

In the world of transportation, maybe the most commonly met question (from the traveller' s point of view) is something like:

" Which is the best way (route) to travel from one place (the origin) to another (the destination)? ".

This question implies that of course there exists a set of possible (or feasible) routes such that connect the origin to the destination, from which to choose. It also implies that the traveller' s final choice depends on certain features that characterize each one of the routes, like the money spent on fairs or gasoline, the time consumed, the possible pleasure that is gained traversing one route and so on.

Clearly, it is easy to assign a positive real number to features like "money spent" or "time consumed". However, this is not possible for features like "pleasure" where we can assign values like "good" or "bad" or "beautiful" and so on. Furthermore, since each person defines, say, "good" in his/her personal attitudes, it seems true that "if you ask

100 different persons why they chose one specific route, you might probably get 100 different answers".

In the case of air transportation, things are not much different. Here there are airports in cities and certain airline companies that offer connections between some of those cities. Each one of these connections can be assigned to one flight descriptor like LH1234 (for a Lufthansa flight) or OA4321 (for an Olympic Airways flight). Each one of these flights operate in a frequency, say Monday, Tuesday, Friday and departs at a specific departure time from the origin to arrive on a precalculated arrival time to the destination. There are also fares, classes, aircraft types and other features known in advance for each flight.

Of course, for every pair of cities, there exists one route that connects them, either direct (i.e. non-stop) or via one or more intermediate cities. In the last case, the total money spent is at least the sum of fares, the total time is at least the sum of the distinct travel times and so on. Additionally, other variables are created, such as the time spent on airports waiting for the next flight to depart and the total time consumed (actual travel time plus waiting time).

It is apparent that the traveller will try to minimize the total money spent, or will place an upper bound (a budget) on it. However, one cannot easily say the same for the other variables. In the case of total travel time, it is known that a longer route might be cheaper or a non-stop flight might be more expensive than one with intermediate stops. As another "real world" example, consider the case in which a client is interested in transferring goods from an origin to a destination. If these goods are, say, clothes, then there is no problem but if these are computer motherboards or medicine, he might be interested in keeping them "on ground", because of the potential danger that the shakings of a flight might harm them.

Generally, the traveller needs to choose from a set of routes where:

- The total cost is low or does not exceed his budget.;
- The total travel time (from the moment he leaves the origin until he is at the destination) is low or should not exceed an upper bound (although this is not always the case; if one is, for example, a tourist);
- There is a departure date (and time) after which the traveller wants to fly and an arrival date (and time) before which he must arrive;
- The total number of intermediate stops (if any) must be low or at least under an upper bound;
- There can be intermediate stops that a traveller wants to avoid ( for example nobody wants to fly to his destination with an intermediate stop in a country which is at war);
- There can be airline companies that should be avoided due to previous bad experience or higher levels of danger and others that are preferred because of, for example, frequent flier programs;

- The total waiting time on airports might have an upper bound too;
- Intermediate stops in expensive cities easily increase the total cost (for example it is cheaper to spend one night in Moscow than in Tokyo)

It is quite difficult for the traveller to form the above mentioned set of routes by himself because of the large number of airline companies and the much larger number of different flights. The airline companies do offer printed schedules, but, of course, they print only their flight – or combinations with flight offered by other companies with which they cooperate. In the modern Computerised Reservation Systems (CRSs) there exists a possibility of creating such choice sets, although one cannot place bounds and other preferences – this is done manually. Another problem with CRSs is that they are owned by one or more airline companies, so the same problem as that with the printed flight schedules arises. Finally, one should not forget, that access to such systems is permitted only through a travel agent, who certainly has to protect his own interests. This means that the propositions made by agents are not always the best for their clients (!).

After the above discussion, it seems that the Computer–Aided Formation of such a choice set is quite interesting. The underlying problem is obviously a multicriterial problem which involves creating paths in a network, as it will be formally defined in the next section.

## 2. THE MODEL

If we stick to the example of air transportation, we can model the situation as follows: to each airport we assign a vertex $V_i$; if it is possible to fly directly from $V_i$ to $V_j$ we denote it by an arc like $(i,j)$; if we are interested in such phenomena like "cost", "comfort", "airline company" etc., we can split each arc $(i,j)$ into two or more parallel arcs, which for example might mean "LH business class" or "LH tourist class" etc. We also consider the time phenomenon so that we can substitute each node by two or more nodes, with the following meaning: $V_i$ corresponds to the $i$th airport today, $V_i'$ and $V_i''$ correspond to the $i$th airport tomorrow and the day after tomorrow and so on. These three nodes can be linked by arcs, like $(i, i')$, $(i', i'')$ and in this way we obtain one time expanded network which seems to be very large due to the big number of airports, airline companies and days when the traveller can book a flight. Further, we will be interested in paths between two given points in the graph – start point and target point, which are "best" in different meanings: cheapest, with least transit stops, most reliable, most preferable etc.

The structure of the network, Figure 1, is changeable, which means that if all tickets for a certain flight are sold, the corresponding arc is not available for the commuter. Further we shall formulate and solve some optimization problems in such networks.

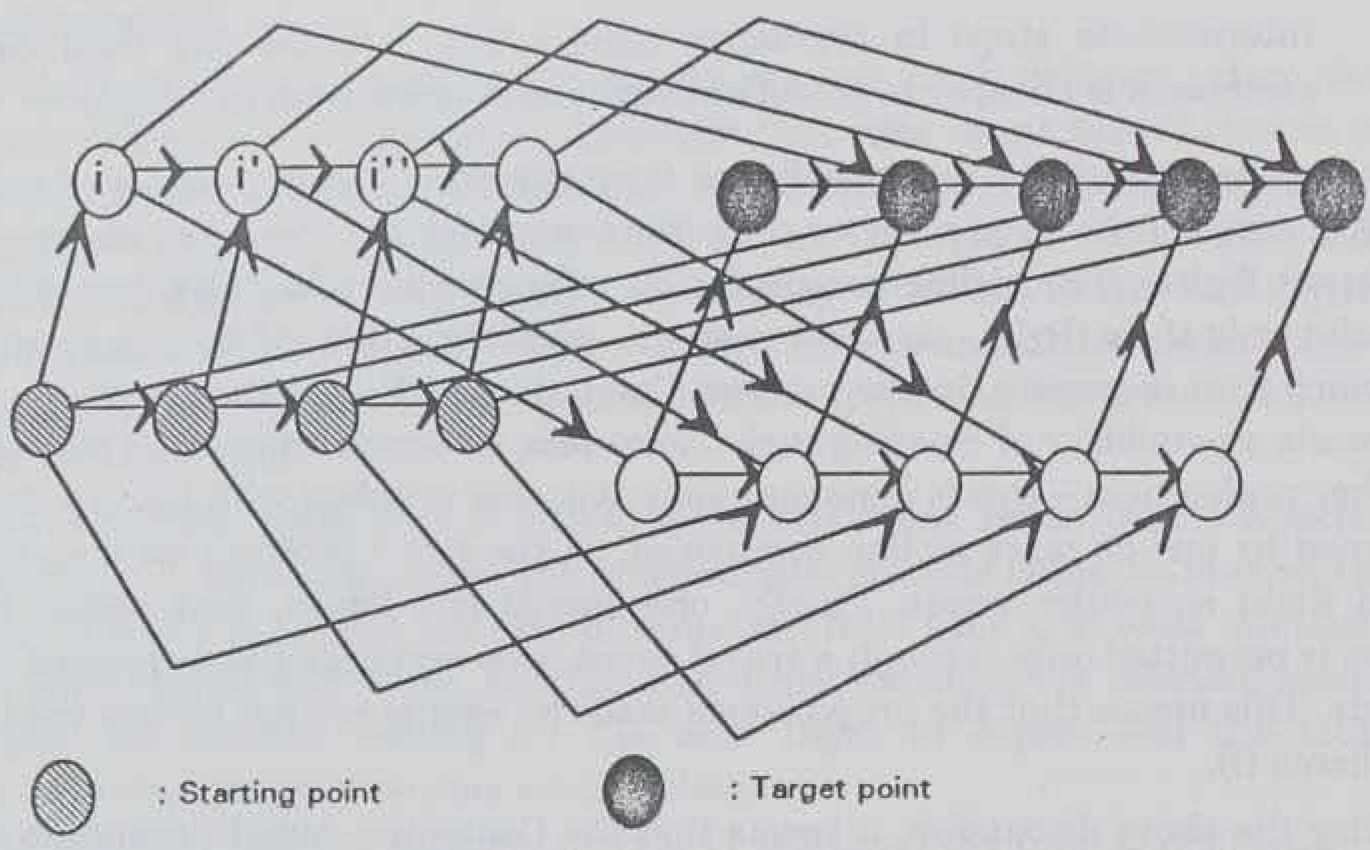



Figure 1.

## 3. MULTICRITERIA ANALYSIS OF PATHS IN A GRAPH

Let us consider a directed connected and finite graph $G = (V, A)$ with a set of vertices (nodes) $V = \{ V_1, V_2, \ldots, V_n \}$ and a set of arcs (links) $A = \{ A_1, A_2, \ldots, A_m \}$. To each arc $A_k = (i, j)$ we assign some weights (non-negative numbers like cost, time etc.) as follows:

- $c_k$ – cost for traversing $A_k$ ;
- $t_k$ – time units for traversing $A_k$ ;
- $p_k$ – reliability of $A_k$ , i.e. the probability that $A_k$ is still working, still available, not closed etc. ;
- $w_k$– capacity of $A_k$, i.e. the largest flow that can traverse the arc per time unit ;
- etc.

Now we define the node $V_1$ as the start point and $V_n$ as the target point and we will consider all paths starting from $V_1$ and ending at $V_n$. We denote the set of all these paths by $PS(A)$. For each path $W$, we define as usual the cost and time for traversing it as

$$c(W) := \sum \{ c_k \,|\, A_k \in W \} \quad \text{and} \quad t(W) := \sum \{ t_k \,|\, A_k \in W \} ,$$

respectively. We define the reliability as

$$p(W) := \prod \{ p_k \,|\, A_k \in W \},$$

but if we take the logarithms of both sides and denote $\ln (p_k)$ by $-r_k$ , we will obtain

$$r(W) := \sum \{ r_k \,|\, A_k \in W \} = -\ln [ p(W) ] = \sum \{ -\ln (p_k) \,|\, A_k \in W \}$$

as a measure of the reliability of $W$. This means that the smaller $r(W)$ is, the more reliable the path $W$ is.

We can define the width of the path $W$ as:

$$w(W) := \min \{ w_k \mid A_k \in W \},$$

but if we denote $-w_k$ by $b_k$, the weight

$$b(W) := \max \{ b_k \mid A_k \in W \}$$

will be a measure for the width of $W$ (the smaller $b(W)$ is, the wider $W$ is).

In this way we have formed four reasonable criteria for each path in the graph in the sense that the smaller each one of them is, the "better" the path is. Thus, we shall try to minimize all of them.

There is no reason to search for the path which is "best" for all of these criteria, since they are contradictable by nature. For example the shorter the path is, the more expensive it might be. A reasonable way would be to order all criteria according to their importance. For this purpose we consider the relation *lexicographical order.*

Given are two vectors $x = (x_1, x_2, \dots x_k)$ and $y = (y_1, y_2, \dots y_k)$. We call $x$ *lexicographically smaller* than $y$ if for some $i$ the following holds:

$$x_1 = y_1,\ x_2 = y_2,\ \dots\ , x_i = y_i,\ x_{i+1} < y_{i+1}$$

and we denote it by $x \prec y$, $y \succ x$ or $x = \text{lexmin}(x, y)$. If $x = y$, we call $x$ lexicographically smaller than $y$ and $y$ lexicographically smaller than $x$. It is easy to show that this is a relation of order (transitive, antisymmertic, reflexive). It follows that we can use here all properties of the relation $\le$ or $\ge$ for the real numbers.

This relation generalizes the relation $\le$ (or "less", or "min"), since the last one can be obtained if the vectors $x$ and $y$ are one–dimensional.

Before formulating any optimization problem, let us now notice that the criteria mentioned above are of two different types: $c(W)$, $t(W)$ and $r(W)$ are sums of the arc weights from $W$ (we shall call them of *type one*, or *additive* criteria), $b(W)$ is a maximum (or minimum) of the arc weight of $W$ (we shall call it of *type two*, or *minimax* criterion).

Further, we shall consider more than one criterion. Instead of enumerating them like $f_1(W)$, $f_2(W)$ etc., we shall preserve the notation mentioned above in order to identify them easier and to make the explanation much clearer.

The problem statements we are interested in are:

$$\text{lexmin} \{ (c(W), t(W)) \mid W \in PS(A) \}, \tag{P1}$$

which means that we are looking for the cheapest path (criterion $c(W)$) and among all cheapest paths we are searching for the shortest one (criterion $t(W)$).

Both criteria here are from *type one.* In this way we can formulate (P1) in the case of more than two criteria from *type one.*

The following problems are:

$$\text{lexmin} \{ (c(W), t(W), b(W)) \mid W \in PS(A) \}, \tag{P2}$$

and

$$\text{lexmin}\ \{\ (b(W), c(W), t(W),)\ |\ W \in PS(A)\ \},\qquad \text{(P3)}$$

where we have criteria of mixed type, so that in (P2) the first two are of *type one* and in (P3) the first one is from *type two;*

$$\text{lexmin}\ \{\ (c(W), b(W), t(W))\ |\ W \in PS(A)\ \},\qquad \text{(P4)}$$

where the criterion in the middle is of *type two,* while the others are of *type one.*

These three problems seem to be equivalent, but the difference is in the way of solving them.

Sometimes, the more realistic problem statement is like the following one:

$$\text{lexmin}\ \{\ (c(W), t(W))\ |\ r(W) \le r_0, b(W) \le b_0, W \in PS(A)\ \},\qquad \text{(P5)}$$

which means that among all paths which are "not too bad" according to $r(W)$ and $b(W)$, we are looking for the "best" one according to the lexicographical sense with respect to $c(W)$ and $t(W)$.

Of course there are other problem formulations which are also logically possible but we believe that they can be formulated and solved in a similar way.

P1 can be solved by means of a small modification of Dijkstra's algorithm: wherever there are inequality signs like "<" or ">" they should be replaced by "$\prec$" and "$\succ$", respectively. Apart from this, now we need to label each node of the graph with labels corresponding to each criterion of *type one.* To each node $V_i$ we assign temporary labels $(C_i, T_i, S_i)$. When they become final, we shall denote them by $C_i^*$ and $T_i^*$, which means $(C_i^*, T_i^*)$ is the lexicographically shortest "length" of the path from the start point $V_1$ to $V_i$ and $S_i$ is the number of the last arc of this path.

For the procedure given below, the parameters have the following meanings:

- $A$ – arc set of the graph ;
- $c$ and $t$ – given arc weight vectors (there can be more than two, the order is of importance) ;
- $C, T, S$ – node labels ;
- $STOP$ is a logical variable; $STOP$ is true if $V_n$ is not reachable from $V_1$ .

procedure *PRP1*$(A, c, t, C, T, S, STOP)$;

set $C_1^* := 0;\ T_1^* := 0;\ S_1^* := m + 1;$

$C_i := \infty;\ T_i := \infty;\ S_i := 0$ for all $i = 2, \ldots, n;$

$i := 1;\ BREAK :=$ false; $STOP :=$ false;

repeat (* until $BREAK$ or $STOP$ *)

investigate all arcs $A_k = (i, j)$ with temporarily labelled $V_j$;

if $(C_j, T_j) \succ (C_i^* + c_k, T_i^* + t_k)$ then

begin $C_j := C_i^* + c_k;\ T_j := T_i^* + t_k;\ S_j := k$ end;

find $(C_p, T_p) = \text{lexmin}\{\ (C_q, T_q)\ |\ V_q$ temporarily labelled$\}$;

if $C_p = \infty$ then

begin $V_n$ is not reachable from $V_1$; *STOP* := true end

else begin label $V_p$ by $(C_p^*, T_p^*)$; $i := p$ end;

if $i = n$ then *BREAK* := true

until *BREAK* or *STOP*.

The justification of the algorithm is obvious. For one criterion, for example $c$, this is exactly Dijkstra's algorithm.

Notice that this procedure can run also in the case when there are no criteria like $c$ or $t$ (and of course $C$ and $T$). In this case, the procedure simply finds a feasible path connecting $V_1$ and $V_n$.

EXAMPLE 1. (for the problem P1)

Consider the network shown in Figure 2. The weights on the arcs follow the scheme $c_k / t_k$, that is, the first number is the cost for traversing the arc and the second number is the time consumed in traversing this arc.

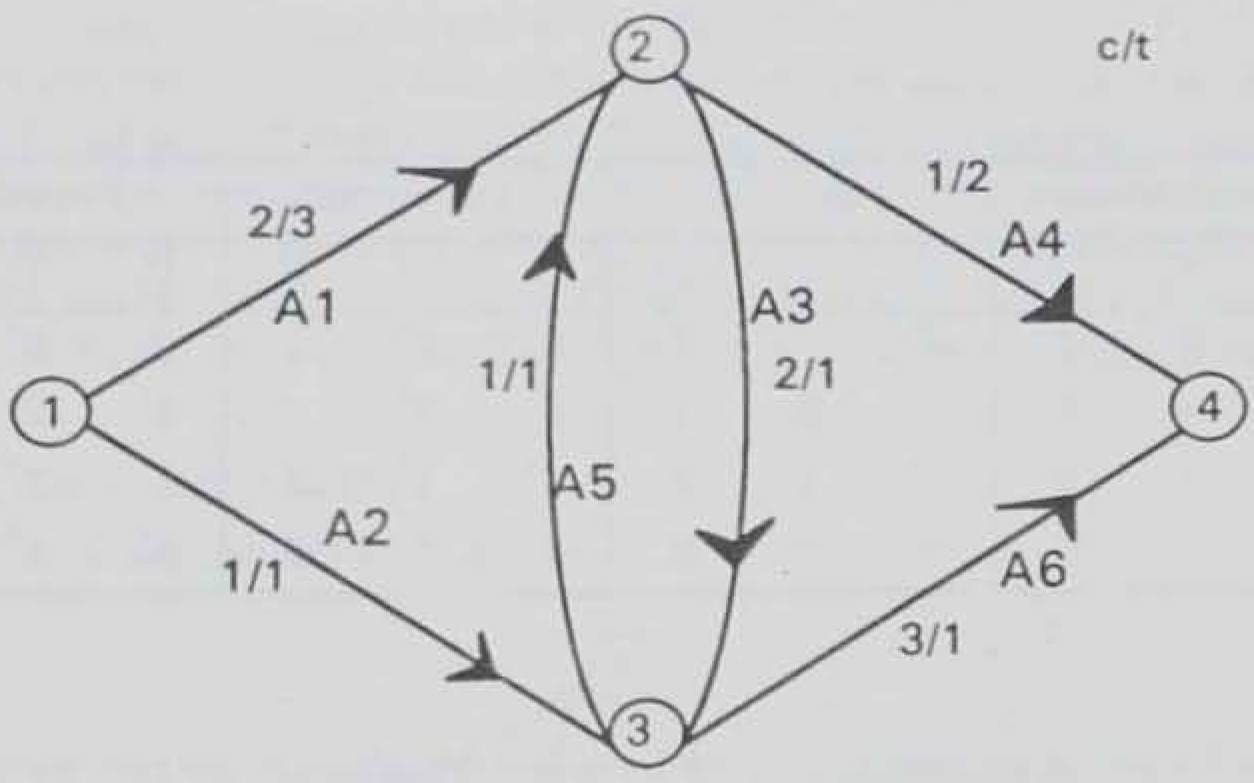



Figure 2.

$i := 1$

$$A_1 = (1,2): \quad (C_2, T_2) = (\infty, \infty) \succ (C_1^*, T_1^*) + (c_1, t_1) = (0,0) + (2,3) = (2,3)$$

$\Rightarrow (C_2, T_2) := (2, 3)$, $S_2 := 1$.

$$A_2 = (1,3): \quad (C_3, T_3) = (\infty, \infty) \succ (C_1^*, T_1^*) + (c_2, t_2) = (0^*, 0^*) + (1,1) = (1,1)$$

$\Rightarrow (C_3, T_3) := (1, 1)$, $S_3 := 2$;

all arcs $(1, j)$ are investigated and

$$(C_3, T_3) = (1,1) = \text{lexmin}\,\{(C_q, T_q) \mid V_q - \text{temporarily labelled}\}$$

$\Rightarrow$ $V_3$ is permanently labelled by $(C_3^*, T_3^*) := (1, 1)$ (end of the first iteration).

$i := 3$

$A_5 = (3, 2): \ C_2, T_2) = (2, 3) \succ (C_3^*, T_3^*) + (c_5, t_5) = (1^*, 1^*) + (1, 1) = (2, 2)$
$\Rightarrow (C_2, T_2) := (2, 2), \ S_2 := 5.$

$A_6 = (3, 4): \ C_4, T_4) = (\infty, \infty) \succ (C_3^*, T_3^*) + (c_6, t_6) = (1^*, 1^*) + (3, 1) = (4, 2)$
$\Rightarrow (C_4, T_4) := (4, 2), \ S_4 := 6;$

all arcs $(3, j)$ are investigated and

$(C_2, T_2) = (2, 2) = \text{lexmin} \{ (C_q, T_q) \mid V_q - \text{temporarily labelled}\}$
$\Rightarrow V_2$ permanently labelled by $(C_2^*, T_2^*) := (2, 2)$ (end of the second iteration).

$i := 2$

$A_4 = (2, 4): \ (C_4, T_4) = (4, 2) \succ (C_2^*, T_2^*) + (c_4, t_4) = (2^*, 2^*) + (1, 2) = (3, 4)$
$\Rightarrow (C_4, T_4) := (3, 4)$ and $S_4 := 4$ (end of the third iteration).

Since all arcs $(2, j)$ are investigated, the only not permanently labelled vertex is $V_4$ and we put $(C_4^*, T_4^*) := (3, 4)$.

The $(C, T)$ – lexicographically best path is $(A_4, A_5, A_2)$, since $S_4 = 4$, $(A_4 = (2, 4))$, $S_2 = 5 \ (A_5 = (3, \ 2))$, $S_3 = 2 \ (A_2 = (1, 3))$, $S_1 = 7 \ (= m + 1)$.

All steps are shown in Table 1.

Table 1.

| | initialization | | | 1 iteration | | | 2 iteration | | | 3 iteration | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | $C_i$ | $T_i$ | $S_i$ | $C_i$ | $T_i$ | $S_i$ | $C_i$ | $T_i$ | $S_i$ | $C_i$ | $T_i$ | $S_i$ |
| 1 | $0^*$ | $0^*$ | 7 | $0^*$ | $0^*$ | 7 | $0^*$ | $0^*$ | 7 | $0^*$ | $0^*$ | 7 |
| 2 | $\infty$ | $\infty$ | 0 | 2 | 3 | 1 | $2^*$ | $2^*$ | 5 | $2^*$ | $2^*$ | 5 |
| 3 | $\infty$ | $\infty$ | 0 | $1^*$ | $1^*$ | 2 | $1^*$ | $1^*$ | 2 | $1^*$ | $1^*$ | 2 |
| 4 | $\infty$ | $\infty$ | 0 | $\infty$ | $\infty$ | 0 | 4 | 2 | 6 | $3^*$ | $4^*$ | 4 |

For the following problems when we have one criterion of *type two*, we need some preparation. Let us partition the arc set $A$ like $A = E_1 + E_2 + ... + E_r$ as follows:

$(A_k, A_l \in E_i \Rightarrow b_k = b_l) \wedge (A_k \in E_i, A_l \in E_{i+1} \Rightarrow b_k < b_l).$

We denote also $E^k := E_1 + ... + E_k$ for $k = 1, 2, ..., r$, so that $E^r = A$.

For P2 we propose the following procedure:

procedure *PRP2*($A$, $c$ – of *type one*, $t$ – of *type one*, $b$ – of *type two*, $C, T, S, STOP$);

construct the sets $E_1, E_2, ... ,E_r$ and $E^1, E^2, ... ,E^r$;

$BREAK$ := false;

$k = r + 1$;

$(c(W), t(W)) := (\infty, \infty)$;

repeat (* until $BREAK$ or $STOP$ *)

$k = k - 1;$

find $W_k$ using procedure $PRP1(E^k, c, t, C, T, S, STOP)$;

if no $STOP$ and $(c(W_k), t(W_k)) \prec (c(W), t(W))$ then

begin

$(c(W), t(W)) := (c(W_k), t(W_k)); W := W_k$

end

else $BREAK$ := true;

if $k = 1$ then $BREAK$ := true

until $BREAK$ or $STOP$.

Notice that we define the relation $\prec$ also in case of identity of both vectors so that at each iteration when $E^k$ decreases, $W$ will be updated even in case $(c(W), t(W)) = (c(W_k), t(W_k))$. On the other hand if there are no criteria like $c$ and $t$ in the problem formulation, the procedure will find a path between $V_1$ and $V_n$ which is most wide in sense of $b$.

At each repeat – iteration the procedure PRP1 solves one problem of type P1 with one, two or more criteria of *type one*. At the same time the width of the found path $W_k$ monotonously increases. If at one iteration $W_{k+1}$ appears to be lexicographically larger than $W_k$ then the procedure stops because $BREAK$ = true with $W = W_k$ as an optimum solution. It can be identified by $S$. If the procedure ends after $STOP$ in the first iteration, there is no path between $V_1$ and $V_n$ and obviously the problem is not solvable. After $STOP$ but not in the first iteration, the solution is defined also by $W_k$ and $S$.

EXAMPLE 2. (for the problem P2)

The network is given in Figure 3.

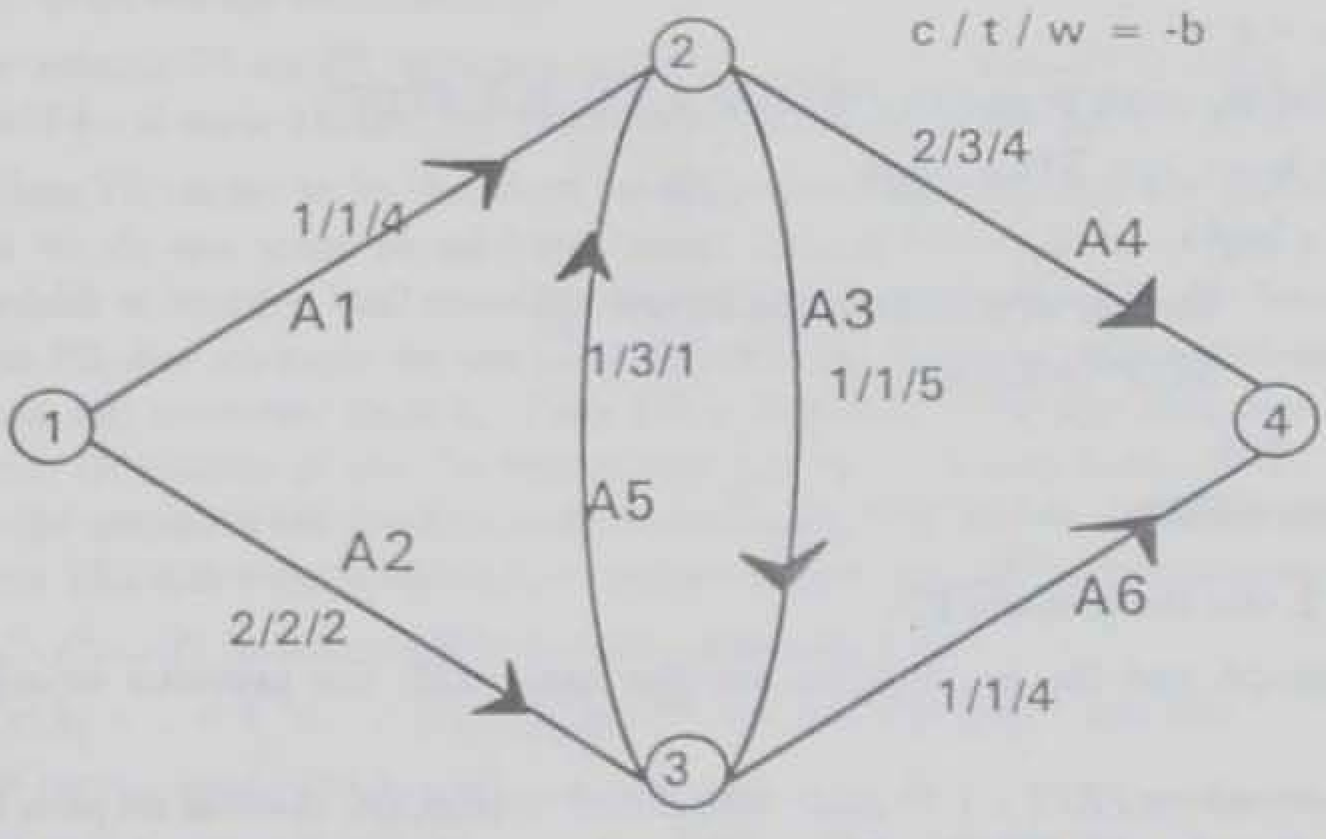



Figure 3.

For the arc subsets we have the following:

$E_1 := \{ A_3 \}$, $\quad E^1 := E_1 = \{ A_3 \}$,

$E_2 := \{ A_1, A_4, A_6 \}$, $\quad E^2 := E_1 + E_2 = \{ A_1, A_3, A_4, A_6 \}$,

$E_3 := \{ A_2 \}$, $\quad E^3 := E_1 + E_2 + E_3 = \{ A_1, A_2, A_3, A_4, A_6 \}$,

$E_4 := \{ A_5 \}$, $\quad E^4 := A = \{ A_1, \dots A_6 \}$.

Further:

First iteration (P1 on $E^4 = A$):

Two solutions : $W' = (A_1, A_3, A_6)$ and $W'' = (A_2, A_4)$ with $(C(W'), T(W')) = (3, 3)$.

Second iteration (P1 on $E^3$):

The same two solutions $W'$ and $W''$.

Third iteration (P1 on $E^2$):

The only solution is $W'$.

Fourth iteration (P1 on $E^1$):

There is no solution in this arc subset since there is no path connecting $V_1$ and $V_4$.

Since the procedure stops after *STOP*, the desired *optimum* solution is $W' = (A_1, A_3, A_6)$, obtained from the previous iteration. It costs 3 units, can be traversed in 3 time units and is 4 units wide.

Problem P3 can be solved as follows:

procedure *PRP3(A, b – type two, c, t –* of *type one, C, T, STOP)*;

```
    construct the sets E_1, E_2, ... ,E_r and E^1, E^2, ... ,E^r;
    k := 0;
    repeat
        k := k + 1;
        find W_k using procedure PRP1(E^k, c, t, C, T, S, STOP);
        if k = r and STOP then
            begin
                there is no path between V_1 and V_n;
                STOP := false
            end
    until no STOP.
```

**Example 3.** (for the problem P3)

The graph and the arc weights are the same with the previous example (see Figure 3).

The procedure *PRP1* for $E^1$ ends with $STOP$ = true, i.e. there is no path between $V_1$ and $V_4$. It ends with $STOP$ = false on $E^2$ and the path found is $(A_1, A_4)$ which is the desired solution: it is 4 units wide, costs 3 units and can be traversed in 4 time units.

Problem P4 can be solved with the following procedure:

Procedure *PRP4(A, c – type I, b – type II, t – type I, C, T, S, STOP)*;

*STOP* := false;

solve P2 with two criteria *c* and *b* using

procedure *PRP2(A, c – type I, b – type II, C, S, STOP)*;

if there is no solution then *STOP* := true

else

begin

let the solution be obtained on $E^k$;

solve P1 with two criteria *c* and *b* using

procedure *PRP1*($E^k$, *c*, *t* – of *type I, C, T, S, STOP*)

and let the solution be *W*

end.

The path *W* is an optimum solution of P4. If we solve P2 with two criteria *c* and *b* from mixed type and if the solution is obtained on $E^k$, then we are sure that the shortest path according to the first criterion on $E^{k-1}, E^{k-2}, \dots, E^1$, is longer than *W*. On the other hand, if we solve P1 with two criteria *c* and *t* on $E^k$, then we are sure that the optimum path will contain at least one arc from $E^k$, which defines the width of *W*.

EXAMPLE 4. (for the problem P4)

We use the same network as in example 2, (Figure 3), but the weights are ordered as $c_k / w_k / t_k$.

By solving P2 we obtain the arc sequence $(A_1, A_4)$ as a *lexmin–path* according to *c* and *w* and as an arc set we come to $E^2$.

After solving P1 on $E^2$, with two criteria *c* and *t*, we obtain the same path as a solution of P4 – it costs 3 units, is 4 units wide and can be traversed in 4 time units.

Problem P5 seems to be the most realistic optimum path problem since most of the times we do not want to minimize some criteria but only to obtain a feasible solution which is "not too bad" according to those criteria as shown in the formulation of problem P5. For example we are searching for the shortest path which is reliable enough and not narrower than $b_0$. Then P5 is formulated for one criterion and two restrictions. Of course it can be formulated for an arbitrary number of objective functions (no matter what the type and the order is), and also an arbitrary number of restrictions. The following problem formulation generalizes all the previous ones:

Let $f^1, f^2, \dots f^p$ be criteria of both types in mixed order;

for $1 \le k_1 < k_2 < \dots < k_q \le p$ $(0 \le q \le p)$, let $f^{k_l}$ are criteria of *type two* like

$$f^{k_l}(W) := \max\{ f^{k_l} \mid A_j \in W \}, l = 1, 2, \dots, q$$

and the others of *type one* like

$$f_k(W) := \sum \{ f^{k_j} \mid A_j \in W \}.$$

In the special case all criteria can be of *type one* ($q = 0$) or of *type two* ($q = p$).

Let $g^1$ and $g^2$ be functions of *type one*, and *type two*, respectively. We assume that all criteria of *type one* are dealing with non-negative arc weights.

The most general problem formulation in our approach, is the following:

$$\text{lexmin}\,\{\,(f^1(W), \dots f^p(W))\,|\,g^1(W) \le G^1, g^2(W) \le G^2, W \in PS(A)\,\}. \qquad \text{(PG)}$$

If we put $G^1 = G^2 = \infty$, we will obtain a problem without any restrictions of inequality type like the problems (P1) - (P4). If in addition we take $p$, $k_1, \dots k_q$ in an appropriate way, we will obtain all the previous formulated problems.

In the procedure given below for solving PG we iteratively solve P2 (and hence also P1). In order for the procedure to be simpler, we prefer the first criterion to be of *type one* and the last one of *type two*. If the first one is of *type two*, we put an artificial criterion $f^o$ of *type one* with $f^o_k := 0$ for $k = 1, \dots, m$. If the last criterion is of *type one*, we put an artificial criterion $f^{p+1}$ of *type two* with $f_k^{p+1} := 1$ for $k = 1, \dots, m$, so that $f^{p+1} = f^{k_q+1}$.

procedure *PRPG*$(A, f^1, \dots f^p$ – of given type, $F^1, \dots, F^p, S, g^1$ of *type I*,
 $g^2$ *of type II*, $G^1, G^2, STOP)$;

$STOP :=$ false;

for each vertex $V_i$ calculate the shortest $g^1$ – distance $\rho_i$ from $V_i$ to $V_n$ using Dijkstra's algorithm;

if $\rho_1$ cannot be defined then there is no path between $V_1$ and $V_n$ and PG has no solution and $STOP :=$ true

else

begin

define $f_k^o := 0$ and $f_k^{p+1} := 1$ for all $k = 1, \dots, m$;

for $l := 1$ to $q + 1$ do

begin

find $W_{k_l}$ as a solution of P2 using

procedure *PRP2*$(A, f^o, f^1, \dots, f^{k_l}, F^o, F^1, \dots, F^{k_l-1}, S, STOP)$

(* When using procedure P1, the only change will be:

investigate all arcs $A_k = (i,j)$ with temporarily labelled $V_j$ and $g_k^2 \le G^2$ and $g^1$ – distance from $V_1$ to $V_i + g_k^1 + \rho_j \le G^1$ *);

define $f^{k_l}$ as of *type one* by $f^{k_l}_j := 0$ for all $j = 1, \dots, m$; $F^{k_l} := 0$;

drop all arcs $A_s$ from $A$ for which $f_s^{k_l} > f^{k_l}(W_{k_l})$

end

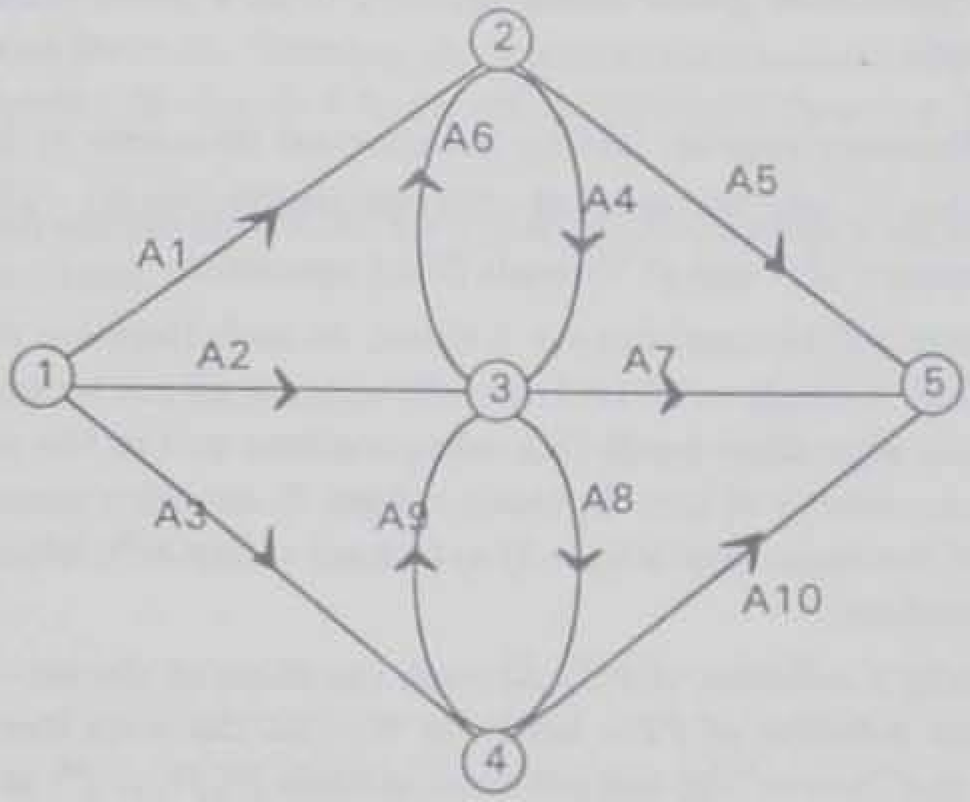
end. (* The optimum solution is $W_{k_{q+1}}$ *)

EXAMPLE 5. (for the problem PG)

The graph and all arc weights are given in Figure 4. The problem is

$$\operatorname{lexmin} \{ (f^1(W), f^2(W), f^3(W), f^4(W)) \mid g^1(W) \le 10, g^2(W) \le -2, W \in PS(A) \},$$

where $f^2$, $f^4$ and $g^2$ are of *type two;* the others are of *type one.*




$f^1 / f^2 / f^3 / f^4 // g^1 / g^2$

| | | | |
|---|---|---|---|
| $A_1$ : | 1 / −5 / 1 / −4 // 1 / −3 | $A_6$ : | 1 / −2 / 2 / −2 // 3 / −2 |
| $A_2$ : | 2 / −4 / 3 / −2 // 4 / −2 | $A_7$ : | 1 / −3 / 1 / −4 // 3 / −3 |
| $A_3$ : | 1/ −1 / 1 / −2 // 3 / − | $A_8$ : | 2 / −2 / 2 / −2 // 4 / −3 |
| $A_4$ : | 1 / −4 / 2 / −5 // 2 / −2 | $A_9$ : | 1 / −2 / 1 / −4 // 3 / −1 |
| $A_5$ : | 3 / −3 / 1 / −5 // 1 / −4 | $A_{10}$: | 1 / −3/ 3 / −1 // 5 /−2 |

Figure 4.

In this example we have $G^1 = 10$, $G^2 = -2$, $p = 4$, $q = 2$, $k_1 = 2$ and $k_2 = 4$.

Since the first criterion is of *type one* and the last criterion is of *type two,* we do not need to define any artificial criteria.

For the first restriction of *type one* $g^1(W) \le G^1$ we have :

$$\rho_5 = 0,\ \rho_4 = 5,\ \rho_3 = 3,\ \rho_2 = 1,\ \rho_1 = 6 \le G^1 = 10.$$

The action of procedure $PRP2(A, f^1$ of *type I*, $f^2$ of *type II*, $F^1, S, STOP)$ gives as an optimal path $W' = (A_1, A_4, A_7)$ and $W'' = (A_2, A_7)$, with $f^1(W') = 3$ and $f^2(W') = -3$. We drop the arcs $A_3$, $A_6$, $A_8$ and $A_9$ from A since their arc weights $f^2$ are larger than $f^2(W') = -3$.

Now after executing procedure $PRP2(A, f^1, 0, f^3$ of *type I*, $f^4$ of *type II*, $F^1, 0, F^3, S, STOP)$ we obtain the only solution $W = (A_1, A_4, A_7)$ with $f^1(W) = 3$, $f^3(W) = 4$,

$f^4(W) = -4$, which is the solution of the problem with optimal values $f^1(W) = 3$, $f^2(W) = -3$, $f^3(W) = 4$, $f^4(W) = -4$ and feasible restriction values $g^1(W) = 6 < 10$ and $g^2(W) = -2 \leq -2$.

THEOREM. The procedure given above solves PG in a finite number of steps with complexity $O(n\, m^2)$.

PROOF. The preliminary step for finding the shortest distances $\bar{n}_i$ from each vertex $V_i$ to the target vertex $V_n$ can be executed with complexity $O(n\, m)$ – Dijkstra's algorithm.

The initialization of $f^0$ and $f^{p+1}$ needs $O(m)$ operations.

The main loop will be executed $q + 1$ times. At each iteration one problem of type P2 has to be solved. It needs $m\, O(n\, m)$ operations since all arc weights are nonnegative – the Dijkstra–like algorithm needs $O(n\, m)$ operations and in the worst case $A$ can be partitioned into $m$ subsets of type $E_i$ (each subset $E_i$ contains exactly one arc). Hence the complexity of the main loop is $(q + 1)\, m\, O(n\, m) = O(n\, m^2)$, which is the complexity of the whole procedure.

Is $W_{k_{q+1}}$ really a solution of PG? At each iteration of the for–loop the procedure finds an optimum solution of P2 – let it be $W_{k_l}$. At the next iteration $W_{k_{l+1}}$ will be neither "better" nor "worse" for the previous criteria $f^0, f^1, \dots f^{k_l}$ since they are in the same order in the problem formulation of P2 for the next iteration and all "worse" arcs like $A_s$ with $f_s^{k_l} > f^{k_l}(W_{k_l})$ have been dropped from the arc set $A$. Hence $W_{k_{p+1}}$ minimizes lexicographically all criteria. The procedure ends after $STOP$ = true only if there is no path from $V_1$ to $V_n$. This will be the case only if $\bar{n}_1$ cannot be defined.

## 4. FINAL COMMENTS

Numerical investigation will be done in the future. The procedure seems to be running fast due to its simplicity and good complexity. It can run after appropriate changes also if some of the arc weights for the criteria of *type one* are negative. In this case PG will be NP–complete since this is also the case for one criterion only.

ACKNOWLEDGMENT. This paper has been prepared during the first author's visit at the University of Macedonia, Thessaloniki, Greece, according to the NATO Scientific Fellowship Programme (1994). We thank the staff of the Department of Applied Informatics for the very good conditions of work. We wish to thank Professors Tsouros and Paparrizos for their support.

## REFERENCES

[1] Bovy,P.H.L., and Stern,E, *Route choice: Wayfinding in transport networks*, Kluwer, 1990.

[2] Current,J., "Multiple objectives in transportation network design and routing (Editorial)", *European Journal of Operational Research* 65 (1993) 1–3.

[3] Current,J., and Marsh,M., "Multiobjective transportation network design and routing problems: Taxonomy and annotation" *European Journal of Operational Research* 65 (1993) 4–19.

[4] Ivanchev,D., and Naegler,G., *Network optimization*, Sofia, TU of Sofia, 1993.

[5] Ivanchev,D., and Ruhe,G., "Finding the k bicriterial foremost edges in planar flow networks", *Comptes rendus de l'Academie bulgarie des Sciences Sofia* 39/9 (1986) 35–38.

[6] Ruhe,G., "A polynomial algorithm for a bicriterial shortest path problem", Private communication, 1986.