

APPENDIX A

PSEUDO-CODE OF ALGORITHM 2

Algorithm 2. CTKP($i, k, pred$);

```

{comment:  $i = center, k = root, pred = p_k^i$ }
begin
if ( $k \neq 0$ ) then
 $d\_path := \sum_{i \in P[i, pred]} d_i;$ 
else
 $d\_path := 0;$ 
end if
if ( $i \neq k$ ) then
Forward_Move( $i, pred, k$ );
end if
 $j := k + 1;$ 
while ( $j \leq LAST(k)$ ) do
begin
if ( $j \neq pred$ ) then
Forward_Move( $i, p_j, j$ );
if ( $j = LAST(j)$ ) then {comment: node  $j$  is a leaf node}
 $w := j;$ 
do
Backward_Move( $i, w, p_w$ );
 $w := p_w;$ 
while ( $LAST(w) = j$  and  $w \neq k$ )
{comment:  $w$  has no successor  $t$  such that  $t > j$  and  $w \neq k$ }
end if
 $j := j + 1;$ 
else
 $j := LAST(pred) + 1;$ 
end if
end
end
end

```

Procedure 2.1 **Forward_Move**(i, j, k);

```

begin
   $d\_path := d\_path + d_k$ ;
  for  $h := \underline{d}$  up to  $d_k - 1$  do     $P(i, k, h) := -\infty$ ;
  for  $h := d_k$  up to  $H$  do
    begin
      if ( $d\_path \leq h$ ) then
         $P(i, k, h) := P(i, j, h - d_k) + \bar{c}_{ik}$ ;
        if ( $p_j = k$ ) then
           $I(i, k, h) := 1$ ;
        end if
      else
         $P(i, k, h) := -\infty$ ;
      end if
    end
  end
end

```

Procedure 2.2 **Backward_Move**(i, j, k);

```

begin
   $d\_path := d\_path - d_j$ ;
  for  $h := \underline{d}$  up to  $H$  do
    begin
      if ( $P(i, k, h) \geq P(i, j, h)$ ) then
         $I(i, j, h) := 0$ ;
      else
         $P(i, k, h) := P(i, j, h)$ ;
         $I(i, j, h) := 1$ ;
      end if
    end
  end
end

```

Procedure 1.1' **Find- γ_k^*** ;

```

begin
  CTKP( $k, k, -1$ );    {comment: as  $p_k^k$  is undefined, we set  $p_k^k = -1$ }
  if ( $k \neq 0$ ) then
     $pred := k + 1$ ;    {comment:  $pred = p_k^i$ }
  end if
end

```

```
for  $i := k + 1$  up to  $LAST(k)$  do  
begin  
if  $(i > LAST(pred))$  then  
   $pred := i$ ;  
end if  
CTKP $(i, k, pred)$ ;  
end
```

$$\gamma_k^* := \max_{i \in T(k)} \max_{1 \leq t \leq m} \{P(i, k, h^t) - \hat{F}_i^t\};$$

```
end if  
end
```

Applied Optimization

1. D.-Z. Du and D.F. Hsu (eds.): *Combinatorial Network Theory*. 1996
ISBN 0-7923-3777-8
2. M.J. Panik: *Linear Programming: Mathematics, Theory and Algorithms*. 1996
ISBN 0-7923-3782-4
3. R.B. Kearfott and V. Kreinovich (eds.): *Applications of Interval Computations*.
1996 ISBN 0-7923-3847-2
4. N. Hritonenko and Y. Yatsenko: *Modeling and Optimization of the Lifetime of
Technology*. 1996 ISBN 0-7923-4014-0
5. T. Terlaky (ed.): *Interior Point Methods of Mathematical Programming*. 1996
ISBN 0-7923-4201-1
6. B. Jansen: *Interior Point Techniques in Optimization. Complementarity, Sensitivity
and Algorithms*. 1997 ISBN 0-7923-4430-8
7. A. Migdalas, P.M. Pardalos and S. Storøy (eds.): *Parallel Computing in Optimization*.
1997 ISBN 0-7923-4583-5
8. F.A. Lootsma: *Fuzzy Logic for Planning and Decision Making*. 1997
ISBN 0-7923-4681-5
9. J.A. dos Santos Gromicho: *Quasiconvex Optimization and Location Theory*. 1998
ISBN 0-7923-4694-7
10. V. Kreinovich, A. Lakeyev, J. Rohn and P. Kahl: *Computational Complexity and
Feasibility of Data Processing and Interval Computations*. 1998
ISBN 0-7923-4865-6
11. J. Gil-Aluja: *The Interactive Management of Human Resources in Uncertainty*. 1998
ISBN 0-7923-4886-9
12. C. Zopounidis and A.I. Dimitras: *Multicriteria Decision Aid Methods for the Predic-
tion of Business Failure*. 1998 ISBN 0-7923-4900-8
13. F. Giannessi, S. Komlósi and T. Rapcsák (eds.): *New Trends in Mathematical Pro-
gramming. Homage to Steven Vajda*. 1998 ISBN 0-7923-5036-7
14. Ya-xiang Yuan (ed.): *Advances in Nonlinear Programming*. Proceedings of the '96
International Conference on Nonlinear Programming. 1998 ISBN 0-7923-5053-7
15. W.W. Hager and P.M. Pardalos: *Optimal Control. Theory, Algorithms, and Applica-
tions*. 1998 ISBN 0-7923-5067-7
16. Gang Yu (ed.): *Industrial Applications of Combinatorial Optimization*. 1998
ISBN 0-7923-5073-1
17. D. Braha and O. Maimon (eds.): *A Mathematical Theory of Design: Foundations,
Algorithms and Applications*. 1998 ISBN 0-7923-5079-0