Iranian Journal of Science & Technology, Transaction A, Vol. 30, No. A3
Printed in The Islamic Republic of Iran, 2006

NETWORK OPTIMIZATION WITH PIECEWISE LINEAR CONVEX COSTS*

S. KETABI

Department of Management, University of Isfahan, I. R. of Iran
Email: sketabi@yahoo.com

Abstract – The problem of finding the minimum cost multi-commodity flow in an undirected and complete network is studied when the link costs are piecewise linear and convex. The arc-path model and overflow model are presented to formulate the problem. The results suggest that the new overflow model outperforms the classical arc-path model for this problem. The classical revised simplex, Frank and Wolf and a heuristic method are compared for the problem.

Keywords – Multi-commodity network flow problem, piecewise linear and convex costs, Arc-path Model, Overflow Model, Frank and Wolfe Method

1. INTRODUCTION

Convex costs, which are a more general cost structure than linear models, occur in numerous settings including costs of power losses in electrical networks due to resistance, congestion costs in city transportation networks and delay cost of communication networks. It is well known that the network minimization problem with piecewise linear convex costs can be restated as a network problem with linear costs [1]. The minimum cost multicommodity flow problem in a network with piecewise linear and convex costs can also be converted to a related problem in a network with linear costs in the same way.

Two models, the *overflow model* and the traditional arc–path model are compared to formulate the multicommodity network flow with piecewise linear and convex costs. This paper is organized as follows: first, some concepts and notations are defined. Next, two models for the problem are presented and their equivalence is discussed. Then, a greedy heuristic method is proposed to obtain a good solution and finally, the computational results are used to compare the models and the proposed method with the existing methods.

2. DEFINITIONS AND NOTATIONS

Consider a complete graph (directed or undirected) with the node set N and the link set E. The overflow model [2] is valid for both directed and undirected networks. The only difference is in distinguishing between ordered or unordered pairs of nodes. It is assumed that a nonnegative traffic matrix $T=(t_{ij}, (i, j)\in E)$ and a nonnegative capacity matrix $Q=(q_{ij}, (i, j)\in E)$ are given. So in the undirected case these matrices are symmetric.

The overflow variables, $\lambda_{(ij)k}$, are double indexed, one index representing a link and the other representing a node. These variables indicate how much flow on the link (i, j) overflows via a third node k, which is immediately subsequent to i; that is, the flow from i to j crossing edges (i, k) and (k, j). Obviously $\lambda_{(ij)k}\geq 0$ must hold. Also the flow on link (i,j) is equal to

$$x_{ij} = t_{ij} + \sum_{k\neq i,j}(\lambda_{(ik)j}+ \lambda_{(kj)i}-\lambda_{(ijk)}).$$

*Received by the editor May 25, 2004 and in final revised form February 4, 2007

This flow includes the offered traffic for the corresponding origin–destination (OD) pair and also the negative or positive effects of all routing decisions. It is necessary to bound the above expression not to be greater than the capacity, q_{ij} ;

$$0 \le t_{ij} - \sum_k \lambda_{(ij)k} + \sum_k \lambda_{(ik)j} + \sum_k \lambda_{(kj)i} \le q_{ij}$$

If there is a lower bound ℓ_{ij} on the flow, then for a feasible flow:

$$\sum_{k \neq i,j} (\lambda_{(ij)k} - \lambda_{(ik)j} + \lambda_{(kj)i}) \le t_{ij} - \ell_{ij}, \forall (i, j) \in E$$

The overflow directed to any link may be overloaded again to other links, therefore the model is as general as the arc–path model. It is preferred to formulate the problem using these variables as it is not necessary to define an index indicating commodities for the variables. For a feasible set of overflow variables, there would be a corresponding multicommodity flow, not necessarily uniquely determined. One can easily calculate the flow on different paths for each commodity using the values of the overflow variables $\lambda_{(ij)k}$ [2], [3]. The other advantage of the model over the arc–path model is the polynomial number of the variables. More details are given in the next section.

The network flow is completely described by the overflow model. This approach is particularly useful for undirected networks, but can also be employed for directed networks. Although the overflow variables are basically defined on complete graphs, the overflow model may also be extended to incomplete graphs considering high costs for the links which do not exist.

3. PROBLEM FORMULATION

a) The Arc–Path Model

Consider an undirected network consisting of nodes *N* and *K* OD pairs (commodities are distinguished by their origin-destination nodes). The cost of flow on each link is assumed to be a piecewise linear convex function with *R* segments. A simple convex piecewise linear function is shown by Fig. 1. It is not necessary that all links' cost functions have the same number of break points, however this assumption may be made without loss of generality. If *(i,j), r* and κ represent a link, range and commodity respectively, the problem can be written as

$$\text{Min} \qquad \sum_{(i,j)\in E} \sum_{r=1}^{R} c_{ij}^r f_{ij}^r \tag{1}$$

$$\text{s.t.} \qquad \sum_r f_{ij}^r - \sum_{\mu \in P} \delta_{ij}^{\mu} x_{\mu} \ge 0 \qquad \forall (i,j) \in E \tag{2}$$

$$\sum_{\mu \in P_k} x_{\mu} = t_k \qquad \forall \kappa = 1,\ldots,K \tag{3}$$

$$f_{ij}^r \le b_{ij}^r - b_{ij}^{r-1} \quad \forall (i,j) \in E, r = 1,\ldots,R \tag{4}$$

$$f_{ij}^r \le 0, x_{\mu} \ge 0 \qquad \forall (i,j) \in E, r = 1,\ldots,R, \kappa \in N, \mu \in P, \tag{5}$$

where

μ: a path

P_κ: the set of paths for commodity κ,

P: the set of all paths $P = \cup_\kappa P_\kappa$,

f_{ij}^r: the flow on link (i,j) in range r,

x_μ: the flow of commodity κ on path $\mu \in P_k$,

C_{ij}^r: the slope of the rth line segment,

δ_{ij}^{μ} : the arc – path incidence element (which is equal to 1 if (i, j) ∈ μ and is equal to 0 otherwise),

t_{κ} : the offered traffic of commodity κ ($t_{ij} \equiv t_k$, i, j are origin-destination nodes for commodity κ),

b_{ij}^{r} : the upper limit of the rth line segment.

It is assumed that $b^0_{ij} = 0$ and that there is no fixed cost. In (2) inequalities are used instead of equalities; this does not affect the solution because the minimization problem guarantees equality, but it is a requirement for applying the revised simplex method. Because of the convexity of the costs, $c_{ij}^1 \le c_{ij}^2 \le \ldots \le c_{ij}^R$, automatically means that the flow in one range cannot be positive, unless the flow on the previous range has reached its upper limit. This is expressed by the following conditions:

$$f_{ij}^{r} > 0 \Rightarrow f_{ij}^{l} = b_{ij}^{l} - b_{ij}^{l-1} \qquad \forall l = 1,2,\ldots,r-1$$

$$f_{ij}^{r} < b_{ij}^{l} - b_{ij}^{l-1} \Rightarrow f_{ij}^{l} = 0 \qquad \forall l = r+1,\ldots,R$$

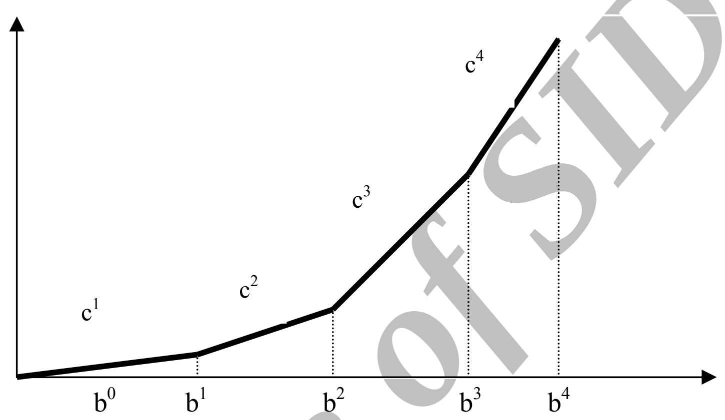

Fig. 1. A simple convex piecewise linear function

b) The Overflow Model

Consider the same assumptions and notations as in the previous section, and let r and i,j represent a range and nodes. Now, the overflow model for the network routing problem with piecewise linear convex cost is:

$$\text{(P)} \quad \text{Min} \ \sum_{(i,j)\in E} \sum_{r=1}^{R} c_{ij}^{r} f_{ij}^{r} \tag{6}$$

$$\text{s.t.} \quad \sum_{r} f_{ij}^{r} + \sum_{k} \lambda_{(ij)k} - \sum_{k} \lambda_{(ik)j} - \sum_{k} \lambda_{(kj)i} = t_{ij} \qquad , \forall (i,j) \in E \qquad (\pi_{ij}) \tag{7}$$

$$f_{ij}^{r} \le b_{ij}^{r} - b_{ij}^{r-1} \ , \forall (i,j) \in E, r = 1,\ldots,R \quad (w_{ij}^{r}) \tag{8}$$

$$f_{ij}^{r} \ge 0, \qquad \lambda_{(ij)k} \ge 0 \qquad , \forall (i,j) \in E, r = 1,\ldots,R, k \in N \tag{9}$$

where the variables in the right parentheses are the corresponding dual variables. The dual of (P) is given by

$$\text{(D) max} \quad \sum_{(i,j)} t_{ij}\pi_{ij} + \sum_{(i,j)} \sum_{r} w_{ij}^{r} (b_{ij}^{r} - b_{ij}^{r-1}) \tag{10}$$

$$\text{s.t.} \quad w_{ij}^{r} \le c_{ij}^{r} - \pi_{ij} \qquad , \forall (i,j), r \tag{11}$$

$$\pi_{ij} \le \pi_{ik} + \pi_{kj} \ , \forall (i,j), k \tag{12}$$

$$w_{ij}^{r} \le 0 \qquad , \forall (i,j), r \tag{13}$$

Constraint (12) expresses the triangle inequality between dual variables π_{ij}, and corresponds to the overflow variables. This constraint binds if the corresponding overflow is positive, otherwise it can hold as inequality; this agrees with the natural idea of overflowing the flow. The complementary slackness conditions are:

$$\text{if } \pi_{ij} < c_{ij}\text{, then } w_{ij}^{r}=0 < c_{ij} - \pi_{ij}\text{, and so } f_{ij}^{r}=0;$$

$$\text{if } \pi_{ij} > c_{ij}\text{, then } w_{ij}^{r}= c_{ij} - \pi_{ij} < 0\text{, and so } f_{ij}^{r}= b_{ij}^{r} - b_{ij}^{r-1};$$

$$\text{if } \pi_{ij} < \pi_{ik} + \pi_{kj}\text{, then } \lambda_{(ij)k}=0,$$

which are equivalent to

$$b_{ij}^{r-1} < F_{ij} < b_{ij}^{r} \Rightarrow \pi_{ij} = c_{ij}^{r}, \tag{14}$$

$$c_{ij}^{r} < \pi_{ij} < c_{ij}^{r+1} \Rightarrow F_{ij} = b_{ij}^{r}, \tag{15}$$

$$\lambda_{(ij)k} > 0 \Rightarrow \pi_{ij} = \pi_{ik} + \pi_{kj}, \tag{16}$$

where $F_{ij} = \sum_r f_{ij}^{r}$.
As can be seen, the problem (6)–(9) is a linear program, which may be solved efficiently by the revised simplex method along with the complementary slackness conditions (14)–(16).

4. EQUIVALENCE OF THE PATH MODEL AND THE OVERFLOW MODEL

In this section, the correspondence between the feasible paths set and the feasible overflows set is verified. A simple proof is presented to show the correctness of the overflow model (6)–(9), as well as to give a deeper insight about the new model.

For every given feasible solution of each model, a corresponding feasible solution of the other model is derived.

- First, if a solution to the path model is given, then the feasible overflow variables are set as follows:

1. All overflow variables are initialized at zero,
2. For every OD pair *i-j* or commodity κ and every path $\mu \in P_\kappa$ with links as i-i_1-i_2-…-i_{l-2}-i_{l-1}-j and positive flow x, the values of overflow variables are changed as below:

$$\begin{aligned}
\lambda_{(ij)i_1} &\leftarrow \lambda_{(ij)i_1} + x_\mu,\\
\lambda_{(i_1 j)i_2} &\leftarrow \lambda_{(i_1 j)i_2} + x_\mu,\\
\lambda_{(i_2 j)i_3} &\leftarrow \lambda_{(i_2 j)i_3} + x_\mu,\\
&\cdots\\
\lambda_{(i_{l-2} j)i_{l-1}} &\leftarrow \lambda_{(i_{l-2} j)i_{l-1}} + x_\mu,
\end{aligned}$$

- Second, if a feasible solution to the overflow model is given, then the feasible path variables are obtained as follows:

3. The set of paths with positive flows, *P*, initially consists of only direct links between OD pairs for routing commodities.
4. For all links *(i, j)* which are contained in at least one path like $\mu_{ij} \in P$, the corresponding overflow decisions are considered, *i.e.* $\forall i_1$: $\lambda_{(ij)i_1} > 0$.
5. The traffic on path μ_{ij} is reduced by $\alpha = \min(\lambda_{(ij)i_1}, x_{\mu_{ij}})$.
6. If $\lambda_{(ij)i_1}$ reroutes all the traffic on *(i, j)*, then path μ_{ij} is removed from *P*; otherwise $x_{\mu_{ij}} \leftarrow x_{\mu_{ij}} - \lambda_{(ij)i_1}$.

7. If path $\mu_{ij} - \{(i\text{-}j)\} \cup \{(i\text{-}i_1),(i_1\text{-}j)\}$ is not in P, it is inserted. Then, the flow on this path will be increased by α.
8. The above procedure will be repeated until all the overflow decisions are assigned to the paths.

The following example will clarify the second part of the above discussion.

Example 1. Consider a network of 5 nodes with the offered traffic as:

$$t_{13} = 5,\ t_{15} = 20,\ t_{25} = 5,\ t_{34} = 10.$$

Assume that feasible overflow variables are given as:

$$\lambda_{(15)2} = 10,\ \lambda_{(25)3} = 5,\ \lambda_{(35)4} = 5.$$

Then, starting from the set $P = \{(1\text{-}5)_{20}, (1\text{-}3)_5, (2\text{-}5)_5, (3\text{-}4)_{10}\}$, in which the subscripts next to the paths show the related flows, using the above procedure, the corresponding paths will be $P = \{(1\text{-}5)_{10}, (1\text{-}2\text{-}5)_{10}, (1\text{-}3)_5, (2\text{-}5)_5, (3\text{-}4)_{10}\}$ at iteration 1, $P = \{(1\text{-}5)_{10}, (1\text{-}2\text{-}5)_5, (1\text{-}2\text{-}3\text{-}5)_5, (1\text{-}3)_5, (2\text{-}5)_5, (3\text{-}4)_{10}\}$ at iteration 2, and finally $P = \{(1\text{-}5)_{10}, (1\text{-}2\text{-}5)_5, (1\text{-}2\text{-}3\text{-}4\text{-}5)_5, (1\text{-}3)_5, (2\text{-}5)_5, (3\text{-}4)_{10}\}$. Figure 2 shows the three successive graphs related to the construction of the paths.

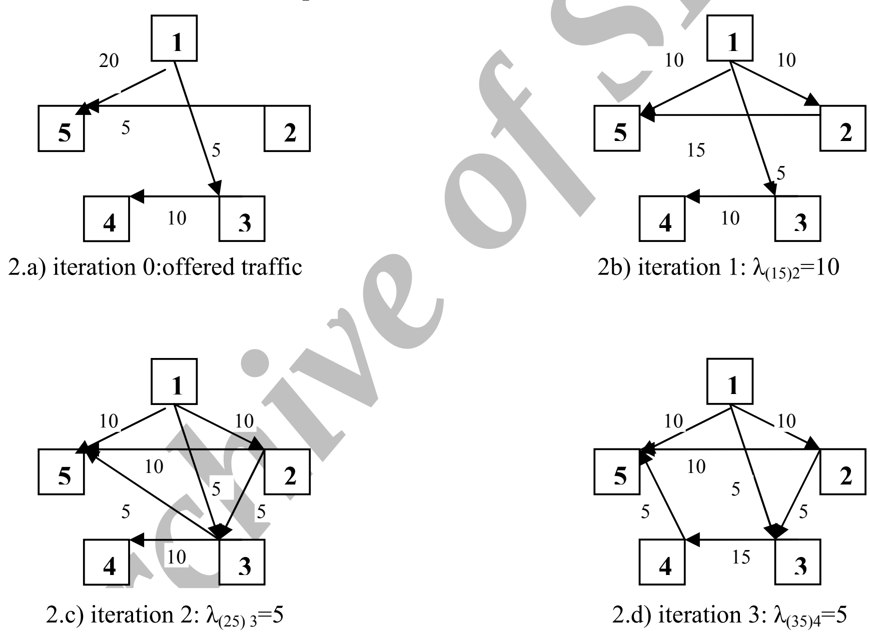

Fig. 2. Successive graphs related to the construction of the paths from overflow variables in example 1

Therefore it is verified that the overflow model correctly formulates the problem, as it is equivalent to the classical path model. In the next sections it will be shown that the overflow model is computationally more efficient for the problem.

5. A HEURISTIC METHOD

Approximate solution procedures are the most common way to solve the multicommodity network flow problems. Indeed the LP formulation of the problem is highly degenerate. Therefore for very large networks, LP methods may be time consuming. In this section an approximate solution will be obtained for the problem. The heuristic greedy-based method presented here considers the properties of a piecewise linear convex cost function in the same way that an equivalent flow problem (with parallel links instead of each link, considering the costs equal to the slopes of the cost function's line segments) does.

At each iteration of the method, a necessary condition is considered for optimality. If the cost of rerouting the flow on a link onto an alternative path is negative, then a better solution will be achieved by rerouting some or all of the flow on that link onto the alternative path. It is noted that the method does not necessarily reroute the whole flow on each link along the desired path.

To explain the method more precisely, suppose that all commodities are routed on the direct links. Then, all overflow variables are checked for the necessary condition of optimality. To do this, two range–indices for each link *(i, j)* are defined: these are inc_{ij} and dec_{ij} , representing incremental and decremental ranges. For every amount f_{ij} of the flow on link *(i, j)*, dec_{ij} is the range that contains f_{ij} and inc_{ij} is the range that contains $f_{ij} + \delta$, for sufficiently small δ. If the flow on the link is equal to one of the breakpoints, these two ranges represent the adjacent ranges to that breakpoint. Otherwise, they are both equal to the range that contains the total flow. Then an overflow variable $\lambda_{(ij)k}$ is chosen to increase when the cost of rerouting the flow is negative, that is $c_{ijk} < 0$, where

$$c_{ijk} = -c_{ij}^{dec_{ij}} + c_{ik}^{inc_{ik}} + c_{kj}^{inc_{kj}} \tag{17}$$

c_{ijk} acts as marginal cost and this condition seems to be similar to the optimality condition on the reduced cost for $\lambda_{(ij)k}$ in the simplex method, except that the dual variables π_{ij} may have any value between $c_{ij}^{dec_{ij}}$ and $c_{ij}^{inc_{ij}}$. Figure 3 represents the steps of the algorithm. Step 3 will calculate the maximum rerouting flow according to the flow in the related range of the candidate link and the remaining capacities of the related ranges of the other two links. Therefore, if all the data are integer, then the rerouting flow also has integer value.

The algorithm may be applied to the path model (1)-(5) with slightly different calculations. The rerouting cost should be defined as the difference between the link cost and the shortest distance between two endpoints of the link, using the link cost as the length. The amount of rerouting flow should also be calculated according to the remaining capacities of the related ranges for all links on the shortest path. The idea to reroute flow from a link to a shortest route according to the rerouting cost was exploited by Minoux [4] for the case of concave cost multicommodity flow problems, and has been improved for discrete cost multicommodity flow problems later [5].

Input.	The offered traffic for all OD pairs and the cost functions of the links.
Variables.	The flow, decremental and incremental ranges.
Step 0.	Route all commodities along direct links, and for each link, find the range which contains the flow, and the incremental and decremental ranges.
Step 1.	If for all c_{ijk} defined by (17), $c_{ijk} \geq 0$, then the current solution is accepted, stop. Otherwise,
Step 2.	Find the best alternative overflow, *i.e.* one with the most negative reduced costs, that is, $c_{ijk} = \min_{m,n,l} c_{mnl}$.
Step 3.	Find the maximum possible flow for rerouting, $\lambda_{(ij)k}$, that is, $\lambda_{(ij)k} \leftarrow \lambda_{(ij)k} + \min\{f_{ij} - b_{ij}^{dec_{ij}}, b_{ik}^{inc_{ik}} - f_{ik}, b_{kj}^{inc_{kj}} - f_{kj}\}$
Step 4.	Update the flow and the decremental and incremental ranges for links (*i,j*), (*i, k*) and (*k, j*), then go to Step 1.

Fig. 3. The heuristic method

Example 2. Consider a network of 3 nodes and 3 commodities with piecewise linear convex cost functions which consist of two segments, with the breakpoints equal to 12 for all links. Table 1 gives the data, and Table 2 shows the initial values of the variables. The computations of each iteration are as follows and and the values of the variables are given in Table 3-4:

Table 1. The data for Example 2

(i, j)	C_{ij}^1	C_{ij}^2	t_{ij}
(1,2)	4	7	15
(1,3)	2	3	10
(2,3)	1	2	10

Table 2. The values of the variables after Step 0

(i, j)	dec_{ij}	inc_{ij}	f_{ij}
(1,2)	2	2	15
(1,3)	1	1	10
(2,3)	1	1	10

Iteration 1:

$c_{123} = -7 + 2 + 1 = -4 < 0$

$c_{132} = -2 + 7 + 1 = 6$

$c_{231} = -1 + 7 + 2 = 8$

$\lambda_{(12)3} = \min\{3, 2, 2\}$.

Table 3. The values of the variables after the first iteration

(i, j)	dec_{ij}	inc_{ij}	f_{ij}
(1,2)	2	2	13
(1,3)	1	2	12
(2,3)	1	2	12

Iteration 2:

$c_{123} = -7 + 3 + 2 = -2 < 0$

$\lambda_{(12)3} = 2 + \min\{1, \infty, \infty\}$

Table 4. The values of the variables after the second iteration

(i, j)	dec_{ij}	inc_{ij}	f_{ij}
(1,2)	1	2	12
(1,3)	2	2	13
(2,3)	2	2	13

Iteration 3:

$c_{123} = -4 + 2 + 3 = 1$

$c_{132} = -3 = 7 = 2 = 6$

$c_{231} = -1 + 3 + 7 = 9$

Stop.

6. COMPUTATIONAL RESULTS AND CONCLUSION

In practice the overflow model gets the optimal solution faster than the path model, even for networks with a relatively small number of OD pairs. Furthermore, the overflow model is simpler and more efficient when there is a large number of commodities; the number of overflow variables is of order $|N|^3$.

The algorithms for the methods are coded in C and run on a Sun Sparc Station using Unix. First, for some given number of nodes and traffic density, random networks are generated and the cost functions are made up of 5 linear segments with the slopes proportional to the length of the edges. The time comparison of the revised simplex method for two models (path and overflow) of the problem is given in Table 5. The first two columns show the number of nodes and commodities, while the second two columns show the running times of the revised simplex method for the two models in seconds. Increase in the number of OD pairs greatly decreases the performance of the revised simplex method for the path model, even though the method uses a column generation routine. Obviously, it does not affect the result of the overflow model, as all $|N|*(|N|-1)*(|N|-2)$ possible overflow variables are checked for optimality, regardless of the number of OD pairs. Even for relatively small networks, the model with path variables cannot be solved in a reasonable time, and for networks with 16 nodes and more, the CPU times are not reported. Experience shows that several factors may affect the efficiency of the revised simplex method such as the number of nodes and commodities and the structure of the costs, especially the number of ranges and the magnitude of the offered traffic between OD pairs compared to the length of the ranges.

The last two columns of Table 5 show the results for the heuristic method applied to two models of the problem. The results show the ratio (percentage) of the best cost obtained by the heuristic method to the optimal cost obtained by the revised simplex method and the CPU times (in seconds) for the running of the algorithm. A comparison between the results for two models, the path and overflow models, shows that the heuristic method for the overflow model is often more efficient.

Column 5 of Table 5 shows the results for the Frank and Wolfe method, which has been known to efficiently solve convex network flow problems [6]. Starting from a feasible flow and using the gradients at that point as the length of the links, the Frank and Wolfe method finds the shortest path between every OD pair (or commodity) and then forms a new flow for the next iteration until the necessary conditions for optimality are obtained. To apply the Frank and Wolfe method, the cost function is required to be differentiable. The piecewise linear convex function is approximated at the neighborhood of the breakpoints by a polynomial function. For the neighboring points of each breakpoint the mean of the slopes of the adjacent line segments is considered as the gradient. As may be seen, the heuristic method for the overflow model works better than this method for the test examples. In fact, the heuristic method tries to make improvements by using a very simple procedure rather than re–finding the shortest path as in the Frank and Wolfe method.

The size of the overflow model, $|N|^3$, is independent from the number K of commodities. In the heuristic method, the flow on each link overflows on a suitable link after, at most, $R*|N|$ iterations. So it takes, at most, $R*|N|^3$ to find an acceptable overflow for all links. Therefore the complexity of the heuristic method will be $R*|N|^3$.

One advantage of the heuristic method is that if all the data are integer, its solution will also be integer. It may be useful to work on the improvement of the heuristic method for the overflow model. How to handle the worse cases and how to find an applicable sufficient condition for optimality will be investigated in the future.

Table 5. A comparison of the results and running–times for different algorithms

node	OD	Revised Simplex (path) time	Revised Simplex (overflow) time	Frank & Wolfe cost — time	Heuristic(path) cost — time	Heuristic(overflow) cost — time
5	4	0.09	0.06	103%— .03	100% — 0.03	101% — 0.01
7	8	1.27	0.74	102%—0.04	100.3%— 0.06	100.5%— 0.03
7	21	1.35	0.66	103%—0.12	100.1%— 0.08	100.1%— 0.03
9	30	2.39	0.65	103%—0.06	100.3%— 0.24	100.2%— 0.09
11	23	220.31	42.52	106%—0.43	105% — 0.83	101% — 0.22
12	11	11.71	32.35	103%—0.21	100.1%— 0.16	100.1%— 0.09
12	14	379.07	31.97	102%—0.18	101% — 0.27	101% — 0.15
12	18	413.87	32.97	103%—0.38	100.5%— 0.32	100.2%— 0.15
14	39	2289.79	234.43	104%—0.68	103% — 2.47	100% — 0.51
16	23	–	41.63	101%—0.41	100.3%— 0.42	100.3%— 0.37
17	29	–	805.08	102%—0.65	100.9%— 1.81	100.4%— 0.48

REFERENCES

1. Ahuja, R. K., Magnanti, T. L. & Orlin, J. O. (1993). Network flows, Prentice Hall, Englewood Cliffs.
2. Ketabi, S. (1997). Network design and routing problems with piecewise linear costs. PhD Thesis, The University of Adelaide, Australia.
3. Salzborn, F. J. M. & Ketabi, S. (1998). An Overflow Model for Network Design Problems, presented paper at ECCOXI.
4. Minoux, M., (1989). Network synthesis and optimum network design problems: Models, solution methods and application. *Networks*, *19*, 313360.
5. Gabriel, V., Knippel, A. & Minoux, M. (2003). A comparison of heuristic for the discrete cost multicommodity network optimization problem. *Journal of Heuristic*, *9*(5), 429445.
6. Leblanc, L. J. (1979). An algorithm for the discrete network design problem. Transportation Science, *9*, 283287.