

# EXTENDED LINEAR MULTI-COMMODITY MULTI-COST NETWORK AND MAXIMAL FLOW LIMITED COST PROBLEMS

Tran Quoc Chien<sup>1</sup> and Ho Van Hung<sup>2</sup>

<sup>1</sup>The University of Education, University of Danang, Danang, Vietnam

<sup>2</sup>Quangnam University, Tamky, Vietnam

## ABSTRACT

*The Graph is a powerful mathematical tool applied in many fields as transportation, communication, informatics, economy, ... In an ordinary graph, the weights of edges and vertexes are considered independently where the length of a path is the sum of weights of the edges and the vertexes on this path. However, in many practical problems, weights at a vertex are not the same for all paths passing this vertex but depend on coming and leaving edges. The presented paper develops a model of the extended linear multi-commodity multi-cost network that can be more exactly and effectively applied to model many practical problems. Then, maximal limit cost flow problems are modeled as implicit linear programming problems. On the base of dual theory in linear programming, an effective approximate algorithm is developed.*

## KEYWORDS

*Graph, Network, Multi-commodity Multi-cost Flow, Optimization, Linear-Programming.*

## 1. INTRODUCTION

Network and its flow is a powerful mathematical tool applied in many fields as transportation, communications, informatics, economics, and so on. So far, most of the applications in the new graph solely considers to the weight of edges and nodes independently, in which the path length merely is the sum of weights of edges and nodes along the path. However, in many practical problems, the weight at one node is not the same for all paths passing through that node, but also depends on coming and leaving edges. For example, the transit time on the transport network depends on the direction of transportation: turn right, turn left or go straight, even some directions are forbidden. Paper [2] proposes switching cost only for directed graphs. Therefore, it is necessary to build an extended mixed network model in order to apply more accurate and effective modeling of practical problems. Multi-commodity flow in traditional network problems have been studied in the papers [1,3,4,5,6]. Multi-commodity flow in extended network problems with extended transport networks were studied in the papers [7-11]. The paper[12] studies maximal multi-commodity multi-cost flow problems.

The paper builds the extended multi-cost multi-commodity model in sections 2 and 3 to enable modeling of more accurate and efficient real problems. Next, in section 4, the maximal limit cost multi-commodity multi-cost flow problem is defined by a hidden linear programming problem model. Based on the duality theory of linear programming, an approximation algorithm with polynomial complexity is developed in section 5.

## 2. EXTENDED LINEAR MULTI-COMMODITY MULTI-COST NETWORK

Given mixed graph  $G = (V, E)$  with node set  $V$  and edge set  $E$ . The edges may be undirected or directed. The symbol  $E_v$  is the set of edges incident vertice  $v \in V$ . There are many kinds of goods circulating on the network. Commodities share the capacities of the edges, but have different costs. The undirected edges represent the two-way edge, in which the goods on the same edge, but reverse directions share the capacity of the edge.

The symbol  $r$  is the commodity number,  $q_i > 0$  is the coefficient of conversion of goods  $i$ ,  $i = 1..r$ . Given the following functions:

*Edge passing capacity function*  $ce: E \rightarrow R^*$ , where  $ce(e)$  is the passing capability of the edge  $e \in E$ .

*Edge service coefficient function*  $ze: E \rightarrow R^*$ , where  $ze(e)$  is the passing ratio of the edge  $e \in E$  (the real capacity of the edge  $e$  is  $ze(e).ce(e)$ ).

*Node passing capability function*  $cv: V \rightarrow R^*$ , where  $cv(u)$  is the passing capability of the node  $u \in V$ .

*Node service coefficient function*  $zv: V \rightarrow R^*$ , where  $zv(u)$  is the passing ratio of the node  $v \in V$  (the real capacity of the node  $v$  is  $zv(v).cv(v)$ ).

The tuples  $(V, E, ce, ze, cv, zv)$  are called *extended networks*.

*Edge cost function*  $i, i = 1..r, be_i: E \rightarrow R^*$ , where  $be_i(e)$  is the cost of passing  $e$  a converted unit of commodity of type  $i$ . Note that with 2-way paths, the cost of each way may vary.

*Node switch cost function*  $i, i = 1..r, bv_i: V \times E_v \times E_v \rightarrow R^*$ , where  $bv_i(u, e, e')$  is the cost of transferring a converted unit of commodity of type  $i$  from edge  $e$  through  $u$  to edge  $e'$ .

The sets  $((V, E, ce, ze, cv, zv, \{be_i, bv_i, q_i | i = 1..r\}))$  are called *the extended linear multi-commodity multi-cost network*.

◇ *Note*: If  $be_i(e) = \infty$ , commodity of type  $i$  is prohibited from circulation on path  $e$ . If  $bv_i(u, e, e') = \infty$ , commodity of type  $i$  is banned from path  $e$  through  $u$  to path  $e'$ .

Let  $p$  be the path from node  $u$  to node  $v$  through edges  $e_j, j = 1..(h+1)$ , and nodes  $u_j, j = 1..h$  as follows

$$p = [u, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, v]$$

The cost of circulating a converted unit of commodity of type  $i, i = 1..r$ , passing the path  $p$ , is denoted by the symbol  $b_i(p)$ , and defined by the following formula:

$$b_i(p) = \sum_{j=1}^{h+1} be_i(e_j) + \sum_{j=1}^h bv_i(u_j, e_j, e_{j+1}) \quad (1)$$

### 3. MULTI-COMMODITY FLOW PROBLEMS IN EXTENDED LINEAR MULTI-COMMODITY MULTI-COST NETWORK

Given a multi-cost multi-commodity network  $G=(V,E,ce, ze, cv, zv, \{be_i, bv_i, q_i|i=1..r\})$ . Assume, for each commodity of type  $i, i=1..r$ , there are  $k_i$  source-target pairs  $(s_{i,j}, t_{i,j}), j=1..k_i$ , each pair assigned a quantity of commodity of type  $i$ , that is necessary to move from source node  $s_{i,j}$  to target node  $t_{i,j}$ .

Denote  $P_{i,j}$  is the set of paths from node  $s_{i,j}$  to node  $t_{i,j}$  in  $G$ , which commodity of type  $i$  can be passed,  $i=1..r, j=1..k_i$ . Set

$$P_i = \bigcup_{j=1}^{k_i} P_{i,j}. \quad (2)$$

For each path  $p \in P_{i,j}, i=1..r, j=1..k_i$ , denote  $x_{i,j}(p)$  the flow of converted commodity of type  $i$  from the source node  $s_{i,j}$  to the destination node  $t_{i,j}$  along the path  $p, i=1..r, j=1..k_i$ .

Denote  $P_{i,e}$  the set of paths in  $P_i$  passing through the edge  $e, \forall e \in E$ .

Denote  $P_{i,v}$  the set of paths in  $P_i$  passing through the node  $v, \forall v \in V$ .

A set

$$F = \{x_{i,j}(p) \mid p \in P_{i,j}, i=1..r, j=1..k_i\} \quad (3)$$

is called a *multi-commodity flow* on the linear extended multi-commodity multi-cost network, if it satisfies the following *edge and node capacity* constraints:

$$\sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,e}} x_{i,j}(p) \leq ce(e).ze(e), \forall e \in E$$

$$\sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,v}} x_{i,j}(p) \leq cv(v).zv(v), \forall v \in V$$

The expressions

$$fv_{i,j} = \sum_{p \in P_{i,j}} x_{i,j}(p), i=1..r, j=1..k_i \quad (4)$$

is called *the flow value of commodity of type  $i$  of the source-target pair  $(s_{i,j}, t_{i,j})$  of  $F$* .

The expressions

$$fv_i = \sum_{j=1}^{k_i} fv_{i,j}, i=1..r \quad (5)$$

is called *the flow value of commodity of type  $i$  of  $F$* .

The expression

$$fv = \sum_{i=1}^r fv_i \quad (6)$$

Allied *the flow value of  $F$* .

#### 4. MAXIMAL LIMITED COST MULTI-COMMODITY FLOW PROBLEMS

Given an extended linear multi-commodity multi-cost network  $G=(V,E, ce, ze, cv, zv, \{be_i, bv_i, q_i|i=1..r\})$ . Assume, for each commodity of type  $i, i=1..r$ , there are  $k_i$  source-target pairs  $(s_{i,j}, t_{i,j}), j=1..k_i$ , each pair assigned a quantity of commodity of type  $i$ , that is necessary to move from source node  $s_{i,j}$  to target node  $t_{i,j}$ . Given a limit cost  $B$ .

The task of the problem is to find the multi-commodity flow such that the value of the flow  $fv$  is maximal. At the same time, the total cost of the flow does not exceed  $B$ .

The problem is expressed by an implicit linear programming model  $(P)$  as follows:

$$\begin{aligned}
 fv &= \sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,j}} x_{i,j}(p) \rightarrow \max \\
 &\text{satisfies} \\
 &\sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,e}} x_{i,j}(p) \leq ce(e).ze(e), \forall e \in E \\
 &\sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,v}} x_{i,j}(p) \leq cv(v).zv(v), \forall v \in V \\
 &\sum_{i=1}^r \sum_{j=1}^{k_i} \sum_{p \in P_{i,j}} x_{i,j}(p).b_i(p) \leq B \\
 &x_{i,j}(p) \geq 0, \forall i=1..r, j=1..k_i, \forall p \in P_{i,j}
 \end{aligned} \quad (P)$$

The dual linear programming problem of  $(P)$ , called  $(D)$ , is constructed as follows: each edge  $e \in E$  is assigned an dual variable  $le(e)$ , each node  $v \in V$  is assigned an dual variable  $lv(v)$  while the dual variable  $\varphi$  assigns the constraint of cost. The problem  $(D)$  states the following

$$\begin{aligned}
 D(le, lv, \varphi) &= \sum_{e \in E} ce(e).ze(e).le(e) + \sum_{v \in V} cv(v).zv(v).lv(v) + B.\varphi \rightarrow \min \\
 &\sum_{e \in p} le(e) + \sum_{v \in p} lv(v) + b_i(p).\varphi \geq 1, \forall p \in P_{i,j} \\
 &le(e) \geq 0, \forall e \in E, lv(v) \geq 0, \forall v \in V, \varphi \geq 0
 \end{aligned} \quad (D)$$

Now, given  $p \in P_i, i=1..r$ , a path from node  $u$  to node  $v$  through edges  $e_j, j=1..(h+1)$ , and nodes  $u_j, j=1..h$ , as follows

$$p = [u, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, v]$$

We define the path length of  $p$ , denoted by  $length_i(p)$ , depending on the variables  $le(e), lv(v)$  and  $\varphi$  so as to the following formula:

$$length_i(p) = \sum_{j=1}^{h+1} le(e_j) + \sum_{j=1}^h lv(u_j) + b_i(p).\varphi \quad (7)$$

Denote  $dist_{i,j}(le,lv,\varphi)$  the shortest path length from  $s_{i,j}$  to  $t_{i,j}$  calculated by function  $length_i(p)$ ,  $\forall i=1..r, \forall j=1..k_i$ .

Set  $\alpha(le,lv,\varphi) = \min\{dist_{i,j}(le,lv,\varphi) \mid i=1..r, j=1..k_i\}$

Consider the problem  $(D_\alpha)$ :

$$\beta = \min \left\{ \frac{D(le,lv,\varphi)}{\alpha(le,lv,\varphi)} \mid le: E \rightarrow R^*, lv: V \rightarrow R^*, \varphi \geq 0 \right\} \quad (8)$$

• **Lemma 4.1.** The problem  $(D)$  is equivalent to the problem  $(D_\alpha)$  such that their optimal value are equal and the optimal solution of one problem derives the optimal solution of the other problem and vice versa.

*Proof*

Denote  $min(D)$  and  $min(D_\alpha)$ , respectively, the optimal values of the problem  $(D)$  and the problem  $(D_\alpha)$ . Given functions  $le: E \rightarrow R^*, lv: V \rightarrow R^*, \varphi > 0$ . Set

$$le'(e) = le(e)/\alpha(le,lv,\varphi), \forall e \in E, lv'(v) = lv(v)/\alpha(le,lv,\varphi), \forall v \in V, \varphi' = \varphi/\alpha(le,lv,\varphi). \quad (9)$$

We have:

$$\sum_{e \in p} le'(e) + \sum_{v \in p} lv'(v) + b_i(p) \cdot \varphi' \geq 1, \forall i=1..r, \forall j=1..k_i, \forall p \in P_{i,j} \quad (10)$$

So  $(le',lv',\varphi')$  is an accepted solution of  $(D)$  and

$$D(le',lv',\varphi') = \frac{D(le,lv,\varphi)}{\alpha(le,lv,\varphi)}. \quad (11)$$

Hence inferred

$$min(D) \leq min(D_\alpha) \quad (12)$$

In contrast, let  $(le,lv,\varphi)$  be an accepted solution of  $(D)$ . Then, we have

$$1 \leq dist_{i,j}(le,lv,\varphi), \forall i=1..r, \forall j=1..k_i \Rightarrow \alpha(le,lv,\varphi) \geq 1 \Rightarrow \frac{D(le,lv,\varphi)}{\alpha(le,lv,\varphi)} \leq D(le,lv,\varphi).$$

Hence Inferred

$$min(D) \geq min(D_\alpha) \quad (13)$$

From (12) and (13), it follows  $min(D) = min(D_\alpha)$ .

Next, if  $(le,lv,\varphi)$  is an optimal solution of the problem  $(D_\alpha)$ , then  $(le',lv',\varphi')$  where  $le'(e) = le(e)/\alpha(le,lv,\varphi), \forall e \in E, lv'(v) = lv(v)/\alpha(le,lv,\varphi), \forall v \in V, \varphi' = \varphi/\alpha(le,lv,\varphi)$ . is an optimal solution of problem  $(D)$ .

Conversely, if  $(le,lv,\varphi)$  is an optimal solution of the problem  $(D)$ , then  $(le,lv,\varphi)$  is an optimal solution of the problem  $(D_\alpha)$ .

## 5. ALGORITHM

### • IDEAS

The algorithm consists of a number of iterative steps, through the function  $length_i(p)$ ,  $p \in P_i$ ,  $i=1..r$ . At each iteration step, find the shortest path  $p$ , with respect to the length function  $length_i(p)$  between the source-destination pairs and convert  $c$  units of exchange through  $p$ , where  $c$  is the minimal edge and node capacity on this path.

Then change the value of functions  $le$ ,  $lv$  and  $\varphi$ . The algorithm stops once  $\alpha \geq 1$ . The initial value of  $le$ ,  $lv$  and  $\varphi$  depends on the approximate value to be achieved.

### • ALGORITHM

◇ **INPUT:** Extended multi-cost multi-commodity network  $G=(V,E, ce, ze, cv, zv, \{be_i, bv_i, q_i | i=1..r\})$ . Assume, for each commodity of type  $i$ ,  $i=1..r$ , there are  $k_i$  source-target pairs  $(s_{i,j}, t_{i,j})$ ,  $j=1..k_i$ , each pair assigned a quantity of commodity of type  $i$ , that is necessary to move from source node  $s_{i,j}$  to target node  $t_{i,j}$ . Given a limited cost  $B$ , an approximation ratio  $\omega$ .

◇ **OUTPUT :** Maximal flow  $F$  represents a set of converged flows at the edges  
 $F = \{x_{i,j}(e) | e \in E, i=1..r, j=1..k_i\}$

with total cost not over the limit cost  $B$ .

### ◇ PROCEDURE

Note  $n=|V|$ ,  $m=|E|$ ,  $B_f$  total cost of the flows  $F$ .

Calculate  $bmin$ , the smallest cost in the paths from the source  $s_{i,j}$  to the destination  $t_{i,j}$ ,  $i=1..r$ ,  $j=1..k_i$ :

$$bmin = \min\{b_i(p) | i=1..r, j=1..k_i, p \in P_{i,j}\}. \quad (14)$$

Calculate  $bmax$ , the largest cost of the paths from the source  $s_{i,j}$  to the destination  $t_{i,j}$ ,  $i=1..r$ ,  $j=1..k_i$ .

$$bmax = \max\{b_i(p) | i=1..r, j=1..k_i, p \in P_{i,j}\}. \quad (15)$$

Set

$$\varepsilon = 1 - \sqrt{1/(1+\omega)} ; \delta = (1+\varepsilon) \frac{1}{[(1+\varepsilon)^2(m+n+bmax/bmin)]^{1/\varepsilon}} ; \quad (16)$$

$$\text{for } e \in E : le(e) = \delta, x_{i,j}(e) = 0 ; \text{ for } v \in V : lv(v) = \delta ; \quad (17)$$

$$\varphi = \delta bmin ; fv = 0 ; B_f = 0 ; \quad (18)$$

**do**

{

Use the algorithm to find the source-destination pair  $(s_{i,j}, t_{i,j})$ ,  $1 \leq i \leq r$  and  $1 \leq j \leq k_i$ , having the shortest path from  $s_{i,j}$  to  $t_{i,j}$  calculated by the length function  $length_i(\cdot)$ . Note that the path  $p$  must be valid

for goods of type  $i$ , i.e., not containing the edge with edge cost  $\infty$  or the node with the switch cost  $\infty$ .

Note

$imin$  and  $jmin$  the indexes of source-destination pairs with the shortest path;  
 $\alpha$  is the shortest path length;  
 $p$  is the shortest path;  
 $c$  is the minimal edge and node capacity on the path  $p$ , i.e.

$$c = \min\{\min\{ce(e).ze(e) | e \in p\}, \min\{cv(v).zv(v) | v \in p\}\};$$

$$B' = c.b_{imin}(p);$$

$$\text{if } (B' > B) \{ c = c.B / B'; B' = B; \}$$

Flow adjustments:

$$\forall e \in p, x_{i,j}(e) = x_{i,j}(e) + c; le(e) = le(e) \cdot (1 + \varepsilon \cdot c / (ce(e).ze(e))); \quad (19)$$

$$\forall v \in p, lv(v) = lv(v) \cdot (1 + \varepsilon \cdot c / (cv(v).zv(v))); \quad (20)$$

$$\varphi = \varphi \cdot (1 + \varepsilon \cdot B' / B); B_f = B_f + B'; fv = fv + c; \quad (21)$$

} while ( $\alpha < 1$ )

Modifying the resulting flows  $F$  and flow value  $fv$ .

$$x_{i,j}(e) = x_{i,j}(e) / \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}; \forall i=1..r, j=1..k_i, \forall e \in E; \quad (22)$$

$$fv = fv / \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}; B_f = B_f / \log_{1+\varepsilon} \frac{1}{\delta}; \quad (23)$$

Modifying flows on scalar edge

for ( $i=1; i \leq r; i++$ )  
 for ( $j=1; j \leq k_i; j++$ )  
 for scalar  $e \in E$

if  $x_{i,j}(e) > x_{i,j}(e') // e'$  is the opposite of the direction  $e$   
 $\{x_{i,j}(e) = x_{i,j}(e) - x_{i,j}(e'); x_{i,j}(e') = 0\};$   
 else  
 $\{x_{i,j}(e') = x_{i,j}(e') - x_{i,j}(e); x_{i,j}(e) = 0\};$

#### • PROOF OF ALGORITHM

Denote

$D(0)$  the initial value of function  $D$

$$D(0) = \sum_{e \in E} ce(e).ze(e) \cdot \delta + \sum_{v \in V} cv(v).zv(v) \cdot \delta + B \cdot \delta bmin$$

$$= \delta \left( \sum_{e \in E} ce(e).ze(e) + \sum_{v \in V} cv(v).zv(v) + B/bmin \right) \quad (24)$$

$D(i)$  is the value of the function  $D$  after the loop  $i, i=1, 2, \dots$

$fv(0) = 0$  is the initial value of the flow  $F$ .

$fv(i)$  is the value of the flow  $F$  after the loop  $i$ ,  $i=1, 2, \dots$

$le_0, lv_0$  and  $\varphi_0$  are, respectively, initial values of the functions  $le$ ,  $lv$  and  $\varphi$ .

$le_i, lv_i$  and  $\varphi_i$  correspond respectively to  $le$ ,  $lv$  and  $\varphi$  in the loop  $i$ .  $i=1, 2, \dots$

$p_i$  is the shortest path  $p$  in the loop  $i$ .  $i=1, 2, \dots$

$\alpha(i) = \alpha(le_i, lv_i, \varphi_i)$ ,  $i = 1, 2, \dots$

$c(i)$  is the value of  $c$  in the loop  $i$ .  $i = 1, 2, \dots$

We have

$$fv(j) = fv(j-1) + c(j), \quad (25)$$

$$\begin{aligned} D(j) &= \sum_{e \in E} ce(e).ze(e).le_j(e) + \sum_{v \in V} cv(v).zv(v).lv_j(v) + B.\varphi_j \\ &= \sum_{e \in E} ce(e).ze(e).le_{j-1}(e) + \sum_{v \in V} cv(v).zv(v).lv_{j-1}(v) + B.\varphi_{j-1} \\ &+ \sum_{e \in p_j} \varepsilon.c(j)le_{j-1}(e) + \sum_{v \in p_j} \varepsilon.c(j)lv_{j-1}(v) + \varepsilon.B'.\varphi_{j-1} \\ &= D(j-1) + \varepsilon.c(j).(\sum_{e \in p_j} le_{j-1}(e) + \sum_{v \in p_j} lv_{j-1}(v) + b_{\min}(p_{j-1}).\varphi_{j-1}) \\ &= D(j-1) + \varepsilon.c(j).\alpha(j-1). \end{aligned} \quad (26)$$

It follows

$$D(i) = D(0) + \varepsilon \sum_{j=1}^i (fv(j) - fv(j-1))\alpha(j-1), \quad \forall i \geq 1 \quad (27)$$

Consider functions  $le_i - le_0$  và  $lv_i - lv_0$  và  $\varphi_i - \varphi_0$ . We have

$$D(le_i - le_0, lv_i - lv_0, \varphi_i - \varphi_0) = D(i) - D(0). \quad (28)$$

Next, consider  $\alpha(le_i - le_0, lv_i - lv_0, \varphi_i - \varphi_0)$ . We have

$$\alpha(le, lv, \varphi) = \min\{dist_{g,l}(le, lv, \varphi) \mid g=1..r, l=1..k_g\} \quad (29)$$

and

$$\begin{aligned} &dist_{g,l}(le, lv, \varphi) \\ &= \min\left\{ \sum_{j=1}^{h+1} le(e_j) + \sum_{j=1}^h lv(u_j) + b_g(p).\varphi \mid p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \end{aligned} \quad (30)$$

Thus

$$\begin{aligned} &dist_{g,l}(le_i - le_0, lv_i - lv_0, \varphi_i - \varphi_0) \\ &= \min\left\{ \sum_{j=1}^{h+1} (le(e_j) - le_0(e_j)) + \sum_{j=1}^h (lv(u_j) - lv_0(u_j)) + b_g(p).\varphi_i - b_g(p).\varphi_0 \right. \\ &\quad \left. \mid p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \end{aligned} \quad (31)$$

$$\begin{aligned} &= \min\left\{ \sum_{j=1}^{h+1} le_i(e_j) + \sum_{j=1}^h lv_i(u_j) + b_g(p).\varphi_i - \left( \sum_{j=1}^{h+1} le_0(e_j) + \sum_{j=1}^h lv_0(u_j) + b_g(p).\varphi_0 \right) \right. \\ &\quad \left. \mid p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \end{aligned} \quad (32)$$



$$\geq \min \left\{ \sum_{j=1}^{h+1} l e_i(e_j) + \sum_{j=1}^h l v_i(u_j) + b_g(p), \varphi_i - (m \cdot \delta + n \cdot \delta + \delta b_g(p) / b_{\min}) \right. \\ \left. | p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \quad (33)$$

$$\geq \min \left\{ \sum_{j=1}^{h+1} l e_i(e_j) + \sum_{j=1}^h l v_i(u_j) + b_g(p), \varphi_i - \delta(m + n + b_{\max} / b_{\min}) \right. \\ \left. | p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \quad (34)$$

$$= \min \left\{ \sum_{j=1}^{h+1} l e_i(e_j) + \sum_{j=1}^h l v_i(u_j) + b_g(p), \varphi_i | p = [s_{i,j}, e_1, u_1, e_2, u_2, \dots, e_h, u_h, e_{h+1}, t_{i,j}] \in P_{g,l} \right\} \\ - \delta(m + n + b_{\max} / b_{\min}) \quad (35)$$

$$= \text{dist}_{g,l}(l e_i, l v_i, \varphi_i) - \delta(m + n + b_{\max} / b_{\min}) \quad (36)$$

Hence,

$$\alpha(l e_i - l e_0, l v_i - l v_0, \varphi_i - \varphi_0) \geq \alpha(i) - \delta(m + n + b_{\max} / b_{\min}). \quad (37)$$

Denote  $i_0$  the smallest index satisfies

$$\alpha(i_0) \geq \delta(m + n + b_{\max} / b_{\min}) \text{ and } \alpha(i_0 - 1) < \delta(m + n + b_{\max} / b_{\min}) \quad (38)$$

On the other hand, we have

$$l e_i(e) \leq l e_{i-1}(e) \cdot (1 + \varepsilon \cdot c(j) / (c e(e) \cdot z e(e))) \leq l e_{i-1}(e) \cdot (1 + \varepsilon), \forall e \in E, \forall i \geq 1 \quad (39)$$

$$l v_i(v) \leq l v_{i-1}(v) \cdot (1 + \varepsilon \cdot c(j) / (c v(v) \cdot z v(v))) \leq l v_{i-1}(v) \cdot (1 + \varepsilon), \forall v \in V, \forall i \geq 1 \quad (40)$$

and

$$\varphi_i = \varphi_{i-1} (1 + \varepsilon \cdot B' / B) \leq \varphi_{i-1} (1 + \varepsilon), \quad (41)$$

$$\Rightarrow \alpha(i) \leq (1 + \varepsilon) \cdot \alpha(i - 1), \forall i \geq 1. \quad (42)$$

It follows

$$\alpha(i_0) \leq (1 + \varepsilon) \cdot \alpha(i_0 - 1) < (1 + \varepsilon) \cdot \delta(m + n + b_{\max} / b_{\min}) \quad (43)$$

Then, we have

$$\beta \leq \frac{D(l e_i - l e_0, l v_i - l v_0, \varphi_i - \varphi_0)}{\alpha(l e_i - l e_0, l v_i - l v_0, \varphi_i - \varphi_0)} \leq \frac{D(i) - D(0)}{\alpha(i) - \delta(m + n + b_{\max} / b_{\min})}, \forall i \geq i_0 + 1 \quad (44)$$

Substituting  $D(i) - D(0)$  from (27) into (44), we obtain:

$$\alpha(i) \leq \delta(m + n + b_{\max} / b_{\min}) + \frac{\varepsilon}{\beta} \sum_{j=1}^i (f v(j) - f v(j - 1)) \alpha(j - 1), \forall i \geq i_0 + 1 \quad (45)$$

We define  $x(i_0), x(i_0 + 1), \dots, x(i_0 + i), \dots$  as follows:

$$x(i_0) = \alpha(i_0) \text{ v\`a } x(i) = \delta(m + n + b_{\max} / b_{\min}) + \frac{\varepsilon}{\beta} \sum_{j=1}^i (f v(j) - f v(j - 1)) x(j - 1), \forall i \geq i_0 + 1 \quad (46)$$

Inductively, it follows

$$\alpha(i) \leq x(i), \forall i \geq i_0 + 1 \quad (47)$$

We have

$$x(i) = \delta(m+n+bmax/bmin) + \frac{\varepsilon}{\beta} \sum_{j=1}^{i-1} (fv(j) - fv(j-1))x(j-1) + \frac{\varepsilon}{\beta} (fv(i) - fv(i-1))x(i-1) \quad (48)$$

$$= x(i-1) (1 + \varepsilon(fv(i) - fv(i-1))/\beta) \leq x(i-1) e^{\varepsilon(fv(i) - fv(i-1))/\beta} \leq x(i-2) e^{\varepsilon(fv(i) - fv(i-2))/\beta} \leq \dots \quad (49)$$

$$\leq x(i_0) e^{\varepsilon(fv(i) - fv(i_0))/\beta} \leq \alpha(i_0) \cdot e^{\varepsilon \cdot fv(i)/\beta} \leq (1 + \varepsilon) \cdot \delta(m+n+bmax/bmin) e^{\varepsilon \cdot fv(i)/\beta}, \quad (50)$$

for  $\alpha(i_0) \leq (1 + \varepsilon) \cdot \delta(m+n+bmax/bmin)$ .

It then follows

$$\alpha(i) \leq (1 + \varepsilon) \cdot \delta(m+n+bmax/bmin) e^{\varepsilon \cdot fv(i)/\beta}, \forall i \geq i_0 + 1. \quad (51)$$

Suppose the algorithm ends in the loop  $t$ ,  $\alpha(t) \geq 1$ . Then

$$1 \leq (1 + \varepsilon) \cdot \delta(m+n+bmax/bmin) e^{\varepsilon \cdot fv(t)/\beta} \quad (52)$$

that follows

$$\frac{\beta}{fv(t)} \leq \frac{\varepsilon}{\ln \frac{1}{(1 + \varepsilon) \cdot \delta(m+n+bmax/bmin)}} \quad (53)$$

**Lemma 5.1.** There exists an accepted flow with value  $\frac{fv(t)}{\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}}$ .

*Proof.* Consider any edge  $e$ . We have :

For each transfer of  $ce(e)ze(e)$  converted units of commodities through  $e$ , the length  $le(e)$  of  $e$  increases by a factor  $\geq (1 + \varepsilon)$ . Indeed, at each iteration we only transfer  $c \leq ce(e)ze(e)$  converted units of commodities through  $e$ . So, in order to transfer of  $ce(e)ze(e)$  converted units of commodities through  $e$ , commodities must be transferred through  $e$  at least in one iteration. Suppose it starts at iteration  $i$ . Let  $q$  be the number of iterations to transfer of  $ce(e)ze(e)$  converted units of commodities through  $e$ . Denote  $c_j$  the value of  $c$  at the  $j$ -th transfer through  $e$ ,  $j=1..q$ . Denote  $l$  the last iteration that transfers  $c_q$  converted units of commodities through  $e$ . We have

$$le_l(e) = le_i(e) \cdot \left(1 + \varepsilon \cdot \frac{c_1}{ce(e)ze(e)}\right) \dots \left(1 + \varepsilon \cdot \frac{c_q}{c(e)ze(e)}\right) \quad (54)$$

$$= le_i(e) \cdot \left(1 + (c_1 + \dots + c_q) \frac{\varepsilon}{c(e)ze(e)} + \dots\right) \quad (55)$$

Then, we have

$$\sum_{j=1}^q c_j \geq ce(e)ze(e). \quad (56)$$

Thus

$$le_i(e) \geq (1+\varepsilon).le_i(e). \quad (57)$$

Assumes  $x(e)$  be the sum of the flows through  $e$ . Set

$$x(e) = h.ce(e)ze(e), h \in R^*. \quad (58)$$

So there are  $h$  times of  $ce(e)ze(e)$  unit of goods passes  $e$ , then the length of  $e$  is

$$le_i(e) \geq (1+\varepsilon)^h.le_0(e) \quad (59)$$

Let  $j, j < t$ , be the last iteration of increasing  $le(e)$ . Then  $le_j(e) < 1$ , (for if  $le_j(e) \geq 1$ , then  $\alpha(j) \geq 1$ , and the algorithm stopped at step  $j < t$ ). So,

$$le_i(e) \leq (1+\varepsilon).le_j(e) < (1+\varepsilon) \quad (60)$$

From (59), (60) and  $le_0(e) = \delta$  it follows

$$(1+\varepsilon)^h.le_0(e) \leq le_i(e) < (1+\varepsilon) \Rightarrow h < \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}. \quad (61)$$

So

$$x(e) \leq ce(e)ze(e). \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}, \forall e \in E. \quad (62)$$

Similarly, we also have

$$x(v) \leq cv(v).zv(v). \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}, \forall v \in V \quad (63)$$

where  $x(v)$  is the total converted units of commodities passing through the node  $v \in V$ .

Thus, divide all flows  $x_{i,j}(e)$  by  $\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ :

$$x_{i,j}(e) = x_{i,j}(e) / \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}, \forall i=1..r, j=1..k_i, \forall e \in E, \quad (64)$$

we receive accepted flows, which satisfies the conditions of edge and node capacities, with the flow value of  $\frac{fv(t)}{\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}}$ .

**Lemma 5.2.** Flow with value  $\frac{fv(t)}{\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}}$  is maximal with approximation ratio  $1+\omega$ .

*Proof*

Denote  $\gamma$  the ratio of the dual optimal value divided by the flow value of  $\{x_{i,j}(e) \mid e \in E, i=1..r, j=1..k_i\}$

We have

$$\gamma = \frac{\beta}{fv(t)} . \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}. \quad (65)$$

Using (53) we have

$$\gamma \leq \frac{\varepsilon}{\ln \frac{1}{(1+\varepsilon)\delta(m+n+b \max/b \min)}} \cdot \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta} = \frac{\varepsilon}{\ln(1+\varepsilon)} \cdot \frac{\ln \frac{1+\varepsilon}{\delta}}{\ln \frac{1}{(1+\varepsilon)\delta(m+n+b \max/b \min)}} \quad (66)$$

Select

$$\delta = (1+\varepsilon) \frac{1}{[(1+\varepsilon)^2(m+n+b \max/b \min)]^{1/\varepsilon}} \quad (67)$$

We have

$$\frac{\ln \frac{1+\varepsilon}{\delta}}{\ln \frac{1}{(1+\varepsilon)\delta(m+n+b \max/b \min)}} = (1-\varepsilon)^{-1} \quad (68)$$

$\Rightarrow$

$$\gamma \leq \frac{\varepsilon}{(1-\varepsilon)(\varepsilon - \varepsilon^2/2)} \leq \frac{1}{(1-\varepsilon)^2} \quad (69)$$

So  $1 < \gamma \leq \frac{1}{(1-\varepsilon)^2}$ . In order to get approximation ratio  $(1+\omega)$ , we choose  $\varepsilon$  such that  $\frac{1}{(1-\varepsilon)^2} \leq (1+\omega)$ . So, with  $0 < \varepsilon \leq 1 - \sqrt{1/(1+\omega)}$ , the flow with value  $\frac{fv(t)}{\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}}$  is maximal at

approximation ratio  $1+\omega$ .

• **Note.** From the converting flows  $\{x_{i,j}(e) \mid e \in E, i=1..r, j=1..k_i\}$  we can deduce the actual flows by dividing the converted flows  $x_{i,j}(e)$  by the conversion factor  $q_i, \forall i=1..r, j=1..k_i, \forall e \in E$ .

• **Lemma 5.3.** The total flow cost in  $t$  iterations does not exceed  $B \cdot \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ . That is, the total cost of the flow after divide by  $\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$  does not exceed  $B$ .

*Proof.* We have  $\varphi_0 = \delta b \min$ . After  $(t-1)$  iterations, we have  $\alpha(t-1) < 1$ , it mean

$$\sum_{e \in p_{t-1}} l_{e_{t-1}}(e) + \sum_{v \in p_{t-1}} l_{v_{t-1}}(v) + b_{\min}(p_{t-1}) \cdot \varphi_{t-1} < 1 \quad (70)$$

$$\Rightarrow \varphi_{t-1} < 1/b_{\min}(p_{t-1}) \leq 1/b \min. \quad (71)$$

Furthermore, while transferring commodities on the networks makes the total cost increasing to  $B$ , so that  $\varphi$  increases to a factor of not less than  $(1+\varepsilon)$ . Thus, calling  $x$  is the number of times the algorithm increases the cost per unit  $B$  in the  $t$  loop, we have  $\varphi_0 \cdot (1+\varepsilon)^x \leq \varphi_t \leq (1+\varepsilon) \cdot \varphi_{t-1} \leq (1+\varepsilon) / b \min$ . Due  $\varphi_0 = \delta b \min$ , it follows  $x \leq \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ .

So the total cost of the flows after  $t$  iterations does not exceed  $B \cdot \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ . When dividing the flow by  $\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ , we also have the total cost decreasing factor  $\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$  to satisfy the requirements of the problem.

## 6. ALGORITHM COMPLEXITY

**Theorem 6.1.** The algorithm has a complexity of

$$O(\omega^{-2} \cdot k \cdot n^3 \cdot (m+n) \cdot \ln(m+n+b_{\max}/b_{\min})),$$

where  $k=k_1+\dots+k_r$ ,  $m$  is the number of edges in the graph,  $n$  is the number of vertices.

*Proof.* Consider the iterations  $i$ . Suppose  $e$  is the edge capacity of passing the smallest  $ce(e)ze(e)=c(i)$  along the shortest path  $p_i$ . We increase the  $le_i(e)$  of  $e$  factor  $(1+\varepsilon)$ . Considering any  $e$ , let  $te$  be the number of iterations in which  $e$  has minimal capacity on corresponding path. Since  $le_0(e) = \delta$  and  $le_t(e) < 1+\varepsilon$ , we have

$$le_0(e) \cdot (1+\varepsilon)^{te} = \delta \cdot (1+\varepsilon)^{te} \leq le_t(e) < 1+\varepsilon, \quad (72)$$

what follows  $te < \log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}$ .

On the other hand,

$$\delta = (1+\varepsilon) \frac{1}{\left[ (1+\varepsilon)^2 (m+n+b_{\max}/b_{\min}) \right]^{1/\varepsilon}} \quad (73)$$

$$\Rightarrow te \leq \frac{1}{\varepsilon} (2 + \log_{1+\varepsilon} (m+n+b_{\max}/b_{\min})) \quad (74)$$

$$\leq \frac{1}{\varepsilon \cdot \ln(1+\varepsilon)} (2 \ln(1+\varepsilon) + \ln(m+n+b_{\max}/b_{\min})). \quad (75)$$

Set

$$t^* = \frac{1}{\varepsilon \cdot \ln(1+\varepsilon)} (2 \ln(1+\varepsilon) + \ln(m+n+b_{\max}/b_{\min})). \quad (76)$$

Then, each edge  $e \in E$  corresponds to at most  $t^*$  times of finding shortest paths.

Similarly, each node  $v \in V$  corresponds to at most  $t^*$  times of finding shortest paths.

So the times of finding shortest paths  $\leq (m+n) \cdot t^*$ .

The algorithm that finds the shortest path between the two source-destination ends has a complexity of  $O(n^3)$  [7, 8], which inferred the shortest path finding algorithm between  $k$  pairs of destination source ends with complexity  $O(k \cdot n^3)$ . Inferring the complexity of the algorithm is

$$O(k \cdot n^3) \cdot O((m+n) \cdot \frac{\ln(m+n+b_{\max}/b_{\min})}{\varepsilon \cdot \ln(1+\varepsilon)}) \quad (77)$$

On the other hand, because of  $\varepsilon \leq 1 - \sqrt{1/(1+\omega)}$ , we have

$$\varepsilon \cdot \ln(1+\varepsilon) \approx \varepsilon^2 \leq (1 - \sqrt{1/(1+\omega)})^2 = (1 - (1 - 0.5\omega + o(\omega)))^2 = O(\omega^2) \quad (78)$$

Finally, from (77) and (78) it deduce the complexity of the algorithm

$$O(\omega^{-2} \cdot k \cdot n^3 \cdot (m+n) \cdot \ln(m+n+b_{\max}/b_{\min})).$$

## 7. CONCLUSIONS

The paper develops a model of extended linear multi-commodity multi-cost network that can be more exactly and effectively applied to model many practical problems. Then, maximal limit cost flow problems are modeled as implicit linear programming problems. On the base of dual theory in linear programming an effective approximate algorithm is developed. Correctness and algorithm complexity are justified. The results of this paper are the basis for studying the multi-commodity multi-cost flow optimization problem.

At last we emphasize that there is no efficient method solving implicit linear problems. Otherwise, it is almost impossible to present this problem as an explicit linear problem, for the number of all paths  $p$ , which determines the variables  $x(p)$ , is as big as  $O(n^n)$ . So, in case this problem is converted to an explicit linear problem and solved by any known method (f.e. simplex method), the complexity is much bigger than  $O(n^n)$ , that is not practically acceptable.

## REFERENCES

- [1] Naveen Garg, Jochen Könemann: Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems, SIAM J. Comput, Canada, 37(2), 2007, pp. 630-652.
- [2] Xiaolong Ma, Jie Zhou: An Extended Shortest Path Problem with Switch Cost Between Arcs, Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol IIMECS 2008, 19-21 March, 2008, Hong Kong.
- [3] Tran Quoc Chien: Linear multi-channel traffic network, Ministry of Science and Technology, code B2010DN-03-52.
- [4] Tran Quoc Chien, Tran Thi My Dung: Application of the shortest path finding algorithm to find the maximum flow of goods. Journal of Science & Technology, University of Danang, 3 (44) 2011.
- [5] Tran Quoc Chien: Application of the shortest multi-path finding algorithm to find the maximum simultaneous flow of goods simultaneously. Journal of Science & Technology, University of Danang, 4 (53) 2012.
- [6] Tran Quoc Chien: Application of the shortest multi-path finding algorithm to find the maximal simultaneous flow of goods simultaneously the minimum cost. Journal of Science & Technology, Da Nang University, 5 (54) 2012.
- [7] Tran Quoc Chien: The algorithm finds the shortest path in the general graph, Journal of Science & Technology, University of Da Nang, 12 (61) / 2012, 16-21.
- [8] Tran Quoc Chien, Nguyen Mau Tue, Tran Ngoc Viet: The algorithm finds the shortest path on the extended graph. Proceeding of the 6th National Conference on Fundamental and Applied Information Technology (FAIR), Proceedings of the Sixth National Conference on Scientific Research and Application, Hue, 20-21 June 2013. Publisher of Natural Science and Technology. Hanoi 2013. p.522-527.
- [9] Tran Quoc Chien: Applying the algorithm to find the fastest way to find the maximum linear and simultaneous minimum cost on an extended transportation network, Journal of Science & Technology, University of Da Nang . 10 (71) 2013, 85-91.

- [10] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue: Optimized Linear Multiplexing Algorithm on Expanded Transport Networks, Journal of Science & Technology, University of Da Nang. 3 (76) 2014, 121-124.
- [11] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue: The problem of linear multi-channel traffic flow in traffic network. Proceedings of the 7th National Conference on Fundamental and Applied Information Technology Research (FAIR'7), ISBN: 978-604-913-300-8, Proceedings of the 7th National Science Conference "Fundamental and Applied Research IT ", Thai Nguyen, 19-20 / 6/2014. Publisher of Natural Science and Technology. Hanoi 2014. p.31-39.
- [12] Tran Quoc Chien, Ho Van Hung: Extended linear multi-commodity multi-cost network and maximal flow finding problem. FAIR-2017.

## AUTHORS

Ass. Prof. DrSc. Tran Quoc Chien ([http://scv.ued.udn.vn/ly\\_lich/chi\\_tiet/275](http://scv.ued.udn.vn/ly_lich/chi_tiet/275)). He has 14 papers in SCIE of Journal (<http://www.kybernetika.cz/contact.html>). Born in 1953 in Dien Ban, Quang Nam, Vietnam. He graduated from Maths\_IT faculty. He got Ph.D Degree of maths in 1985 in Charles university of Prague, Czech Republic and hold Doctor of Science in Charles university of Prague, Czech Republic in 1991. He received the title of Ass. Pro in 1992. He work for university of Danang, Vietnam. His main major: Maths and computing, applicable mathematics in transport, maximum flow, parallel and distributed process, discrete mathematics, graph theory, grid Computing, distributed programming.



M.Si. Ho Van Hung (<http://qnamuni.edu.vn/viewLLKH.asp?MaGV=134>). Born in 1977 in Thang Binh, Quang Nam, Vietnam. He graduated from Faculty of Information Technology – College of Sciences – Hue University in 2000. He got master of science (IT) at Danang university of technology. His main major: Applicable mathematics in transport, maximum flow, parallel and distributed process, graph theory and distributed programming.

