EXTENDED LINEAR MULTICOMMODITY MULTICOST NETWORK ANDMAXIMAL CONCURENT FLOW PROBLEMS

Ho Van Hung[1] and Tran Quoc Chien[2]

[1] Quangnam University, Tamky, Vietnam
[2]The University of Education, University of Danang, Danang, Vietnam

ABSTRACT

Graph is a powerful mathematical tool applied in many fields as transportation, communication, informatics, economy, ... In ordinary graph the weights of edges and vertexes are considered indepently 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. In the article [2], a kind of weights, called switch cost, is defined. The papers [3-6] study multicomodity flow problems in ordinary networks. The papers [3-6] study multicomodity flow problems in extended networks, where switch costs are defined for mixed graphs. The papers [12,13] develops a model of extended linear multicommodity multicost network and studies the maximal linear multicomodity multicost flow problems. The papers [14,15] study the maximal multicomodity multicost flow limited cost problems. Model of extended linear multicommodity multicost network can be applied to modelize many practical problems more exactly and effectively. The presented paper studies the maximal concurent linear multicomodity multicost flow problems, that are modelized as implicit linear programming problems. On the base of dual theory in linear programming an effective aproximate algorithms is developed.

KEYWORDS

Graph, Network, Multicommodity Multicost Flow, Optimization, Linear Programming.

1. INTRODUCTION

Network and its flow is a powerful mathematical tool applied in many fields such 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 [3-6]. Multicommodity singlecost flow flows problems in extended transport networks are studied in the papers [7-11]. The papers [12,13] study maximal multicommodity multicost flow problems. The papers [14,15] study maximal multicomodity multicost flow limited cost problems.

DOI : 10.5121/ijmnct.2019.9101

The paper studies maximal concurent linear multicommodity multicost flow problems which are modeled by implicit linear programming problems. On the base of dual theory in linear programming, an effective approximate algorithm is developed.

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 commodities 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 commodities on the same edge, but reverse directions, share the capacity of the edge.

The symbol r is the number of commodity types, $q_i > 0$ is the coefficient of conversion of commodity type$i$, $i = 1..r$.

Given the following functions:

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

Edge service coefficient function $ze:E \to R^*$, *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$cv:V \to R^*$, where $cv(u)$ is the passing capability of the node $u \in V$.

Node service coefficient function $zv:V \to R^*$, 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, $i=1..r$, $be_i:E \to R^*$, 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 direction may vary.

Node switch cost function i, $i=1..r$, $bv_i:V \times E_v \times E_v \to R^*$, *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 multicommodity multicost network.*

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

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

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:

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

3. Multicommodity Flows In Extended Linear Multicommodity Multicost Network

Given a multicommodity multicost network G = ($V,E,ce, ze, cv, zv, \{be_i, bv_i, q_i | i=1..r\}$). Assume, for each commodity 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 comodity of type i, that is necessary to move from source node $s_{i,j}$ to target node $t_{i,j}$.

Denote $P_{i,j}$ 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

$$P_i = \bigcup_{j=1}^{k_i} P_{i,j}$$

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.

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

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

is called a *multicommodity flow* on the linear extended multicommodity multicost network, if it satisfies the following *edge and node capacity* constraints:

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

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

The expressions

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

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

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

are called *the flow value of commodity type i of F*.

The expressions

$$fv = \sum_{i=1}^{r} fv_i$$

is called *the flow value of F*.

4. MAXIMAL CONCURENT MULTICOMMODITYMULTICOST FLOW PROBLEMS

Given an extended linear multicommodity multicost network G=(V,E, ce, ze, cv, zv, $\{be_i$, bv_i, $q_i | i$=1..$r\}$). Assume, for each commodity 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 comodity of type i, that is necessary to move from source node $s_{i,j}$ to target node $t_{i,j}$

Each type of commodities i, i=1..r, required tomove D_{ij}units of commodity type i from source node $s_{i,j}$ to target node $t_{i,j}$, $\forall j = 1, ..., k_i$.

The task of the problem is to find a maximum number such that there exists a transfer flow $\lambda.D_{i,j}$unit of commodity type i, i=1..r, from source node $s_{i,j}$ to target node $t_{i,j}$,$\forall j$ = 1..k_i. Set

$$d_{i,j} = q_i.D_{i,j},\ \forall i=1..r,\ \forall j=1..k_i$$

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

$$\lambda \to max$$

satisfies

$$\left.\begin{array}{ll}
\sum_{i=1}^{r}\sum_{j=1}^{k_i}\sum_{p\in P_{i,e}} x_{i,j}(p) & \le ce(e).ze(e),\ \forall e\in E \\
\sum_{i=1}^{r}\sum_{j=1}^{k_i}\sum_{p\in P_{i,v}} x_{i,j}(p) & \le cv(v).zv(v),\ \forall v\in V \\
\sum_{p\in P_{i,j}} x_{i,j}(p) & \ge \lambda.d_{i,j},\ \forall i=1..r,\ \forall j=1..k_i \\
\lambda \ge 0,\ x_{i,j}(p) \ge 0,\ \forall i=1..r,\ \forall j=1..k_i,\ \forall p\in P_{i,j} &
\end{array}\right\} \quad (P)$$

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 $lv(v)$, each requirement$d_{ij}$is assigned variablez_{ij}, $\forall i=1..r$, $\forall j=1..k_i$. The problem (D) states the following:

$$\left.\begin{array}{l}
D(le,lv) = \sum_{e\in E} ce(e).ze(e).le(e) + \sum_{v\in V} cv(v).zv(v).lv(v) \to min \\
\sum_{e\in p} le(e) + \sum_{v\in p} lv(v) \ge z_{i,j},\ \ \forall i=1..r,\ \forall j=1..k_i,\ \forall p\in P_{i,j} \\
\sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.z_{i,j} \ge 1 \\
le(e) \ge 0, \forall e\in E,\ lv(v) \ge 0, \forall v\in V, \\
z_{i,j} \ge 0,\ \forall i=1..r,\ \forall j=1..k_i
\end{array}\right\} \quad (D)$$

Now, given $p \in P$ a path from node u to node v through edges e_i, i=1..(h+1), and nodes u_i, i=1..h, as follows

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

We define the path length of p, denoted by $length(p)$, depending on the variables $le(e)$, $lv(v)$ by the following formula:

$$length(p) = \sum_{j=1}^{h+1} le(e_j) + \sum_{j=1}^{h} lv(u_j)$$

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

$$\alpha(le,lv) = \sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.dist_{i,j}(le,lv).$$

Consider the problem (D_α):

$$\beta = \min\left\{ \frac{D(le,lv)}{\alpha(le,lv)} \middle| le: E \to R^*, lv: V \to R^* \right\}$$

4.1 Lemma

The problem (D) is equivalent to the problem (D_α) 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.

Prove

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

$$le'(e) = le(e)/\alpha(le,lv), \forall e \in E,\ lv'(v) = lv(v)/\alpha(le,lv), \forall v \in V,$$
$$z'_{i,j} = dist_{i,j}(le',lv') = dist_{i,j}(le,lv)/\alpha(le,lv),\ \forall i=1..r,\ \forall j=1..k_i.$$

We have

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

and

$$\sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.z'_{i,j} = \frac{1}{\alpha(le,lv)} \sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.dist_{i,j}(le,lv) = 1$$

So (le',lv') is an accepted solution of (D) and $D(le',lv') = \frac{D(le,lv)}{\alpha(le,lv)}$. Hence, $min(D) \leq min(D_\alpha)$ (*).

In contrast, let $(le,lv, z_{i,j})$ be an accepted solution of (D). Then, we have:

$$z_{i,j} \leq dist_{i,j}(le,lv),\ \forall i=1..r, \forall j=1..k_i \Rightarrow \alpha(le,lv) = \sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.dist_{i,j}(le,lv) \geq \sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}.z_{i,j} \geq 1.$$

It follows $\frac{D(le,lv)}{\alpha(le,lv)} \leq D(le,lv)$.Hence,$min(D) \geq min(D_\alpha)$ (**).

From (*) and (**)it follows $min(D) = min(D_\alpha)$.

Next, if (le,lv) is an optimal solution of the problem (D_α), then ($le',lv', z_{i,j}$)where

$$le'(e) = le(e)/\alpha(le,lv), \forall e\in E; lv'(v) = lv(v)/\alpha(le,lv), \forall v\in V;$$
$$z'_{i,j} = dist_{i,j}(le',lv') = dist_{i,j}(le,lv) / \alpha(le,lv),\ \forall i=1..r,\ \forall j=1..k_i.$$

is an optimal solution of problem (D).

Conversely, if ($le,lv, z_{i,j}$) is an optimal solution of the problem (D), then (le,lv) is an optimal solution of the problem (D_α).

5. ALGORITHM

Ideas

Algorithmisimplemented through several phases. Each phase consists of k loops. At the loop $[i,j]$, $i=1..r$, $j=1..k_i$of a period$t$we moved$d_{i,j}$converted units of commodity type $i$from source node $s_{i,j}$ to target node $t_{i,j}$. This move is implemented in a number of steps.

Algorithm

Input: Extended multicommodity multicost 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 comodity $D_{ij}$of type i, that is necessary to move from source node $s_{i,j}$ to target node $t_{i,j}$.

Denote $n=|V|$, $m=|E|$ and ω is the approximation ratio to be achieved.

Output : Maximal flow F represents a set of converted flows at the edges
$F = \{f_{i,j}(e) \mid e \in E, i=1..r, j=1..k_i \}$
Total cost B_f. Maximal coefficient λ.

Procedure
//Initialization: Calculate ε and δ

$$\varepsilon = 1 - \sqrt[3]{\frac{1}{1+\omega}}\ ; \delta = \left(\frac{m+n}{1-\varepsilon}\right)^{-\frac{1}{\varepsilon}}\ ;$$

$d_{i,j} = D_{i,j}.q_i$, $\forall i=1..r, j=1..k_i$
$le(e) = \delta/(ce(e)ze(e))$; $x_{i,j}(e) = 0$; $\forall e \in E$,
$lv(v) = \delta/(cv(v)zv(v))$; $\forall v\in V$.

$$D(le,lv) = \sum_{e\in E} ce(e).ze(e).le(e) + \sum_{v\in V} cv(v).zv(v).lv(v) = (m+n)\delta;$$

$t = 0$;// phase counts in iteration……while ($D(t) < 1$)

//Denote le_i, lv_i, $D(t)$, $\alpha(t)$ the corresponding quantities after step t.

$D(0) = (m+n)\delta$; $le_0(e) = le(e)$; $\forall e \in E$,
$lv_0(v) = lv(v)$; $\forall v\in V$, $B_f= 0$;
do // phases
{
for (i=1; i<=r; i++)
for (j=1; j<=k_i; j++) // loops

```
{
d'i,j = di,j ;
do // steps
{
```

Find the shortest path p từ $s_{i,j}$ đến $t_{i,j}$ calculated by function *length*(.). Note that the path p must be valid for commodity of type i, i.e., not containing the edge with edge cost ∞ or the node with the switch cost ∞.$dist_{i,j}(le,lv)$ is the shortest path$p$ từ $s_{i,j}$ đến $t_{i,j}$ calculated by function *length*(p), $be_i(p)$ is the cost of a converted unit of commodity type $i$on the path$p$.

```
Set c = min{min{ce(e).ze(e)|e∈p},min{cv(v).zv(v)|v∈p}, d'i,j}
//Flow adjustments:
∀e∈p, xi,j(e) = xi,j(e) + c;  le(e) = le(e).(1+ε.c/(ce(e).ze(e)));
        ∀v∈p, lv(v)= lv(v).(1+ε.c/(cv(v).zv(v)));
        d'i,j = d'i,j–c;  Bf = Bf + c.bei(p);
        D(le,lv) = D(le,lv) + ε.c.disti,j(le,lv);
        } while (d'i,j> 0);
        }// for … for …
t = t + 1;
D(t) = D(le,lv);
let(e) = le(e) ; ∀e ∈ E,
lvt(v) = lv(v) ; ∀v ∈ V.
if (D(t)< 1)
for (i=1; i<= r; i++)
for (j=1; j<= ki; j++)
for (e∈E)
fi,j(e) = xi,j(e);                // fi,j(e) denote optimal flow
} while (D(t) < 1)
//Modifying the resulting flows F
Bf = Bf / log_{1+ε} (1/δ);
for (i=1; i<= r; i++)
for (j=1; j<= ki; j++)
    for (e∈E)
    fi,j(e) = fi,j(e) / log_{1+ε} (1/δ);

//Modifying flows on scalar edge
for (i=1 ; i<=r ;i++)
for (j=1 ; j<=ki ;j++)
fore∈E, e scalar
    if fi,j(e)>=fi,j(e')          // e' is the opposite of the direction e
    {
        Bf = Bf–fi,j(e')(bei(e)+bei(e'));
        fi,j(e)=fi,j(e) – fi,j(e') ;
        fi,j(e')=0 ;
    }
else
    {
    Bf = Bf–fi,j(e)(bei(e)+bei(e'));
        fi,j(e')=fi,j(e') – fi,j(e) ;
        fi,j(e)=0 ;
```

}

//Convert the flow $f_{i,j}(e)$ to the actual flow $rf_{i,j}(e)$ by dividing the conversion flow by the conversion coefficient

$$rf_{i,j}(e) = f_{i,j}(e)/q_i\ ,\ \forall e\in E,\ i=1..r,\ j=1..k_i$$

// Maximum approximation coefficient

$\lambda = (t-1) / \log_{1+\varepsilon}\frac{1}{\delta}$;

//The End.

Proof Of Algorithm

Remarks: In $(t-1)$ phases of implementation of the above algorithm, $\forall i=1..r$, $j=1..k_i$, we have transferred $(t-1).d_{i,j}$units of converting from $s_{i,j}$ to $t_{i,j}$. However, the transferred flow may exceed the throughout capacity of the edges.

The following lemma resolves the above problem

5.1 Lemma

$$\lambda > \frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}.$$

Proof. Consider any edge *e*. Initiationly,

$$le(e)=\delta/(ce(e)ze(e)),\ \forall e \in E, lv(v)=\delta/(cv(v)zv(v)),\ \forall v\in V.$$

After $(t-1)$ phases implemented, we have$D(t-1) < 1$, it means

$$\sum_{e\in E} ce(e).ze(e).le_{t-1}(e) + \sum_{v\in V} cv(v).zv(v).lv_{t-1}(v) < 1,$$

that follows

$$le_{t-1}(e) < 1/(c(e).ze(e)),\ \forall e\in E, lv_{t-1}(v) < 1/(cv(v)zv(v)),\ \forall v\in V.$$

Let$fe(e)$ be the sum of the converted units of commodities passing$e\in E$ and $fv(v)$is the sum of the converted units of commodities passing $v\in V$ in$(t-1)$ phases.

Consider $e\in E$. Suppose in the process of constructing fe(e) there is$ce(e)ze(e)$ units of the flow are passed *e* through *q* steps, each step transfers $g_s$converted units of commodity,

i.e. $\sum_s g_s = ce(e)ze(e)$. Through each step,le(e) is increased by the factor $(1+\varepsilon.g_s/(ce(e)ze(e)))$. So, through *q* steps, le(e) is increased by the factor $\prod_s\left(1+\varepsilon.g_s/(ce(e)ze(e))\right) >$ $(1+\varepsilon.\sum_s g_s/(ce(e)ze(e))) = (1+\varepsilon)$.

We see, for every edge$e\in E$, for each transfer of $ce(e)ze(e)$ converted units of commodities through *e*, $le(e)$increases by at least one factor $(1+\varepsilon)$.

Similarly, for every node $v\in V$, for every $cv(v)zv(v)$ converted units of commodity passed v,$lv(v)$ increases by at least one factor $(1+\varepsilon)$.

On the other hand, the number of times to send $ce(e).ze(e)$ converted unit of commodity over each edge $e\in E$is at least $fe(e)/(ce(e).ze(e))$ and the number of times to send $cv(v)zv(v)$ converted unit of commodity through each node$v\in V$is at least $fv(v)/(cv(v)zv(v))$.

At this point, the edge and node functions will satisfy the following inequality:

$le_{t-1}(e) \geq le_0(e).(1+\varepsilon)^{fe(e)/(ce(e).ze(e))}, \forall e\in E$
and
$lv_{t-1}(v) \geq lv_0(v).(1+\varepsilon)^{fv(v)/(ve(v).zv(v))}, \forall v\in V$

Hence inferred

$$fe(e)\leq ce(e).ze(e).\ \log_{1+\varepsilon}\frac{le_{t-1}(e)}{le_0(e)} < ce(e).ze(e).\ \log_{1+\varepsilon}\frac{1/(ce(e),ze(e))}{\delta/(ce(e).ze(e))}$$

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

and

$$fv(v)\ \leq cv(v).zv(v).\ \log_{1+\varepsilon}\frac{lv_{t-1}(v)}{lv_0(v)} < cv(v).zv(v).\ \log_{1+\varepsilon}\frac{1/(cv(v)zv(v))}{\delta/(cv(v)zv(v))}$$

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

Thus, divide $fe(e)$ by $\log_{1+\varepsilon}\frac{1}{\delta}, \forall e\in E$, that follows$fv(v)$ is divided by $\log_{1+\varepsilon}\frac{1}{\delta}$, we receive accepted flows

The above analysis shows that after $(t-1)$ of phases, $\forall i=1..r$, $j=1..k_i$, we have moved $(t-1).d_{i,j}$converted units from $s_{i,j}$to $t_{i,j}$.However, in order for the flow to be accepted, we must divide the flow by $\log_{1+\varepsilon}\frac{1}{\delta}$. So, $\forall i=1..r$, $j=1..k_i$, we move $\frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}} d_{i,j}$converted units of commodities from $s_{i,j}$to $t_{i,j}$.. Then, flow through edge$e$ is no greater than $ce(e).ze(e)$, $\forall e\in E$, and flow through v is not greater than $cv(v).zv(v)$, $\forall v\in V$. So, we have

$$\lambda > \frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}.$$

5.2 Lemma

Assume$\beta\geq 1$. The algorithm's found flow, after being divided by $\log_{1+\varepsilon}\frac{1}{\delta}$, is the concurent maximal flow, where $\frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}$ is the maximal coefficient with the approximation ratio $(1+\omega)$.

Proof

Since le and lv functions are incremental after each update,

$$D(q) \le D(q-1) + \varepsilon.\alpha(q),\ \forall q = 1..t. \qquad (2)$$

Next, we have

$$\frac{D(q)}{\alpha(q)} \ge \beta \Rightarrow \alpha(q) \le \frac{D(q)}{\beta},\ \forall q = 1..t. \qquad (3)$$

From (2) and (3) we receive:

$$D(q) \le D(q-1) + \varepsilon.D(q)/\beta,\ \forall q = 1..t,$$

$$\Rightarrow D(q) \le \frac{D(q-1)}{1-\varepsilon/\beta} \le \frac{D(q-2)}{(1-\varepsilon/\beta)^2} \le \ldots \le \frac{D(0)}{(1-\varepsilon/\beta)^q} = \frac{(m+n).\delta}{(1-\varepsilon/\beta)^q},\ \forall q = 1..t.$$

For $\beta \ge 1$, we have

$$D(q) \le \frac{(m+n).\delta}{(1-\varepsilon/\beta)^q} = \frac{(m+n).\delta}{(1-\varepsilon/\beta)} \cdot \left(1 + \frac{\varepsilon}{\beta-\varepsilon}\right)^{q-1}$$

$$\le \frac{(m+n).\delta}{(1-\varepsilon/\beta)}\ e^{\frac{\varepsilon(q-1)}{\beta-\varepsilon}} \le \frac{(m+n).\delta}{(1-\varepsilon)}\ e^{\frac{\varepsilon(q-1)}{\beta-\varepsilon}},\ \forall q = 1..t.$$

With regard tostop condition of algorithm $D(t) \ge 1$, we have

$$1 \le D(t) \le \frac{(m+n).\delta}{(1-\varepsilon)}\ e^{\frac{\varepsilon(t-1)}{\beta-\varepsilon}}$$

Hence

$$\frac{\beta}{t-1} \le \frac{\varepsilon}{(1-\varepsilon)\ln\frac{1-\varepsilon}{(m+n)\delta}}$$

(4)

Set $\gamma = \frac{\beta}{t-1}\log_{1+\varepsilon}\frac{1}{\delta}$ and entail $\frac{\beta}{t-1}$ từ (4),we have

$$\gamma < \frac{\varepsilon.\log_{1+\varepsilon}\frac{1}{\delta}}{(1-\varepsilon)\ln\frac{1-\varepsilon}{(m+n).\delta}} = \frac{\varepsilon}{(1-\varepsilon)\ln(1+\varepsilon)}\ \frac{\ln\frac{1}{\delta}}{\ln\frac{1-\varepsilon}{(m+n).\delta}}$$

For $\delta = \left(\frac{m+n}{1-\varepsilon}\right)^{-\frac{1}{\varepsilon}}$, we have $\frac{\ln\frac{1}{\delta}}{\ln\frac{1-\varepsilon}{(m+n).\delta}} = (1-\varepsilon)^{-1}$, and so

$$\gamma < \frac{\varepsilon}{(1-\varepsilon)^2\ln(1+\varepsilon)} \le \frac{\varepsilon}{(1-\varepsilon)^2(\varepsilon-\varepsilon^2/2)} \le (1-\varepsilon)^{-3}$$

On the other hand, by duality, we have $\gamma \ge 1$. Vì $\varepsilon = 1-\sqrt[3]{\frac{1}{1+\omega}}$, we have $\gamma < (1+\omega)$. Then, the value $\lambda = \frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}$ is the maximal coefficient with the aproximation ratio $(1+\omega)$.

Lemma 5.3

Assume β<1. Let l>1 such that $l.\beta$> 1. Apply the algorithm to the requirements

$$d'_{i,j} = \frac{1}{l} d_{i,j},\ \forall i=1..r,\ j=1..k_i.$$

The algorithm's found flow, after divide $\log_{1+\varepsilon}\frac{1}{\delta}$, is the maximal concurent flow of the original problem, wherethe maximal coefficient is $\frac{1}{l}\frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}$with the approximation ratio (1+ω).

Proof

According to Lemma 5.2, the algorithm's found flow, after dividing by $\log_{1+\varepsilon}\frac{1}{\delta}$, is the maximal concurent flow of the problem to the requirement $d'_{i,j} = \frac{1}{l} d_{i,j}$, $\forall i=1..r$, $j=1..k_i$. and $\lambda = \frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}$ is the maximal coefficient with the aproximation ratio (1+ω). Hence, $\frac{1}{l}\lambda = \frac{1}{l}\frac{t-1}{\log_{1+\varepsilon}\frac{1}{\delta}}$ is the maximal coefficient with the approximation ratio (1+ω) of the original problem.

6. ALGORITHM COMPLEXITY

6.1 Theorem

The algorithm's complexity isO(ω^{-2}.($cemax/dmax$).(χ+k).$m.n^3$.$\log_2(m+n)$), where $m$is the number of edges and$n$ is the number of vertices of the network, $k= k_1+\ldots+k_r$,$cemax$ = max{$ce(e).ze(e)$ | $e\in E$ }, $dmax$ = max{$d_{i,j}$| i=1..r, j=1..k_i},andχ = $\sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j}$ /$cmin$, with$cmin$=min{$cemin$, $cvmin$}, $cemin$=min{$ce(e).ze(e)$ | $e\in E$ } and$cvmin$=min{$cv(v).zv(v)$ | $v\in V$ }

Proof

First, we find the number of phases the algorithm has taken. According to the proof of lemma 5.2 above and for$\beta = \lambda$, we have

$$1 \le \gamma = \frac{\beta}{t-1}\log_{1+\varepsilon}\frac{1}{\delta} \Rightarrow t \le 1+\beta.\log_{1+\varepsilon}\frac{1}{\delta} \Rightarrow t = \log_{1+\varepsilon}\frac{1}{\delta}.\mathrm{O}(\beta),$$

whereε and δdepend on ω.Besides, t depends on β.

Further, denote $imax$, $jmax$ indexes satisfying

$$dmax = \max\{d_{i,j} \mid i=1..r, j=1..k_i\} = d_{imax,jmax}.$$

From the constraint of the problem (P)

$$\sum_{p\in P_{i,j}} x_{i,j}(p) \geq \lambda.d_{i,j},\ \forall i=1..r,\ \forall j = 1..k_i$$

we have

$$\lambda \leq \sum_{p\in P_{i\max,j\max}} x_{i\max,j\max}(p) / d_{imax,jmax} \leq \sum_{e\in E_{s_{i\max,j\max}}} c(e).z(e) / dmax$$

$$\leq \left|E_{s_{i\max,j\max}}\right|.cemax / dmax \leq m.cemax / dmax$$

that implies

$$\beta = \lambda \leq m.cemax/dmax \Rightarrow t = \log_{1+\varepsilon}\frac{1}{\delta}.O(m.cemax/dmax),$$

Replacing $\delta = \left(\frac{m+n}{1-\varepsilon}\right)^{-\frac{1}{\varepsilon}}$ to the above expresion, we have

$$t = \frac{1}{\varepsilon}\log_{1+\varepsilon}\frac{m+n}{1-\varepsilon}.O(m.cemax/dmax).$$

On the other hand, each phase implements k loop, so the loop number is $k.t$. Consider the loop transfering$d_{i,j}$the converted units of commodities from $s_{i,j}$to $t_{i,j}$, i=1..r, j=1..k_i. Since $cmin$ is the the minimal capacity of edges and nodes, the number of steps required to execute the loop is not exceeded ($d_{i,j}/cmin$+1). The main procedure in each step, finding the shortest path from $s_{i,j}$to $t_{i,j}$, has a complexity of n^3 [8]. So the complexity of the loop is ($d_{i,j}$ / $cmin$+1).n^3. Hence the complexity of each loop is:

$$\sum_{i=1}^{r}\sum_{j=1}^{k_i}\left(d_{i,j}/c\min+1\right)n^3 = (\sum_{i=1}^{r}\sum_{j=1}^{k_i} d_{i,j} /cmin + k).n^3 = (\chi+k).n^3.$$

So the algorithm's complexity is

$$t.(\chi+k).n^3 = \frac{1}{\varepsilon}\log_{1+\varepsilon}\frac{m+n}{1-\varepsilon}.O(m.cemax/dmax).(\chi+k).n^3$$

$$= O(\frac{1}{\varepsilon}\log_{1+\varepsilon}\frac{m+n}{1-\varepsilon}.m.(cemax/dmax).(\chi+k).n^3)$$

$$= O(\omega^{-2}.(cemax/dmax).(\chi+k).m.n^3.\log_2(m+n)),$$

for $\varepsilon = 1 - \sqrt[3]{\frac{1}{1+\omega}} = O(\omega)$ và $\log_2(1+\varepsilon)\approx\varepsilon$.

7. CONCLUSIONS

The paper develops a model of extended linear multicommodity multicost network that can be more exactly and effectively applied to model many practical problems. Then, maximal concurent 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 further multicommodity multicost flow optimization problems.

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 multicommodity multicost network and maximal flow finding problem. Proceedings of the 7th National Conference on Fundamental and Applied Information Technology Research (FAIR'10), ISBN: 978-604-913-614-6, p.385-395. Publisher of Natural Science and Technology. Hanoi 2017.

[13] Tran Quoc Chien, Ho Van Hung: Applying algorithm finding shortest path in the multiple-weighted graphs to find maximal flow in extended linear multicomodity multicost network, EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 12.2017, Volume 4, Issue 11, pp 1-6.

[14] Tran Quoc Chien, Ho Van Hung: Extended Linear Multi-Commodity Multi-Cost Network and Maximal Flow Limited Cost Problems, The International Journal of Computer Networks & Communications (IJCNC), Volue 10, No. 1, January 2018, pp 79-93.

[15] Ho Van Hung, Tran Quoc Chien, Applying algorithm finding shortest path in the multiple- weighted graphs to find maximal flow limited cost in extended linear multicomodity multicost network, International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing (EECCMC), 2018. Accepted and to be published.

AUTHORS

Ass. Prof. DrSc. Tran Quoc Chien (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 tittle 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 mathemetics, graph theory, grid Computing, distributed program

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.