MAXIMAL CONCURRENT MINIMAL COST FLOW PROBLEMS ON EXTENDED MULTI-COST AND MULTI-COMMODITY NETWORKS

Ho Van Hung[1*], **Tran Quoc Chien**[2]
[1]*Quangnam University*
[2]*The University of Danang - University of Education*
[*]Corresponding author: hovanhung@qnamuni.edu.vn
(Received: October 21, 2020; Accepted: May 23, 2021)

Abstract - The graph is a great mathematical tool, which has been effectively applied to many fields such as economy, informatics, communication, transportation, etc. It can be seen that in an ordinary graph the weights of edges and vertexes are taken into account independently where the length of a path is the sum of weights of the edges and the vertexes on this path. Nevertheless, in many practical problems, weights at vertexes are not equal for all paths going through these vertexes, but are depending on coming and leaving edges. Moreover, on a network, the capacities of edges and vertexes are shared by many goods with different costs. Therefore, it is necessary to study networks with multiple weights. Models of extended multi-cost multi-commodity networks can be applied to modelize many practical problems more exactly and effectively. The presented article studies the maximal concurrent minimal cost flow problems on multi-cost and multi-commodity networks, which are modelized as optimization problems. On the base of the algorithm to find maximal concurrent flow and the algorithm to find maximal concurrent limited cost flow, an effective polynomial approximate procedure is developed to find a good solution.

Key words - Network; Graph; Multi-cost Multi-commodity Flow; Linear Optimization; Approximation.

1. Introduction

Network and its flow is a excellent mathematical tool applied in many practical problems, but up to now, most of the applications in traditional network have only considered the weights of edges and nodes which are taken into account independently where the path length is the sum of weights of the edges and the nodes on that path. However, there are many problems in practice, where the weight at a vertex is not equal for all paths passing through that vertex, but also depends on the incoming and outgoing edges of that vertex. For instance, the transit time on the transport network depends on the direction of transportation: going straight, turning left or turning right, and even some directions are forbidden. In order to solve the above problems, the article [1] introduces switching cost only for directed graphs. In addition, there are many types of goods on the network, with different costs for each type of goods. From that, the authors in the work [2] have given the idea of using the theory of duality in linear programming to solve these problems. Consequently, it is necessary to build a multi-commodity extended mixed network model to be able to apply the modeling of real problems more accurately and effectively. The articles [3-11] the authors have studied multi-commodity flows on ordinary networks. Besides, in articles [12-22] scientists have studied the problems of single-cost multi-commodity flow in logistics and transportation systems, economic and energy sectors, and communications and computer networks. The maximal multi-cost multi-commodity flow problems presented by the authors in the work [23-24]. In the articles [25-26] the authors have studied the maximal multi-cost multi-commodity flow limited cost problems. The maximal concurrent flow problems on extended multi-cost multi-commodity networks is presented in the works [27], [28], and in the works [29], [30] the authors have studied the maximal concurrent multi-commodity multi-cost flow problems.

This article studies maximal concurrent minimal cost, multi-cost and multi-commodity flow problems which are modeled as optimization problems. On the base of the algorithm to find the maximal concurrent flow and the algorithm to find the maximal concurrent limited cost flow, an effective polynomial approximate procedure is developed to find a good solution.

2. Multi-commodity flows in extended multi-cost multi-commodity network

Let $G = (V, E)$ be a mixed graph, where V is the node set and E is the edge set. The edges may be directed or undirected. For all nodes $u\in V$ we denote symbol E_u the set of edges incident node u. There are some kinds of goods circulating on the network. The nodes and the edges of the graph are shared by goods with 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.

Let r denote the number of commodities, $q_l > 0$ is the coefficient of conversion of commodity type l, $l = 1..\ r$.

We define the following functions:

Edge circulating capacity function $cv{:}E\rightarrow R^*$, where $cv(v)$ is the circulating capability of the edge $v\in E$.

Edge service coefficient function $zv{:}E\rightarrow R^*$, *where* $zv(v)$ is the circulating ratio of the edge $v\in E$ (the real capacity of the edge v is $zv(v).cv(v)$).

Node circulating capability function $cu{:}V\rightarrow R^*$, where $cu(u)$ is the circulating capability of the node $u\in V$.

Node service coefficient function $zu{:}V\rightarrow R^*$, where $zu(u)$ is the circulating ratio of the node $u\in V$ (the real capacity of the node u is $zu(u).cu(u)$).

The tuple (V, E, cv, zv, cu, zu) is called an *extended network.*

Edge cost function of commodity kind l, l=1..r, $bv_l{:}E\rightarrow R^*$, where $bv_l(v)$ is the cost of circulating the edge

$v \in E$ a converted unit of commodity of kind l. Note that with undirected edges, the costs of each directions may vary.

Node switch cost function of commodity kind l, l=1..r, bu_l*:*$V \times E_u \times E_u \to R^*$*, where* $bu_l(u,v, v')$ is the cost of passing a converted unit of commodity of kind l from edge v through node u to edge v'.

The set $(V, E, cv, zv, cu, zu,\{bv_l,bu_l, q_l| l=1..r\})$ is called *the multi-cost multi-commodity extended network*.

Note: If $bv_l(v)=\infty$, goods of kind l is forbidden from passing on edge v. If $bu_l(u,v,v') = \infty$, goods of kind l is forbidden from edge v through vertex u to edge v'.

Let p be the path from vertex u to vertex n through edges v_j, $j=1..(h+1)$, and vertices u_j, $j=1..h$ as follows:

$$p = [u, v_1, u_1, v_2, u_2, \ldots, v_h, u_h, v_{h+1}, n] \quad (1)$$

The cost of transferring a converted unit of commodity of kind l, $l = 1..r$, on the path p, is denoted by the symbol $b_l(p)$, and calculated as following:

$$b_l(p) = \sum_{j=1}^{h+1} bv_l(v_j) + \sum_{j=1}^{h} bu_l(u_j, v_j, v_{j+1}) \quad (2)$$

Given a multi-cost multi-commodity extended network $(V, E, cv, zv, cu, zu,\{bv_l, bu_l, q_l| l=1..r\})$. Assume that for each goods of kind l, $l=1..r$, there are k_l source-target pairs $(s_{l,j}, t_{l,j})$, $j=1..k_l$, each pair assigned a quantity of goods of kind l, that is necessary to move from source node $s_{l,j}$ to destination node $t_{l,j}$.

Let $Q_{l,j}$ denote the set of paths from node $s_{l,j}$ to node $t_{l,j}$ in G, which goods of kind l can be circulated, $l=1..r$, $j=1..k_l$. Set

$$Q_l = \bigcup_{j=1}^{k_l} Q_{l,j}, \forall l = 1...r \quad (3)$$

For each path $p \in Q_{l,j}$, $l=1..r$, $j=1..k_l$, denote $x_{l,j}(p)$ the flow of converted commodity of kind l from the source node $s_{l,j}$ to the target node $t_{l,j}$ along the path p.

Let $Q_{l,v}$ denote the set of paths in Q_l passing through the edge v, $\forall v \in E$.

Let $Q_{l,u}$ denote the set of paths in Q_l passing through the vertex u, $\forall u \in V$.

A set $F = \{x_{l,j}(p) \mid p \in Q_{l,j}, l = 1..r, j = 1..k_l\}$ is called a *multi-commodity flow* on the multi-cost and multi-commodity extended network, if the following *node and edge capacity* constraints are satisfied:

The *edge capacity* constraints:

$$\sum_{l=1}^{r}\sum_{j=1}^{k_l}\sum_{p \in Q_{l,v}} x_{l,j}(p) \le cv(v).zv(v), \forall v \in E \quad (4)$$

and the *vertex capacity* constraints:

$$\sum_{l=1}^{r}\sum_{j=1}^{k_l}\sum_{p \in Q_{l,u}} x_{l,j}(p) \le cu(u).zu(u), \forall u \in V \quad (5)$$

The expressions

$$fv_{lj} = \sum_{p \in Q_{l,j}} x_{l,j}(p), l = 1...r, j = 1...k_l \quad (6)$$

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

The expressions

$$fv_l = \sum_{j=1}^{k_l} fv_{l,j}, l = 1...r \quad (7)$$

are called *the flow value of commodity type l of the multi-commodity flow F*.

The expressions

$$fv = \sum_{l=1}^{r} fv_l \quad (8)$$

is called *the flow value of the multi-commodity flow F*.

3. Maximal concurrent minimal cost, multi-cost and multi-commodity flow problems

Given a multi-cost multi-commodity extended network $G=(V, E, cv, zv, cu, zu, \{bv_l, bu_l, q_l|l=1..r\})$. Assume that for each goods kind l, $l=1..r$, there are k_l source-target pairs $(s_{l,j}, t_{l,j})$, $j=1..k_l$, each pair assigned a quantity $D_{l,j}$ of goods of type l, that is required to transferred from source node $s_{l,j}$ to target node $t_{l,j}$.

The mission of the problem is to find a maximal concurrent coefficient λ with approximation ratio ω such that there exists a flow converting λ.Dl,j unit of goods kind l, l=1..r, from source node $s_{l,j}$ to target node $t_{l,j}$,$\forall j = 1..k_l$, and the total cost is minimal.

Set

$$d_{l,j} = q_l.D_{l,j}, \forall l=1..r, \forall j=1..k_l \quad (9)$$

The problem is expressed by means of an optimization model (P) as follows:

$$(P)\quad \begin{cases} \lambda \to \max \\ \text{satisfies} \\ \sum_{l=1}^{r}\sum_{j=1}^{k_l}\sum_{p \in Q_{l,v}} x_{l,j}(p) \le cv(v).zv(v), \forall v \in E \\ \sum_{l=1}^{r}\sum_{j=1}^{k_l}\sum_{p \in Q_{l,u}} x_{l,j}(p) \le cu(u).zu(u), \forall u \in V \\ \sum_{p \in Q_{l,j}} x_{l,j}(p) \ge \lambda.d_{l,j}, \forall l = 1..r, \forall j = 1..k_l \\ x_{l,j}(p) \ge 0, \forall l=1..r, \forall j=1..k_l, \forall p \in Q_{l,j} \end{cases}$$

and the total cost

$$\sum_{l=1}^{r}\sum_{j=1}^{k_l}\sum_{p \in Q_{l,j}} x_{l,j}(p).b_l(p)$$

is reduced as much as possible.

4. Algorithm

◊***Input:*** Multi-cost multi-commodity extended network $G=(V, E, cv, zv, cu, zu, \{bv_l, bu_l, q_l|l=1..r\})$, $n=|V|$, $m=|E|$. Assume that for each goods of kind l, $l=1..r$, there are k_l source-target pairs $(s_{l,j}, t_{l,j})$, $j=1..k_l$, each pair assigned a quantity $D_{i,j}$ of goods of kind l, that is necessary to move from source node $s_{l,j}$ to target node $t_{l,j}$. Given ω be the required approximation ratio.

◊***Output***: Maximal concurrent flow F represents a set of converged flows at the edges

$F = \{x_{l,j}(v) \mid v\in E, l=1..r, j=1..k_l\}$

with minimal total cost B_f.

◊ ***Algorithm***

Phase 1:

Run program maximal concurrent flow [28] with approximation ratio ω to get the maximal concurrent ratio λ, the maximal concurrent flow F_0 and the total cost B_f.

Set: $\lambda_{max} = \lambda$;

$B_0 = B_f$.

Phase 2:

Run program maximal concurrent limited cost flow [30] with the limited cost B_0 and the approximation ratio ω to get the maximal concurrent ratio λ_1, the maximal concurrent flow F_1 and the total cost B_1;

// B_1<= B_0 and λ_1<= λ_{max}

Phase 3:

```
i = 1;
while (λi>=λmax) do
{
    Run program maximal concurrent limited cost flow article [30] with the limited cost Bi and the approximation ratio ω to get the maximal concurrent ratio λi+1, the maximal concurrent flow Fi+1 and the total cost Bi+1;
    i=i+1;
}
k = i; B = Bk-1;
```

Phase 4:

```
while (λi<λmax) do
{
    Run program maximal concurrent limited cost flow [30] with the limited cost Bi=Bi-1*(λmax/λi) and approximation ratio ω to get the maximal concurrent ratio λi+1, the maximal concurrent flow Fi+1 and the total cost Bi+1;
    i=i+1;
}
```

Result: Maximal concurrent ratio: λ_{max}

Maximal flow : F_i

Minimal total cost : B_i

•**Theorem 1**. The algorithm gives maximal flow minimal cost with approximation ratio ω.

Proof

Obviously B_1<= B_0 and $\lambda_1 \le \lambda_{max}$.

The phase 3 ends after finite loops for the the costs are strictly descending

$B_0>B_1>B_2> \dots >B_i>B_{i+1}> \dots$

We prove that the phase 4 also ends after finite loops. Suppose the coefficients λ_i are rounded to p digits after the decimal point. We have

$B_i = B_{i-1}*(\lambda_{max}/\lambda_i)$, $\forall i\geq k$ and $\lambda_i<\lambda_{max}$

We note that from $\lambda_i<\lambda_{max}$ it follows $\lambda_i \leq\lambda_{max}-10^{-p}$ and $(\lambda_{max}/\lambda_i) \geq (\lambda_{max}/\lambda_{max}-10^{-q}) = q> 1$.

Finally we have

$B_i \geq B_{i-1}*q \geq B_{i-2}*q^2 \geq \dots \geq B_{i-(i-k+1)}*q^{(i-k+1)} = B*q^{(i-k+1)}$, $\forall i\geq k$ and $\lambda_i<\lambda_{max}$

Because $q^n\to\infty$ when $n\to\infty$, the phase 4 also ends after finite loops.

• Theorem 2.

The algorithm's complexity is

$O((t_1+t_2).\omega^{-2}.(cvmax/dmax).(\chi+k).m.n^3.\log_2(m+n+1))$,

where t_1 is the number of loops of the phase 3 and t_2 is the number of loops of the phase 4, m is the number of edges and n is the number of vertices of the network,

$k= k_1+\dots+k_r$, $cvmax = \max\{cv(v).zv(v) \mid v\in E\}$,

$dmax = \max\{d_{l,j} \mid l=1..r, j=1..k_l\}$,

and $\chi = \sum_{l=1}^{r}\sum_{j=1}^{k_l} d_{l,j} / cmin$

with $cmin=\min\{cvmin, cumin\}$,

$cvmin=\min\{cv(v).zv(v) \mid v\in E\}$

and

$cumin=\min\{cu(u).zu(u) \mid u\in V\}$.

Proof

It follows from the complexity of the algorithm finding maximal concurrent limited cost flow [29].

5. Test

Consider the example in the article [28]. Applying the above algorithm we get the following results.

• The results of running the program

Phase 1: Run the program to find maximal concurrent flow:

Table 1. *The results of running the program to find maximal concurrent flow*

Approximation ratio (ω)	Maximal concurrent ratio (λ)	Total Cost (B_f)
0.050	0.772	59392

Phase 2: Run program to find maximal concurrent limited cost flow with B=59392:

Table 2. *The results of running the program to find maximal concurrent limited cost flow with B=59392*

Limited cost (B)	Approximation ratio (ω)	Maximal concurrent ratio (λ)	Total cost (B_f)
59392	0.050	0.772	57582

The Maximal concurrent ratio $\lambda = 0.772 = \lambda_{max}$

Phase 3:

Run program to find maximal concurrent limited cost flow with B=57582:

Table 3. *The results of running the program to find maximal concurrent limited cost flow with B=57582*

Limited cost (B)	**Approximation ratio (ω)**	**Maximal concurrent ratio (λ)**	**Total cost (B_f)**
57582	0.050	0.770	56971

The Maximal concurrent ratio =0.770<λ_{max}. The phase 3 is ended and the phase 4 begins.

Pha se 4:

1st loop. Run program to find maximal concurrent limited cost flow with B=57731=(0.772/0.770)*57582:

Table 4. *The results of running the program to find maximal concurrent limited cost flow with B=57731*

Limited cost (B)	**Approximation ratio (ω)**	**Maximal concurrent ratio (λ)**	**Total cost (B_f)**
57731	0.050	0.771	57016

The Maximal concurrent ratio =0.771<λ_{max}. Next loop is executed.

2nd loop. Run program to find maximal concurrent limited cost flow with B=57805=(0.772/0.771)*57731:

Table 5. *The results of running the program to find maximal concurrent limited cost flow with B=57805*

Limited cost (B)	**Approximation ratio (ω)**	**Maximal concurrent ratio (λ)**	**Total cost (B_f)**
57805	0.050	0.771	57034

The Maximal concurrent ratio =0.771<λ_{max}. The next loop is executed.

3rd loop. Run program to find maximal concurrent limited cost flow with B=57880=(0.772/0.771)*57805:

Table 6. *The results of running the program to find maximal concurrent limited cost flow with B=57880*

Limited cost (B)	**Approximation ratio (ω)**	**Maximal concurrent ratio (λ)**	**Total Cost (B_f)**
57880	0.050	0.771	57043

The Maximal concurrent ratio =0.771<λ_{max}. Next loop is executed.

4th loop. Run program to find maximal concurrent limited cost flow with B=57955=(0.772/0.771)*57880:

Table 7. *The results of running the program to find maximal concurrent limited cost flow with B=57955*

Limited cost (B)	**Approximation ratio (ω)**	**Maximal concurrent ratio (λ)**	**Total cost (B_f)**
57955	0.050	0.771	57057

The Maximal concurrent ratio =0.772=λmax. The phase 4 is ended. The total cost is reduced from B_0 = 59392 to minimal cost 57057. Finally, we obtain the result as shown in the example:

Table 8. *The results of running the program to find maximal concurrent minimal cost*

Approximation ratio (ω)	**Maximal concurrent ratio (λ)**	**Minimal Cost (B_f)**
0.050	0.772	57057

The maximal concurrent flows:

**** Commodity type: 1***

Source: 1, Target: 4, C.flow: 154.324, R.flow: 154.324

Egde	*C.flow*	*R.flow*
(1, 2)	*154.324*	*154.324*
(2, 3)	*154.324*	*1543.24*
(3, 4	*1543.24*	*154.324*

Source: 1, Target: 5, C.flow:115.752, R.flow:115.752

Egde	*C.flow*	*R.flow*
(1, 2)	*114.886*	*114.886*
(2, 3)	*57.653*	*57.653*
(3, 4)	*11.934*	*11.934*
(4, 5)	*11.934*	*11.934*
(6, 5)	*103.808*	*103.808*
(7, 6)	*58.090*	*58.090*
(8, 7)	*0.856*	*0.856*
(3, 6)	*45.718*	*45.718*
(2, 7)	*57.234*	*57.234*
(1, 8)	*0.856*	*0.856*

Source: 1, Target: 9, C.flow: 231.486, R.flow: 231.486

Egde	*C.flow*	*R.flow*
(13, 9)	*231.486*	*231.486*
(1, 15)	*231.486*	*231.486*
(14,13)	*231.486*	*231.486*
(15,14)	*231.486*	*231.486*

**** Commodity type: 2***

Source: 12, Target: 4, C.flow: 192.905, R.flow: 38.581

Egde	*C.flow*	*R.flow*
(1, 2)	*0.289*	*0.058*
(2, 3)	*0.304*	*0.061*
(3, 4)	*0.304*	*0.061*
(9, 4)	*192.601*	*38.520*
(10, 9)	*192.601*	*38.520*
(11,10)	*192.601*	*35.520*
(11, 2)	*0.015*	*0.005*
(12,11)	*192.616*	*38.523*
(12, 1)	*0.289*	*0.058*

Source: 12, Target: 5, C.flow: 192.905, R.flow: 38.581

Egde	*C.flow*	*R.flow*
(1, 2)	*24.204*	*4.841*
(2, 3)	*39.450*	*7.890*
(3, 4)	*4.328*	*0.866*
(4, 5)	*63.183*	*12.637*
(6, 5)	*129.727*	*25.944*
(7, 6)	*94.600*	*18.920*
(8, 7)	*42.502*	*8.500*
(3, 6)	*35.122*	*7.024*
(2, 7)	*52.098*	*10.420*
(1, 8)	*42.502*	*8.500*
(9, 4)	*58.855*	*11.771*
(10, 9)	*58.855*	*11.771*
(11,10)	*58.855*	*11.771*
(11, 2)	*67.344*	*13.469*
(12,11)	*126.199*	*25.240*
(12, 1)	*66.706*	*13.341*

Source:12, Target: 9, C.flow: 96.453, R.flow: 19.290

Egde	*C.flow*	*R.flow*
(10, 9)	*0.395*	*0.079*

(13, 9)	*96.058*	*19.212*
(11,10)	*0.395*	*0.079*
(12,11)	*0.395*	*0.079*
(12,15)	*96.058*	*19.212*
(14,13)	*96.058*	*19.212*
(15,14)	*96.058*	*19.212*

**** Commodity type: 3***

Source: 12, Target: 13, C.flow: 192.905, R.flow:19.290

Egde	*C.flow*	*R.flow*
(7, 6)	*192.905*	*19.290*
(8, 7)	*192.905*	*19.290*
(6, 3)	*192.905*	*19.290*
(1, 8)	*192.905*	*19.290*
(3,10)	*192.905*	*19.290*
(10,13)	*192.905*	*19.290*
(12, 1)	*192.905*	*19.290*

Source:12, Target: 16, C.flow: 192.905, R.flow:19.290

Egde	*C.flow*	*R.flow*
(12,15)	*192.905*	*19.290*
(15,16)	*192.905*	*19.290*

Source: 13, Target: 16, C.flow: 192.905, R.flow:19.290

Egde	*C.flow*	*R. flow*
(6, 7)	*192.905*	*19.290*
(7, 8)	*192.905*	*19.290*
(8,16)	*192.905*	*19.290*
(3, 6)	*192.905*	*19.290*
(10, 3)	*192.905*	*19.290*
(13,10)	*192.905*	*19.290*

**** Commodity type: 4***

Source: 13, Target:16, C.flow: 154.324, R.flow: 7.716

Egde	*C.flow*	*R.flow*
(6, 7)	*154.324*	*7.716*
(7, 8)	*154.324*	*7.716*
(8,16)	*154.324*	*7.716*
(3, 6)	*154.324*	*7.716*
(10, 3)	*154.324*	*7.716*
(13,10)	*154.324*	*7.716*

• Analyzing the results

The final result when applying the above algorithm with the example in the article [28] is as follows:

- Approximation ratio (ω) : 0.050
- Maximal concurrent ratio (λ) : 0.772
- Minimal cost (B_f) : 57057.703

The maximal concurrent flows is as follows:

***Commodity type: 1** (Conversion factor of commodity $q = 1$)

Stroke ▬ ▬ illustration of the flow of commodity type 1.

For the source-target pair (1,4), the program will threading as shown in Figure 1 and the value of the stream as follows:

- Conversion flow value (C.flow) : 154.324
- Real flow value (R.flow) : 154.324

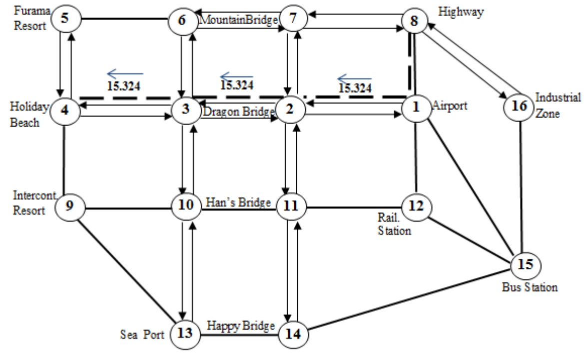

Figure 1. *Flow diagrams of commodity type 1 with target source pair (1,4)*

For the source-target pair (1,5), the program will threading as shown in Figure 2 and the value of the stream as follows:

- Conversion flow value (C.flow) : 115.752
- Real flow value (R.flow) : 115.752

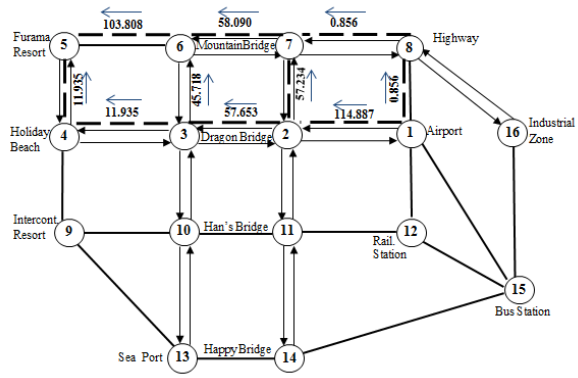

Figure 2. *Flow diagrams of commodity type 1 with target source pair (1,5)*

For the source-target pair (1,9), the program will threading as shown in Figure 3 and the value of the stream as follows:

- Conversion flow value (C.flow) : 231.486
- Real flow value (R.flow) : 231.486

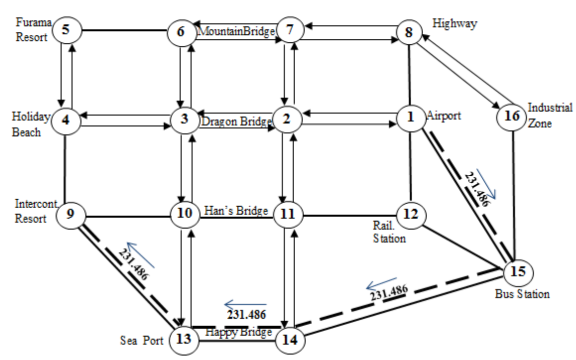

Figure 3. *Flow diagrams of commodity type 1 with target source pair (1,9)*

***Commodity type: 2** (Conversion factor of commodity $q = 5$)

Stroke ▪▪▪▪▪▪ illustration of the flow of commodity type 2.

For the source-target pair (12,4), the program will threading as shown in Figure 4 and the value of the stream as follows:

- Conversion flow value (C.flow) : 192.920
- Real flow value (R.flow) : 38.581

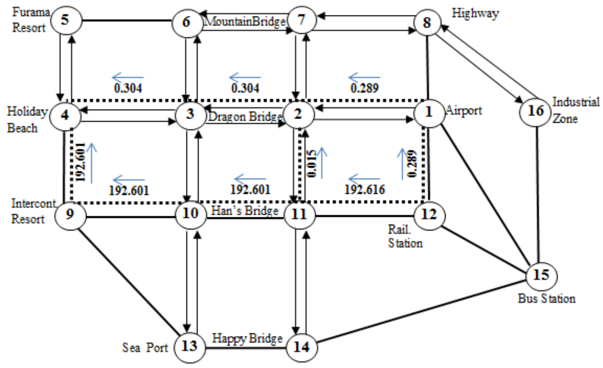

Figure 4. Flow diagrams of commodity type 2 with target source pair (12,4)

For the source-target pair (12,5), the program will threading as shown in Figure 5 and the value of the stream as follows:

- Conversion flow value (C.flow) : 192.905.
- Real flow value (R.flow) : 38.581

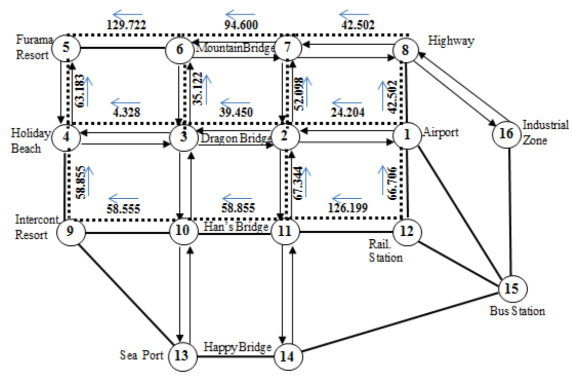

Figure 5. Flow diagrams of commodity type 2 with target source pair (12,5)

For the source-target pair (12,9), the program will threading as shown in Figure 6 and the value of the stream as follows:

- Conversion flow value (C.flow) : 96.453
- Real flow value (R.flow) : 19.290

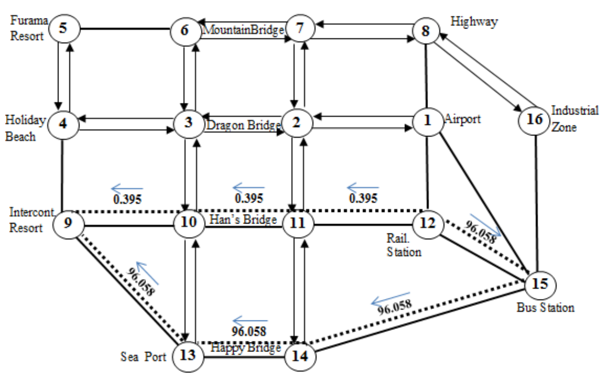

Figure 6. Flow diagrams of commodity type 2 with target source pair (12,9)

***Commodity type: 3** (Conversion factor of commodity $q = 10$)

Stroke — · · — illustration of the flow of commodity type 3.

For the source-target pair (12,13), the program will threading as shown in Figure 7 and the value of the stream as follows:

- Conversion flow value (C.flow) : 192.905
- Real flow value (R.flow) : 19.290

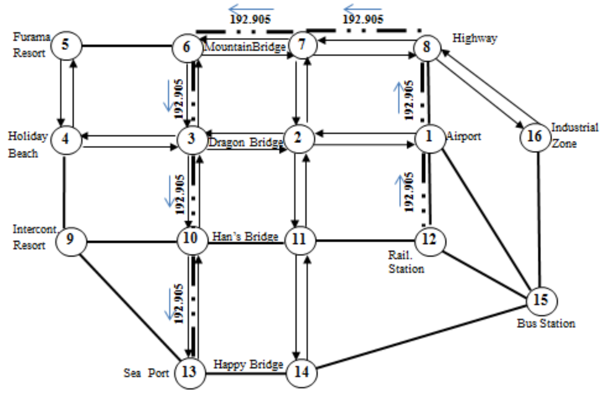

Figure 7. Flow diagrams of commodity type 3 with target source pair (12,13)

For the source-target pair (12,16), the program will threading as shown in Figure 8 and the value of the stream as follows:

- Conversion flow value (C.flow) : 192.905
- Real flow value (R.flow) : 19.290

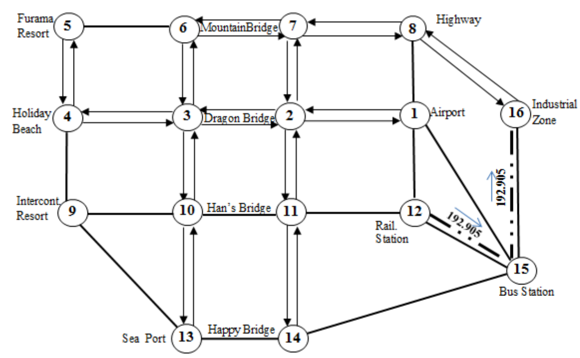

Figure 8. Flow diagrams of commodity type 3 with target source pair (12,16)

For the source-target pair (13,16), the program will threading as shown in Figure 9 and the value of the stream as follows:

- Conversion flow value (C.flow) : 192.905
- Real flow value (R.flow) : 19.290

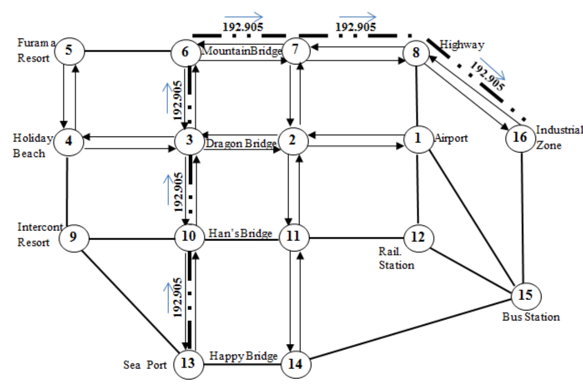

Figure 9. Flow diagrams of commodity type 3 with target source pair (13,16)

***Commodity type: 4** (Conversion factor of commodity $q = 20$)

Stroke – · – · – illustration of the flow of commodity type 3.

For the source-target pair (12,13), the program will threading as shown in Figure 10 and the value of the stream as follows

- Conversion flow value (C.flow) : 154.324
- Real flow value (R.flow) : 7.716

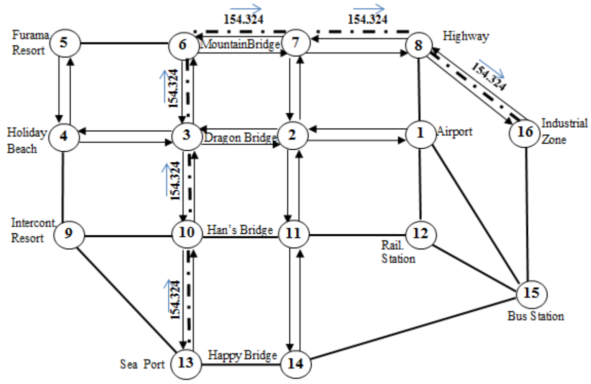

Figure 10. *Flow diagrams of commodity type 4 with target source pair (13,16)*

6. Conclusions

The article has studied the maximal concurrent minimal cost flow problems on multi-commodity and multi-cost extended networks, which can be applied to model many practical problems more accurately and efficiently. The maximal concurrent minimal cost flow problems are modeled as optimization problems. On the base of the algorithm to find the maximal concurrent flow in the article [27], [28] and the algorithm to find maximal concurrent limited cost flow in the article [29], [30] an effective polynomial approximate algorithm is developed to find a good solution. Correctness and complexity of the algorithm are proved. The algorithm is tested on an example and brings reliable results.

REFERENCES

[1] Xiaolong Ma and Jie Zhou, "An Extended Shortest Path Problem with Switch Cost Between Arcs", *IIMECS 2008*, 19-21 March, 2008, Hong Kong.

[2] Naveen Garg and Jochen Könemann, "Faster and Simpler Algorithms for Multi-Commodity Flow and Other Fractional Packing Problems", *SIAM Journal. Comput, Canada*, 37(2), 2007, pp. 630-652.

[3] Ellis L. Johnson, Geo L. Nemhauser; Joel S. Sokol, and Pamela H. Vance, "Shortest Paths and Multi-Commodity Network Flows", *A Thesis Presented to the Academic Faculty*, 2003, pp. 41-73.

[4] Xiaolong Ma and Jie Zhou, *"An extended shorted path problem with switch cost between arcs"*, IMECS 2008, Hong Kong.

[5] L. K. Fleischer, "Approximating fractional Multi - Commodity flow independent of the number of commodities", *SIAM J. Discrete Math.*, vol.13, no.4, 2000.

[6] G. Karakostas, "Faster approximation schemes for fractional Multi-Commodity flow problems", *In Proceedings, ACMSIAM Symposium on Discrete Algorithms*, vol.4, no.1, 2002.

[7] Aleksander, "*Faster Approximation Schemes for Fractional Multi-Commodity Flow Problems via Dynamic Graph Algorithms*" Massachusetts Institute of Technology, 2009.

[8] Tran Quoc Chien, "Linear multi-channel traffic network", *Ministry of Science and Technology*, code B2010DN-03-52.

[9] Tran Quoc Chien and Tran Thi My Dung, "Application of the shortest path finding algorithm to find the maximum flow of goods", *The University of Danang - Journal of Science and Technology*, 3 (44) 2011.

[10] Tran Quoc Chien, "Application of the shortest multi-path finding algorithm to find the maximum simultaneous flow of goods simultaneously", *The University of Danang - Journal of Science and Technology*, 4 (53) 2012.

[11] Tran Quoc Chien, "Application of the shortest multi-path finding algorithm to find the maximal simultaneous flow of goods simultaneously the minimum cost", *The University of Danang - Journal of Science and Technology*, 5 (54) 2012.

[12] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue, "Optimized Linear Multiplexing Algorithm on Expanded Transport Networks", *The University of Danang - Journal of Science and Technology*. 3 (76) 2014, pp.121-124.

[13] Tran Quoc Chien; Nguyen Mau Tue; and 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)*, Viet Nam, 2017. pp.522-527.

[14] Xiangming Yao, Baomin Han, Baomin Han, Hui Ren, *"Simulation-Based Dynamic Passenger Flow Assignment Modelling for a Schedule-Based Transit Network"*, Discrete Dynamics in Nature and Society- Hindawi, 2017.

[15] Samani A and Wang M, MaxStream: "SDN-based flow maximization for video streaming with QoSenhancement", *In: IEEE 43rd conference on local computer networks (LCN)*, 2018, pp. 287–290.

[16] Wright M, Gomes G, Horowitz R and Kurzhanskiy A, "On node models for highdimensional road networks", *Transp Res Part B: Methodol 105*, 2017, pp.212–234.

[17] Mohammadi M, Jula P and Tavakkoli Moghaddam R, "Design of a reliable multimodal multicommodity model for hazardous materials transportation under uncertainty". *Eur J Oper Res 257(3)*, 2017, pp.792–809.

[18] Xu X, Zhang Y and Lu J, "Routing optimization of small satellite networks based on multicommodity flow", *In: International conference on machine learning and intelligent communications. Springer, Cham*, 2017, pp.355–363.

[19] Fortz B, Gouveia L, Joyce-Moniz M, "Models for the piecewise linear unsplittable multicommodity fow problems", *Eur J Oper Res* 261(1), 2017, pp. 30–42

[20] Balma A, Salem S, Mrad M and Ladhari T, "Strong Multicommodity flow formulations for the asymmetric traveling salesman problem", *Eur J Oper Res 27*, 2018, pp. 72–79.

[21] Vahdani B, Veysmoradi D, Shekari N and Mousavi S, "Multi-objective, multi-period locationroutingmodel to distribute relief after earthquake by considering emergency roadway repair", *Neural ComputAppl 30(3)*, 2018, pp.835–854.

[22] Lu Y, Benlic U and Wu Q, "A population algorithm based on randomized tabu thresholding for themulticommodity pickup-and-delivery traveling salesman problem", *Comput Oper Res 101*, 2019, pp.285–297.

[23] Tran Quoc Chien, Ho Van Hung, "Extended linear Multi-Commodity multi-cost 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, pp.385-395.

[24] Tran Quoc Chien, Ho Van Hung, "Applying algorithm finding shortest path in the multiple-weighted graphs to find maximal flow in extended linear multi-comodity multi-cost network", *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 2017, Volume 4, Issue 11, pp.1-6.

[25] 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. (SCOPUS).

[26] Ho Van Hung, Tran Quoc Chien, "Implement and Test Algorithm finding Maximal Flow Limited Cost in extended multi-comodity multi-cost network", *The International Journal of Computer Techniques (IJCT)*, Volume 6 Issue 3, May – June 2019, pp.1-9.

[27] Ho Van Hung, Tran Quoc Chien, "Extended Linear Multi-Commodity Multi-Cost Network and Maximal Concurrent Flow Problems", *The International Journal of Mobile Netwwork Communications & Telematics (IJMNCT)*, Vol.9, No.1, February 2019, pp 1-14.

[28] Ho Van Hung, Tran Quoc Chien, "Installing Algorithm to find Maximal Concurrent Flow in Multi-cost Multi-commodity Extended", *International Journal of Innovative Science and Research Technology (IJISRT)*, Volue 4, Issue 12, December 2019, pp 1110-1119.

[29] Ho Van Hung, Tran Quoc Chien, Maximal Concurrent Limited Cost Flow Problems on Extended Linear Multi-Commodity Multi-Cost Networks, *American Journal of Applied Mathematics*, Vol. 8, No. 3, 2020, pp 74-82.

[30] Ho Van Hung, Tran Quoc Chien, Implement Algorithm Finding Maximal Concurrent Limited Cost Flow on Extended Multi-commodity Multi-cost Network, *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22.2 (2020), pp 34-44.