Universally Maximum Contraflow for Evacuation Planning

Ram Chandra Dhungana[1*], Shree Ram Khadka[2], Urmila Pyakurel[3] and Tanka Nath Dhamala[4]

Central Department of Mathematics, Tribhuvan University,
P.O.Box 13143, Kathmandu, Nepal
E-mails: [1]dhungana.ramchandra@gmail.com, [2] shreeramkhadka@gmail.com, [3] urmilapyakurel@gmail.com and [4]dhamala@yahoo.com

Abstract

Efficient evacuation transportation is important for the evacuation planning since one needs to evacuate as many evacuees as possible within the limited time. Contraflow problem, which increases the outbound capacity of the transportation network by reversing the direction, has been important for efficient evacuation. Various aspects of the contraflow problem i.e. the maximum contraflow, the maximum dynamic contraflow, the earliest arrival contraflow, the lexicographic contraflow have been studied in the literature. In this paper, we consider universally maximum contraflow problem which maximizes the amount of evacuees at every possible step of time with consideration of contraflow.

Keywords: Contraflow evacuation problem; Evacuation planning; Emergency mitigation

1. Introduction

Management of transportation routes for the effective and prompt evacuation has been a major problem during evacuation. Network flow models have been extensively applied for the study of finding efficient transportation routes during evacuation. However, there exist scattered models and solution methods based on the special features considered, Pardalos and Arulselvan [17], see also [1]. The time expanded graphs proposed by Ford and Fulkerson [6, 7] may yield theoretical solutions to most of the evacuation problems under the assumption that the parameters are kept constant. Similar extensions of time expanded graphs for flow-dependent travel times with some approximate algorithms for solving network flow problems exist in Kohler *et al*. [14]. The maximum flow evacuation problem, which sends maximum number of evacuees, with constant travel time has been solved with temporally repeated flow technique, Ford and Fulkerson [7]. Minieka [16] and Megiddo [15] introduce the concept of lexicographically maximal flow problem, which sends maximum number of evacuees in a given priority order, with pseudo-polynomial time solution procedure. It is noteworthy that Hoppe [11] and Hoppe and Tardos [12] present a polynomial time algorithm to solve the general case of the problem. Hamacher and Tufekci [9] uses lexicographical minimum cost dynamic flow problem for

[*] corresponding author.

the evacuation from buildings which prevents unnecessary movements within the building. Hoppe and Tardos [12, 10] deals with the quickest evacuation problem using the lexicographically maximum (lex-max) solver as a precondition. However, the simplest version of this quickest flow problem has already been solved by Burkard *et al.* [2].

The contraflow approach as a general evacuation problem with lane reversals is NP-hard, Rebennack *et al*. [21], Kim *et al*. [13]. However, the simplest single-source single-sink maximum contraflow evacuation problem is solvable in polynomial time via temporarily repeated flows, Rebennack *et al*. [21]. A polynomial time algorithm as a solution to the earliest arrival contraflow problem on single source and single sink series parallel graphs has been recently investigated, Dhamala and Pyakurel [4]. Moreover, a pseudo-polynomial time algorithm on single source and single sink lossy network for the general case also exist in Pyakurel *et al*. [20]. Pyakurel and Dhamala [18] give systematic formulations of different contraflow problems, and propose efficient algorithms for earliest arrival contraflow and lexicographically maximal dynamic contraflow problems see also [19]. We refer Dhamala [3] for a survey on discrete evacuation problems. In this paper, we propose a solution procedure to the universally maximum contraflow problem.

The organization of the paper is as follows. We summarize some results and notions used in the paper in Section 1 and 2. In Section 3, we discuss universally maximum flow and, we present procedures to obtain universally maximum contraflow in Section 4 and 5.

2. Formulations

Consider a two terminal dynamic network $G = [N, A, T]$ with source s and sink z, where N represents the finite set of nodes, A the set of all arcs taken from N, and T is the permissible time. Time horizon is discretized into time intervals from 0 to T. To each arc (x, y) a certain capacity $c(x, y) \in \mathbb{Z}^+$ with $0 \le c(x, y) \le \infty$ is assigned and $a(x, y)$ is the transit time of the arc $(x, y) \in A$, which also refers as length and cost of the arc. The formulation of maximal dynamic flow problem described by Ford and Fulkerson [6, 7] is:

Maximize $v(T)$

Subject to the constrants

$$\sum_{\tau=0}^{T} \sum_{y\in N} [f(x,y;\tau) - f(y,x;\tau - a(x,y))] = v(T), \qquad \text{for } x = s \qquad (1)$$

$$\sum_{y\in N} [f(x,y;\ \tau) - f(y,;\ \tau - a(x,y)] = 0, \qquad \text{for } x \neq s,z;\ \tau = 0,1,\dots,T \qquad (2)$$

$$\sum_{\tau=0}^{T} \sum_{y\in N} [f(x,y;\ \tau) - f(y,x;\ \tau - a(x,y))] = -v(T), \qquad \text{for } x = z \qquad (3)$$

$$0 \le f(x,y;\ \tau) \le c(x,y), \qquad \text{for } \tau \in \{0,1,\dots,T\}\ \ (x,y) \in A . \qquad (4)$$

Suppose $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ be a set of proper chain flows. The set Γ is a chain decomposition of static flow f if $\sum_{i=1}^{k} \gamma_i = f$, and Γ is a standard chain decomposition of f if all chain flows in Γ use edges in the same direction as feasible dynamic flow f does. If Γ is a standard chain decomposition of f, each chain is no longer than T and f is feasible, then Γ induces a feasible dynamic flow which can be obtained by adding up the dynamic flows induced by each flow in Γ. A dynamic flow computed by this manner is called a standard chain decomposable flow and it is denoted by $[\Gamma]^T$.

3. Universally Maximum Dynamic Flow

Gale [8] described the existence of universally maximum dynamic flow (UMDF) problem in single source – single sink dynamic network. Wilkinson [22] and Minieka [16] presented their algorithms on the time-expanded graphs, which has pseudo-polynomial time complexity. Both of them did not use any form of chain-decomposable flow on their algorithms. Wilkinson [22] considered chain-decomposable flow, but only to prove that standard chain-decomposable flows are insufficient to solve the UMDF problem. Hoppe and Tardos [10] presented the first polynomial algorithm using chain-decomposable flow to approximate UMDF.

A static minimum-cost maximum flow f^* in $G = [N, A, T]$ is computed by shortest augmenting paths. Let $\gamma_1, \gamma_2, \ldots, \gamma_l$ be the sequence of augmentations, $\Gamma^i = \{\gamma_1, \gamma_2, \ldots, \gamma_i\}$, and $f^i = \sum_{j=1}^{i} \gamma_j$. Then Γ^i is a chain decomposition of f^i. Since $a(\gamma_i) \leq a(\gamma_{i+1})$, where $a(\gamma_i)$ is the length of γ_i there is some k between 0 and l such that Γ^k is the set of all chain flows γ_i not longer than T.

Algorithm 1. (Wilkinson [22] and Minieka [16]) *Exact* UMDF

Begin
$\Gamma \leftarrow \phi$
f $\leftarrow$ *0*
while $d_f(s, z) \leq T$ {
P $\leftarrow$ *shortest* $s - z$ *path in the residual network* $\overline{G}(f)$
$\bar{v} \leftarrow$ *minimum residual capacity of* P
augment f by $\bar{v}$ along P
$\Gamma \leftarrow \Gamma + \{\langle \bar{v}, P \rangle\}$
}
return Γ
End.

Theorem 1. (Wilkinson [22] and Minieka [16]) Let $[\Gamma^k]^T$ *denote the dynamic flow induced by the chain-decomposition* Γ^k *obtained by Algorithm 1, then* $[\Gamma^k]^T$ *is a feasible dynamic flow*.

Theorem 2. (Wilkinson [22] and Minieka [16]) *The chain-decomposable flow obtained by* $[\Gamma^k]^T$ *is a* UMDF.

Hoppe and Tardos [10] and Hoppe [11] presented an efficient capacity-scaling algorithm that approximates a UMDF within a factor of $1+\mathcal{E}$, for $\mathcal{E} > 0$.

Algorithm 2. (Hoppe and Tardos [10]) *Approximation of* UMDF

Require: dynamic network $G = [N, A, T]$ with capacity $c(x, y)$ transit time $a(x, y)$ for all $(x, y) \in A$ and let m denote the number of arcs of the network.

***Initially**: chain decomposition set* $\Gamma = \phi$, *scaling factor* $\Delta = 1$, *rounded capacity* $\bar{c}(x, y) = c(x, y)$, *flow* $f(x, y) = 0$, *some* > 0.

while *there exist a* $s - z$ *path in the residual network* $\overline{G}_{\bar{c}}(f)$ *with length* $\leq T$ **do**

set $\sigma = 0$

while $\sigma < \frac{m\Delta}{}$ **do**

find the shortest s - z *path in* $\overline{G}_{\bar{c}}(f)$, *and denote it by P*

$l = \min\{\bar{c}(x, y): (x, y) \in P\}$,

augment the flow f *by* l *along* P *and update the residual capacities* $\bar{c}$ *update the chain decomposition set* $\Gamma = \Gamma \cup \{\langle l, P\rangle\}$

set $\sigma = \sigma + l$

end while

increase the scaling factor: $\Delta = 2\Delta$

round the residual capacities for all $(x, y) \in \overline{A}(f)$,

set $\bar{c}(x, y) = \bar{c}(x, y) - \bar{c}(x, y) \bmod \Delta$.

end while

If $\Gamma \neq \emptyset$, *the dynamic flow* f *can be found by repeating all path in* Γ.

There are $k+1$ scaling phases during the algorithm, numbered 0 to k. We index phases so that $\Delta = 2^i$ during the inner loop of phase i. Let $\Gamma^{-1} = \emptyset$; Γ^i denote the set of chain flows at the end of phase i; T^i denote the length of the largest chain flow in Γ^i; f^i denote the static flow after phase i.

Theorem 3. (Hoppe and Tardos [10]) *The dynamic flow* $[\Gamma^k]^T$ *obtained by Algorithm* 2 *inducing the chain-decomposition* Γ^k *is a feasible dynamic flow.*

The algorithm reduces arc capacities between scaling phases, so that the maximum dynamic flow in the rounded network may be less than the original maximum dynamic flow.

Theorem 4. (Hoppe and Tardos [10]) *Let* $0 \le \tau \le T$, *the maximum dynamic flow value at time* τ *be* v_τ^* *and the chain-decomposable flow* $[\Gamma^k]^T$ *at time* τ *be* $|[\Gamma^k]^T|_\tau$, *then Algorithm 2 computes* $[\Gamma^k]^T$ *in time* $O(m\varepsilon^{-1}(m + n\log n)\log U)$ *such that* $v_\tau^* \le (1+\varepsilon)|[\Gamma^k]^T|_\tau$.

4. Universally Maximum Dynamic Contraflow

The universally maximum dynamic contraflow (UMDCF) problem is the maximum dynamic flow problem in which the sink z must receive as much flow as possible by every intermediate time step up to and including T, and flow should leave the source s as late as possible where arc can be reversed at any integer time step. We propose an algorithm to solve UMDCF problem which is based on algorithm of Wilkinson [22] and maximum dynamic contraflow algorithm of Rebennack *et al.* [21].

Algorithm 3. (Dhungana [5]) *Exact* UMDCF

---

1. *Construct the transformed graph* $\tilde{G} = [N, \tilde{A}, T]$ *of given network,* $G = [N, A, T]$ *where the arc set* $\tilde{A}$ *is defined as:* $(x,y) \in \tilde{A}$ *if* (x, y) or $(y, x) \in A$, *the arc capacity function* $\tilde{c}(x,y)$ *is defined as:* $\tilde{c}(x,y) = c(x,y) + c(y,x)$ *for all* $(x,y) \in \tilde{A}$ *and transit time remains same as in given evacuation network.*
2. *Solve the corresponding UMDF problem on the transformed graph by Algorithm 1 and remove cycle flow.*
3. *The arc* $(y, x) \in A$ *is reversed, if and only if the flow along arc* (x, y) *is greater than* $c(x, y)$ *or if there is a non-negative flow along arc* $(x, y) \notin A$.
4. *Obtain universally maximum dynamic contraflow on* G.

Theorem 5. (Dhungana [5]) *The dynamic flow obtained by Algorithm* 3 *is an optimal UMDCF.*

Proof. We use shortest path in residual network so there is only a flow along one direction of two nodes at the same time as well as at different time period. Thus, the obtained flow is feasible and the problem reduces to a general network. By Theorem 1, obtained flow is feasible dynamic flow, and then by Theorem 2 this flow is UMDF. Hence, Algorithm 3 computes UMDCF optimally. ■

Example 1. *Consider the evacuation network of Figure 1, where the first number of the order pair represents capacity and the second number represents the transit time of the arc. Apply Algorithm 3 to find the exact* UMDCF *to the given evacuation network. First, construct the transformed network as defined by the algorithm then find* UMDF.

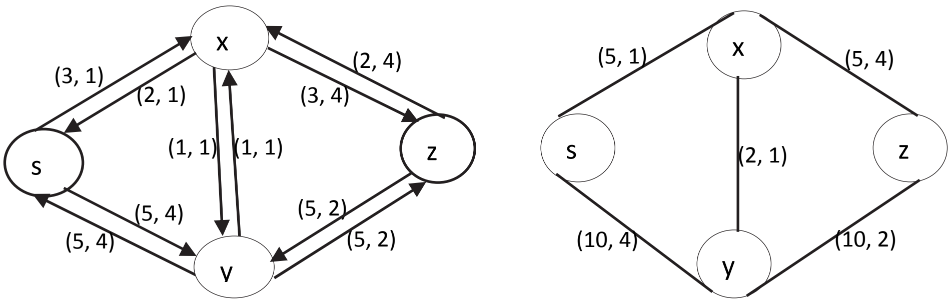

Figure 1: Evacuation and transformed networks respectively.

In Figure 2, the boxes in the arcs contain the following information: upper left numbers are the arc capacities; upper right numbers are the traversal time for the arcs; lower left numbers are the flows through the arcs and lower right numbers are the arc numbers. The numbers on the nodes are the node numbers of dual variable π′s.

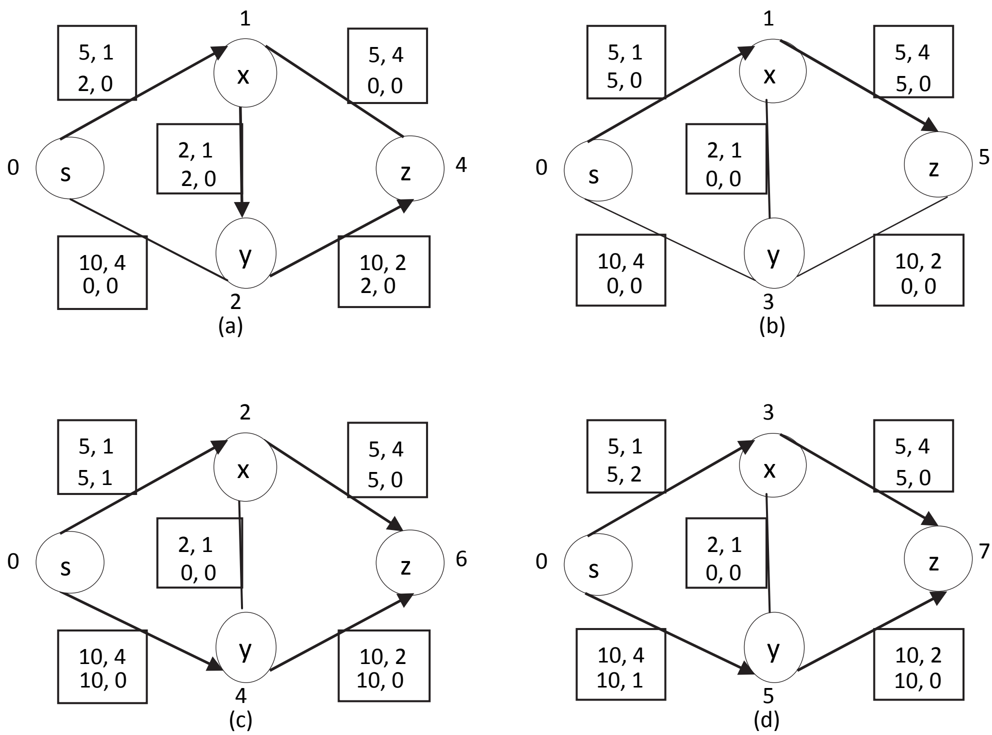

Figure 2: Universal - chain flow graphs at time 4, 5, 6 and 7 respectively.

Initially, all node numbers are zero and unlabeled except the sink whereas the sink has the number $\pi_s = 1$ and is labeled with $[-, \infty]$. If $\pi_x + a(x,y) = \pi_y$ holds for any arc $(x,y) \in A$, then label y from node x; if $\pi_x + a(x,y) \neq \pi_y$ and $\pi_x + a(x,y) < \pi_y$, then increase the arc number $\gamma(x,y)$ by 1 successively until $\pi_x + a(x,y) + \gamma(x,y) = \pi_y$ holds, where $\gamma(x,y)$ represents the repetition of the arc then after label y from x. The universal solutions at time 4, 5, 6 and 7 are shown in the Figure 2(a); Figure 2(b); Figure 2(c) and Figure 2(d) respectively. The universal-chain contraflow of the evacuation network computed by Algorithm 3 is shown in Table 1.

A universal-chain contraflow of evacuation network				
Time	Chain	Length	Flow	Dynamic Flow
4	s-x-y-z	4	2	2
5	s-x-z	5	5	7
6	s-x-z	5	5	12
	s-y-z	6	10	22
7	s-x-z	5	5	27
	s-y-z	6	10	37

Table 1: Universal −chain flow of Figure 2.

5. Approximation of Universally Maximum Dynamic Contraflow

In this section we develop a capacity-scaling algorithm that approximate the universally maximum contraflow in polynomial time, which is based on approximation algorithm of Hoppe and Tardos [10] and maximum dynamic contraflow algorithm of Rebennack *et al.* [21]. Scaling algorithms work initially with capacities rounded by a scaling factor so that large capacity arcs are more important than small capacity arcs. However, in dynamic contraflow, a small capacity short arc might carry more flow than a large capacity long arc. Our algorithm is based on short chain flows.

Algorithm 4. (Dhungana [5]) *Approximation of* UMDCF

---

1. *Set* $f = 0,\ \Delta = 1,\ c'(x,y) = c(x,y),\ \Gamma = \emptyset$ and $\varepsilon > 0$.

2. *Construct the transformed graph* $\widetilde{G} = [N, \widetilde{A}, T]$ *of given network, where the arc set is defined as:* $(x,\ y) \in \widetilde{A}$ *if* $(x,\ y)$ or $(y,\ x) \in A$, *the arc capacity function* $\tilde{c}\,(x,\ y)$ *is defined as:* $\tilde{c}(x,\ y) = c(x,\ y) + c(y,\ x)$ *for all* $(x,\ y) \in \widetilde{A}$ *and transit time remains same as in given evacuation network.*

3. *Solve the corresponding UMDF problem on the transformed graph using Algorithm* 2 *and remove cycle flows.*

4. *Arc* $(y, x) \in A$ *is reversed; iff the flow along arc* (x, y) *is greater than* $c(x, y)$ *or if there is a non-negative flow along arc* $(x, y) \notin A$.

5. *Obtain* $(1+\varepsilon)-$ *approximate UMDCF.*

Algorithm 4 computes minimum-cost flow via shortest augmenting paths in a repeatedly rounded network and use the chain decomposition defined by the sequence of augmentations to induce a dynamic contraflow. The rounding guarantees that the number of augmentations can be bounded by a polynomial in n, $\log U$, and ε^{-1}, where n the number of vertices and U the maximum capacity on transformed graph.

Theorem 6. (Dhungana [5]) *Obtained dynamic flow* $[\Gamma^k]^T$ *by Algorithm* 4 *is feasible.*

Proof. Construction of transformed network and the approximation of UMDF on transformed network are well defined, so it suffices to show that there is flow only in one direction at the same time as well as at different time periods. In approximation algorithm we use shortest path in residual network, so there is only a flow along one direction of two nodes at the same time as well as at different time period. Thus, algorithm is well defined.

We reduce our contraflow network into a general dynamic network and solve the problem by Algorithm 2. By Theorem 3, obtained solution is feasible dynamic flow and hence obtained flow is a feasible dynamic contraflow. ■

We reduce arc capacities between scaling phases, so the maximum dynamic contraflow in the rounded network may be less than the original maximum dynamic contraflow. Thus the approximation of UMDCF cannot exceed the exact UMDCF.

Theorem 7. (Dhungana [5]) *The dynamic flow obtained by Algorithm* 4 *is a* $(1+\varepsilon)-$ *approximate* UMDCF.

Proof. We convert dynamic contraflow problem into general dynamic flow problem. Theorem 4 implies that, the solution obtained on transformed graph is $(1+\varepsilon)-$ approximate UMDF. Also by Theorem 3, obtained solution is feasible $(1+\varepsilon)-$ approximate UMDF on transformed graph. Hence, Algorithm 4 computes $(1+\varepsilon)-$ approximate universally maximum contraflow. ■

Theorem 8. (Dhungana [5]) *The Algorithm 4 computes* $(1+\varepsilon)-$ *approximation* UMDCF *in time* $O(m\varepsilon^{-1}(m + n\log n)\log U)$, *where* m *is the number of arcs,* n *is the number of vertices and* U *is the maximum capacity of the network.*

Proof. The construction of transformed graph and the arc can be reversed in linear time, so the complexity of the algorithm depends upon the running time of Algorithm 2. By Theorem 4, the UMDF problem can be solved in time $O(m\varepsilon^{-1}(m + n\log n)$, hence the approximation solution can be obtain in time $O(m\varepsilon^{-1}(m + n\log n)\log U)$. ■

Example 2. *Let us consider a dynamic contraflow evacuation network as given in Figure 1. The new increased capacities are obtained by adding capacities of both arcs; however the travel time remains the same. A complication arises on the intersecting paths. For example, which direction of arc* (x, y) *or* (y, x) *with added capacity gives the efficient flow through the network? Here, we are choosing the arc* (x, y).

First, we set the scaling factor $\Delta = 1$, *the flow* $f(x, y) = 0$ *and* $\Gamma = \emptyset$. *There exists a* (s, z) *- path with length less or equal* T. *We start with* $\sigma = 0$ *and we have* $\frac{m\Delta}{\varepsilon} = \frac{5\times1}{1.25} = 4$. *Here* $\left(\sigma = 0 < 4 = \frac{m\Delta}{\varepsilon}\right)$ *so we apply the inner loop. The shortest* (s, z)*- path is* $P_1 = \{s, x, y, z\}$ *with length* 4 *and minimal residual capacity* $l = 2$, *thus we augment the flow* f *along* P *by* l. *We set flow value* $v(f) = 2$ *and update the residual capacities and add P to the chain decomposition set* Γ *and set* $\sigma = \sigma + 2 = 2$.

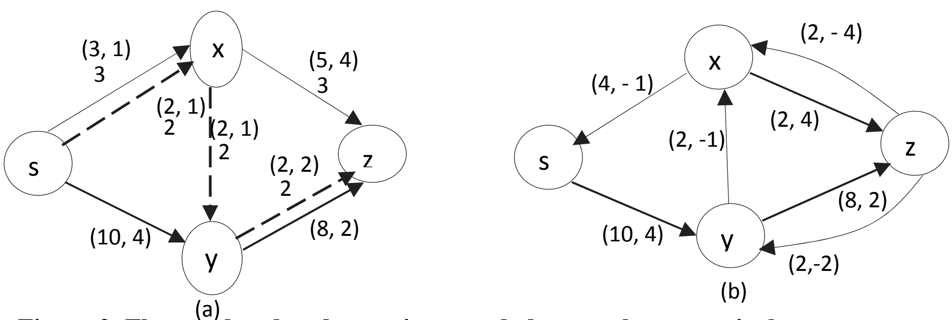

Figure 3: **Flow updated and capacity rounded networks respectively.**

Again, $\sigma = 2 < 4 = \frac{m\Delta}{\varepsilon}$ *and we have* (s, z) *path* $P_2 = \{s, x, z\}$ *with length* 5 *and flow value* 3. *Update the residual network and set* $\sigma = \sigma + 3 = 5$. *Since* $5 < \frac{m\Delta}{\varepsilon}$ *is not possible so we have to increase the scaling factor. Then we set* $\Delta = 2\Delta = 2$ *and round the capacities as shown in Figure 3(b). In the next iteration there exists another* (s, z) *path* $P_3 = \{s, y, z\}$ *with length* 6 *and flow value* 8 *in the rounded residual network. We get* $\sigma = 0$ *and find* $\frac{m\Delta}{\varepsilon} = \frac{5\times2}{1.25} = 8$, *here* $\sigma < \frac{m\Delta}{\varepsilon}$ *so we update the flow value* $v(f) = 5 + 8 = 13$, *the residual capacities and set* $\sigma = \sigma + 3 = 5$. *Further cannot repeat the inner loop again, since* $\sigma < 8$ *is not possible. We can increase the scaling factor to* $\Delta = 4$ *and round the arc capacities, but cannot find new s-z path so that the algorithm terminates.*

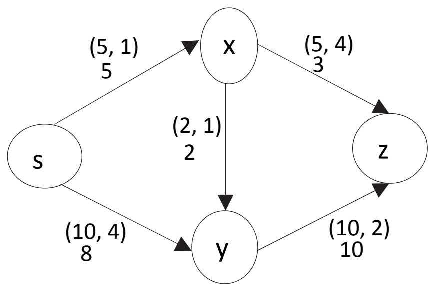

Figure 4: Approximation of universally maximum flow at T=6.

6. Concluding Remarks

We discussed universally maximum flow in Section 3 and universally maximum contraflow in Section 4. Then, we present algorithms for the problem in Section 4, where the sink z receive as much flow as possible by every intermediate time step up to and including T and flow should leave the source s as late as possible where the outbound capacity of the network can be increased by reversing the direction of the arcs towards the sink.

It is still an open question, whether there exists a polynomial time algorithm for the dynamic contraflow problem on multi-terminal network with fixed demand and supply, or not.

Acknowledgements: **The author Tanka Nath Dhamala acknowledges the support of Alexander von Humboldt Foundation.**

References

[1] Arulselvan, A. (2009). *Network model for disaster management*, PhD Thesis, University of Florida, USA.

[2] Burkard, R.E., Dlaska K. Klinz, B. (1993) The quickest flow problem, *ZOR-Methods and Models of Operations Research* **37**, 31-58

[3] Dhamala, T. N., (2015). A survey on models and algorithms for discrete evacuation planning network problems. *Journal of Industrial and Management Optimization,* **11**(1)*,* 265-289.

[4] Dhamala, T.N. and Pyakurel, U., (2013). Earliest arrival contraflow problem on series-parallel graphs, *International Journal of Operations Research*, **10**(1), 1-13.

[5] Dhungana, R.C., (2014). *Efficient Algorithm on Universally Maximum Contraflow*. Master Thesis, Department of Mathematics, Tribhuvan University.

[6] Ford, L.R and Fulkerson, D.R., (1958). Constructing maximal dynamic flows from static networks. *Operations Research*, **6**, 419-133.

[7] Ford, L.R. and Fulkerson, D.R., (1962). Flows in networks. Princeton University Press, Princeton, New Jersey.

[8] Gale, D., (1959). Transient flows in networks. *Michigan Mathematical Journal*, 659-63.

[9] Hamacher, H.W. and Tufekci, S., (1987). On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*, ***34***, 487-503.

[10] Hoppe, B. and Tardos, E., (1994). Polynomial time algorithms for some evacuation problems. Proc. *Of 5th Ann. ACM-SIAMSymp. On Discrete Algo- rithms, pages 433441.*

[11] Hoppe, B., (1995). *Efficient dynamic network flow algorithms*. PhD thesis, Cornell University.

[12] Hoppe, B. and Tardos, E., (2000). The quickest transshipment problem. Mathematics of *Operations Research*, **25**, 36-62.

[13] Kim, S., Shekhar, S. and Min, M., (2008). Contraflow transportation network reconfiguration for evacuation route planning. *IEEE Transactions on Knowledge and Data Engineering*, **20**, 1115-1129.

[14] Koehler, E., Langkau, K. Skutella, M., (2002). Time-expanded graphs for flow-dependent transit times, R. Moehring and R. Raman (Eds.): ESA 2002, LNCS 2461, Springer-Verlag Berlin Heidelberg, 599- 611.

[15] Megiddo, N., (1974). Optimal flows in networks with multiple sources and sinks, Mathematical Programming 7, 97-107.

[16] Minieka, E., (1973). Maximal lexicographic and dynamic network flows. *Operations Research*, **21**, 517-527.

[17] Pardalos, P.M. and Arulselvan, A., (2009). Multimodal solutions for large scale evacuations, Center for Multimodal Solutions for Congestion Mitigation, Department of Industrial and System Engineering, University of Florida.

[18] Pyakurel, U. and Dhamala, T. N., (2014). Lexicographically Contraflow Problem for Evacuation Planning. *Second International Conference on Operations Research Nepal,* **287-294.**

[19] Pyakurel, U. and Dhamala, T.N., (2014). Models and Algorithms on contraflow evacuation Planning Network Problems. *International Journal of Operations Research*, (Reviewed).

[20] Pyakurel, U., Hamacher, H.W. and Dhamala T.N., (2014). Generalized maximum dynamic contraflow on lossy network, *International Journal of Operations Research Nepal*, **3(1)**, 27-44.

[21] Rebennack, S., Arulselvan, A., Elefteriadou, L. and Pardalos, P.M., (2010). Complexity analysis of maximum flow problem with arc reversal. *Journal of Combinatorial Optimization*, **29**, 200-216.

[22] Wilkinson, W.L., (1971). An algorithm for universal maximal dynamic flows in a network. *Operations Research*, **19**, 1602-1612.