International Journal of Operations Research

International Journal of Operations Research Vol. 10, No. 1, 1—13 (2013)

Earliest Arrival Contraflow Problem on Series-Parallel Graphs

Tanka Nath Dhamala[1] and **Urmila Pyakurel**[2*]

[1]Central Departments of Mathematics/CSIT, Tribhuvan University, P.O. Box 13143, Kathmandu, Nepal

[2]Central Departments of Mathematics, Tribhuvan University, P.O. Box 13143, Kathmandu, Nepal

Received 9 December 2012; Revised 2 March 2013; Accepted 19 March 2013

Abstract — Both the earliest arrival flow and the contraflow problems have been highly focused in the research of evacuation planning. We formulated the earliest arrival contraflow problem, where as many evacuees as possible should be sent from dangerous places to safe destinations in every time period by reversing the direction of roads at time zero. A strongly polynomial time algorithm for the earliest arrival contraflow problem on a two-terminal series-parallel graph model having capacities and transit times on the arcs has been presented.

Keywords— Evacuation planning; series-parallel graphs; polynomial algorithms; maximum contraflow; quickest contraflow; earliest arrival contraflow

1. INTRODUCTION

In order to solve emergency problems caused by different disasters, evacuation plans are developed for buildings, stadiums, ships, districts, cities or whole sub-national regions. After any kind of disasters like hurricane, earthquakes, tsunami, flooding, industrial and nuclear accidents, fire and terrorist attacks, etc, there are different real life questions. Most cited examples include the Chichi, Bam and Kashmir earthquakes in Taiwan, Iran and Pakistan, the tsunami in the Indian Ocean and Japan, Asian flooding including China, the September 11 attacks, two major hurricanes Katrina and Rita and recent hurricane Sandy in the USA. There is limited time to take the evacuees from the dangerous state to safe place. We want to evacuate as many evacuees as possible within the limited time because no one wants to die. During the evacuation process, the mathematical model of the transportation network plays an important role. However, it is remarkable that these dynamic network flow models are not limited to the evacuation scenarios but also do have applications in transportation planning like rush office hours, sporting events, concert and mass-meetings, data evacuation on a path in communication networking and quickest execution of jobs by linked processors in network scheduling, for example, Baumann (2007) and Hoppe and Tardos (2000).

A building or a region to be evacuated is given as a network modeled as the set of nodes (i.e., rooms in a building or intersection of streets in a region) and arcs (i.e., doors between rooms, or streets in region), each arc having a capacity and travel time. The sources contain the evacuees and the sinks wait for them with certain capacities. The group of evacuees is considered as flow which passes through the evacuation network. For the transportation system planners the main issue has been the traffic jam. A variant of mathematical modeling have been dealt covering discrete as well as continuous time problems, macroscopic and microscopic behavioral considerations, optimization and simulations techniques and approximate solution techniques including heuristics. More realistic results depend on the time-varying and dynamic time frame with continuous time periods. However, these models and the microscopic based models are very much time consuming through they would provide better accuracy, if a solution could be obtained in reasonable time. In practice, time can be discretized converting a continuous model to a discrete one for approximate solutions in reasonable time. An assumption of constant parameters such as travel time in dynamic network flow results in a lower bound of the real evacuation time.

A majority of the linear programming based methods, which are still applicable for small evacuation scenarios like building evacuations, are heavily suffered from computational time and seek some relaxations on parameters and constraints. The easy heuristic solution methods must compromise with quality although reliable on computational time and scalability. Although a large number of investigations have been made and developed solution methods and algorithms by the communities in mathematics, engineering, computer science and information technology, there is still a lack of efficient solution methods and software implementation because of fast growing applications of huge size problems in this area. We consider the discrete flow over time model.

* Corresponding author's email: urmilapyakurel@gmail.com

Ford and Fulkerson (1958) defined a maximal dynamic flow (MDF) problem to obtain the maximum amount of flow from the sources to the sink for given time period T. The general MDF is equivalent to the static flow in the time-expanded graph they introduced. However, the single source single sink version of the MDF has been solved by temporarily repeating the polynomial time static solution in the given graph itself considered the transit times of the arcs as cost coefficients. A significant research has been carried out since then for the maximum flow and quickest flow evacuation problems. The latter problem concerns with transferring the given number of evacuees from the sources to the destinations in the earliest possible time. Burkard *et al.* (1993) solve the single-source single sink quickest flow problem in polynomial time by linking it with the maximum flow problem and then making use of the binary search and parametric search techniques. Hoppe and Tardos (2000) solve the general quickest flow problem via lexicographically maximum flow solution in the framework of nonstandard chain decomposition. The lexicographically maximum flow problem maximizes the amount of flow leaving each terminal in a given priority order.

Extending this concept of Ford and Fulkerson (1958) and Gale (1959) introduced the earliest arrival flow (EAF) (equivalently, universal maximal flow) to obtain the maximum amount of flow for every time steps θ where $0 \leq \theta \leq T$. solution to this problem avoids the estimated evacuation time requirement and also saves the computational cost by continued maximum solution from the very beginning. Minieka (1973) and Wilkinson (1971) gave exact algorithms for the EAF problem which needed exponential time. Both solutions adopt the successive shortest path augmenting algorithm. Hoppe (1995) reviews these algorithms and represents as chain decomposable flows. A fully polynomial time approximation scheme for this problem with a single source and a single sink was presented by Hoppe and Tardos (1994). Tjandra (2003) proposed an algorithm for the problem with time dependent transit times and capacities which is polynomial in the time horizon T. and the maximum capacity. For multiple sources and a single sink, Baumann (2007) and Baumann and Skutella (2009) gave an algorithm which is polynomial time in the input and output size of the instance. Their solution method does not rely on time-expansion rather recursively construct an earliest arrival flow pattern and then convert it into earliest arrival flow. The earliest arrival flow does not necessarily exist in network with multiple sources and multiple sinks (Schmidt and Skutella, 2010). Gross *et al.* (2012) presented constructive approximations of time and value for earliest arrival flows in arbitrary networks with multiple sinks. Fleischer and Tardos (1998) present a number of efficient continuous-time dynamic network flow algorithms extending the respective discrete-time dynamic flow algorithms.

Steiner (2009) and Ruzika *et al.* (2011) presented a polynomial time algorithm for the earliest arrival flow problem on series-parallel graphs. They constructively proved the existence of a temporally repeated flow on series-parallel graphs with a single source and a single sink having the earliest arrival property. Their exact greedy algorithm is based on a result of Bein *et al.* (1985) for solving minimum cost flow problem on series-parallel graphs. Their algorithm does not use negative circulations as the augmenting paths do not use backward arcs differing to the general graph approach.

In order to solve the problem of traffic jam, contraflow has been considered as a potential remedy by increasing the outbound evacuation route capacity which also maximizes the number of evacuees reaching to the safe destination for each time period. There are only few optimization techniques dealing with contraflow problem in literature. It has been formulated as integer programming problem (Kim *et al.*, 2008). The first polynomial algorithm for contraflow transportation network has been presented by Rebennack *et al.* (2010) and Arulselvan (2009) that solves the maximum static contraflow problem in general and the maximum dynamic contraflow problem for networks with a single source and a single sink in general graphs. They also showed that the quickest contraflow problem can be solved in a strongly polynomial time complexity for a single source and a single sink but the problem in the multiple sources and multiple sinks are NP-hard.

Some heuristic approaches like greedy heuristic and bottleneck relief heuristic of Kim *et al.* (2008), tabu search heuristic of Tuydes and Ziliaskopoulos (2007) are presented in the literature. Experimental results applied to the real-world problems illustrate that the contraflow approach can reduce the evacuation time by 40 percent or more by reversing 30 percent or less of the total arcs. Kim and Shekhar (2005) proposed an iterated simulated annealing heuristic which yields a local minimum with evacuation egress time as the objective function, and random flipping based perturbations. Theodoulon and Wolshon (2004) proposed a non analytical approach of contraflow problem which relies on simulation based methods and decision tools. Unfortunately, there is still lack of study on contraflow approach analytically (see Pardalos and Arulselvan (2009) for the comments).

The maximum flow, the quickest flow and the earliest arrival flow evacuations are the mostly considered objectives in the literature. The requirement of earliest arrival solution also fulfills the other two goals. On the other hand, the contraflow approach though not enough analyzed theoretically takes an important place from the prospective of reducing the traffic mitigation. Our approach combines the earliest arrival flow model as central aspect of evacuation planning and the contraflow configuration as widely implemented evacuation model in practice; for instance (Schmidt and Skutella, 2010; Kim *et al.*, 2008).

In this paper, we introduce the earliest arrival contraflow evacuation problem and present a polynomial time algorithm for this problem on the class of series-parallel graphs. Our algorithm is based on the algorithms of Rebennack *et al.* (2010) and Arulselvan (2009) for computing the maximum dynamic contraflow on general directed graphs and algorithms of Ruzika *et al.* (2011) and Steiner (2009) for computing the earliest arrival flow on series-parallel graphs. To the best of our knowledge, this is the first polynomial time algorithm to find the earliest arrival contraflow. We organize the paper as follows. Section 2 provides some classical results and notions used in the article. In Section 3, we establish our main result on the earliest arrival contraflow on series-parallel graphs. Concluding remarks are given in Section 4.

2. PRELIMINARIES

A dynamic network is defined as a directed graph $G=(V,E,T)$, where V is the set of nodes including a source S^+ and a sink S^- and E is the set of arcs with capacity $u_e \in Z^+$ and transit time $\tau_e \in Z^+$ for each $e \in E$. For each arc $e \in E$, the tail and head are denoted by $t(e)$ and $h(e)$, respectively. A directed path $P(i_0, e_1, i_1, e_2, \ldots, e_k, i_k)$ is an alternating sequence of nodes and arcs with $i_l = t(e_{l+1}) = h(e_l)$ for all $l = 1,\ldots,k-1, i_0 = t(e_1)$ and $i_k = h(e_k)$. A path with $i_k = i_0$ is a directed cycle. The capacity $u_e \in Z^+$ denotes the maximum amount of flow that may enter the arc $e \in E$ per time period and the transit time $\tau_e \in Z^+$ gives the time needed to travel one unit of flow on the arc *e* from node $t(e)$ to node $h(e)$. We assume a finite time horizon $\tau_e \in Z^+$ for the flow to travel through the network with discrete time steps.

Let x be a dynamic network flow function defined as $x: E \times \{0,\ldots,T\} \rightarrow R_0^+$ and $x_e(\theta)$ be the amount of flow that enters arc e at time θ and reaches node $h(e)$ at time $\theta + \tau_e$. Let $G(y) = (V, E^+ \cup E^-)$ be the residual network of graph $G=(V,E)$, where $y: E \rightarrow R_0^+$, is a static network flow, $E^+ = \{+e = e | y_e < u_e\}$, i.e., the set of forward arcs having capacity $u_e - y_e$ and transit time τ_e and $E^- = \{-e = (h(e), t(e) | y_e > 0\}$,i.e., the set of backward arcs having capacity y_e and a transit time of $-\tau_e$. Assume that there are neither arcs e with $h(e) = S^+$ nor with $t(e) = S^-$.

The MDF problem to obtain the maximum amount of flow from the source S^+ to the sink S^- within given time period T can be formulated as a linear programming problem (Ford and Fulkerson, 1958). The formulation is restated as follows (Ruzika *et al.*, 2011).

$$\max f \tag{1}$$

$$s.t. \sum_{\substack{e\in E \\ t(e)=S^+}} \sum_{\theta=0}^{T} x_e(\theta) = f \tag{2}$$

$$\sum_{\substack{e\in E \\ h(e)=S^-}} \sum_{\theta=0}^{T} x_e(\theta - \tau_e) = f \tag{3}$$

$$\sum_{\substack{e\in E \\ h(e)=i}} x_e(\theta - \tau_e) = \sum_{\substack{e\in E \\ t(e)=i}} x_e(\theta) \qquad \forall i \in v \setminus \{s^+, s^-\}, \theta \in \{0,\ldots,T\} \tag{4}$$

$$0 \le x_e(\theta) \le u_e \quad \forall e \in E, \theta \in \{0,\ldots,T\} \tag{5}$$

The objective (1) is to maximize the total amount of flow value f that can be sent from the source S^+ to the sink S^- within the time horizon T. The flow conservation at nodes is described by the constraints (2), (3) and (4). The flow must leave node $t(e)$ at time $\theta - \tau_e$ to reach node $h(e)$ at time θ. The constraint (5) gives the boundary of capacity that may enter arc e at time $\theta \in \{0,\ldots,T\}$. The constraints (2) to (5) should be fulfilled by every feasible dynamic flow. It is assumed that no flow can wait at the intermediate nodes.

For graph $G = (V,E,T)$, a dynamic network flow is equivalent to a static flow in the time expanded network $G_T = (V_T, E_T)$ and vice versa (Ford and Fulkerson, 1958; 1962). Time expanded networks are defined as the expansion of the dynamic network where each node i of the static graph is copied T times to obtain a node $i(\theta)$ for each $i \in V$ and each $\theta \in \{0,\ldots,T\}$. For each arc $e = (i,j) \in E$, let $i(\theta)$ be the tail and $j(\theta + \tau_s)$ be the head of arc e having capacity u_e, called movement arcs. For each arc $e = (i,j) \in E$, let $i(\theta)$ be the tail and $i(\theta + \tau_e)$ be the head of arc e having capacity ∞, called holdover arcs which allows storage at the nodes. We use only movement arcs of the time expanded network because holdover arcs need not be considered for the MDF (Ford and Fulkerson, 1958), and as they do not improve the EAF, (Hoppe and Tardos, 1994), see also Hamacher and Tjandra (2002) for details. But time expanded networks have pseudo-polynomial time complexity.

Consider the underlying static network $G=(V,E)$ to find the minimum cost of sending a flow of given value from the source to the sink, where transit times of arcs are interpreted as cost coefficients. However, a solution of the MDF problem needs to maximize the flow so that the cost of this flow is minimal and the flows arrive at the sink within the given time bound T. Ford and Fulkerson (1958) presented a primal dual algorithm to solve the minimum cost flow problem with respect to the time horizon and then decomposed the obtained flow into the standard chain flows. Afterwards, they obtained the MDF in polynomial time by the temporary repeated flows of these chain flows over time as far as the time permits. The max-flow min-cut theorem on the time-expanded graph establishes a proof that this obtained flow is maximal. Recall that if P_k be the chains obtained from the chain decomposition of the optimal minimum cost flow problem, then the maximum dynamic flow is given by $\sum_{p_k}\left(T+1-\tau_k\right)x_k$ where x_k is the flow along the k^{th} path and τ_k is time taken to travel the k^{th} path.

On the other hand, Klein (1967) presented a polynomial time algorithm which transforms a maximum flow x obtained by some appropriate algorithm on the underlying static network into a minimal cost flow. He makes a use of the result of Busacker and Satty (1965) that a flow x in the static network is of minimal cost if and only if there does not exist a directed cycle of negative total cost in the residual graph $G(x)$. However, his algorithm has to be altered to get a temporary repeated flow for solving the MDF problem. In order to overcome this, an additional arc $\tilde{e}$ from the sink to the source with infinite capacity and transit time $-(\mathrm{T}+1)$, has been added. The transit time τ_e of this modified network $\tilde{G}=(V,\tilde{E})$ with $\tilde{E}=E\cup\{\tilde{e}\}$ is then interpreted as costs of arcs. Then a minimum cost circulation flows (MCCF) problem can be solved as in Algorithm 1. The flow along arc $\tilde{e}$ is ignored and the resulting flow from S^+ to S^- be decomposed into paths. Along each path the static flow value obtained in the path decomposition is sent at each time step for which the flow can reach the sink before time T. The flow obtained by this process is called temporally repeated flow which solves the MDF, (Ford and Fulkerson ,1962), see also Ahuja *et al.* (1993), Steiner (2009) and Ruzika *et al.* (2011).

Algorithm 1: Minimum Cost Circulation Flows (MCCF)

Input: Dynamic network $G=(V,E,T)$ with $u_e\in\mathcal{Z}^+$ and $\tau_e\in\mathcal{Z}^+$, an arc $\tilde{e}=(S^-,S^+)$ with $u_s{}^-,\mathrm{s}^+=\infty$, and $\tau_{s^-,s^+}=-(T+1)$.

1. Set the flow $x_{ij}=0$ for all arcs $(i,j)\in E$
2. Apply "Minimum Cost Flow Algorithm of Klein (1967)" to find a minimum cost flow x^* with transit time of each arc as cost.
3. Decompose x^* into path flows using "The Algorithm of Ford and Fulkerson (1958)".

Output: A maximal dynamic flow

But the approach of standard chain decomposition fails to address a number of other emerging evacuation problems unless some restrictions are imposed, for instance, zero transit times on all arcs. A non-standard flow decomposition approach has been introduced by Hoppe (1995) and Hoppe and Tardos (2000) to solve various evacuation problems including the general lexicographically maximum dynamic flow problem and the multi-terminal quickest flow problem. Note that among these efficient decompositions of flows, the former uses the edges in the same direction of flow whereas the latter allows oppositely directed flow on the edges. The most related problem to the MDF problem is the EAF problem that search a feasible dynamic flow from S^+ to S^- which is maximal for all time periods $0\le\theta\le T$. The flow to be maximal for all times $0\le\theta\le T$ is called the earliest arrival property. By definition, every EAF is also a MDF. Steiner (2009) observed with an example that the converse is not true in general graph. Note that neither every MDF (EAF) is temporarily repeated nor vice versa. It is well known that there might not exist any temporally repeated flow having the earliest arrival property in general networks.

Steiner (2009) and Ruzika *et al.* (2011) considered the earliest arrival flow problem on two-terminal series-parallel graphs as a proper subset of acyclic digraphs. Then, they find temporally repeated flow by standard chain decomposition for the MDF problem and the particular EAF problem in this class of graphs. There exists none series-parallel graph for which a MDF exists by Algorithm 1 that does not satisfy the earliest arrival flow property (c.f., fig. 3(a), Subsection 3.1)

A single arc $e=(S^+,S^-)$ is series-parallel with starting terminal S^+ and end terminal S^-. Let G_1 and G_2 be two series-parallel graphs with starting terminals S_1^+ and S_2^+ and the end terminals S_1^- and S_2^-, respectively. Then the graph $S(G_1, G_2)$ obtained by identifying S_1^- as S_2^- in the series combination is a series-parallel graph, with S_1^+ and S_2^+ as its terminals. The graph $P(G_1, G_2)$ obtained by identifying S_1^+ as S_2^+ and also S_1^- as S_2^- in the parallel combination is a series-parallel graph with $S_1^+\left(=S_2^+\right)$ and $S_1^-\left(=S_2^-\right)$ as its terminals. A two-terminal series parallel graph is denoted by TTSP-graph. Note that Valdes *et al.* (1982) presented a polynomial time algorithm for the recognition of the class of series parallel graphs. The algorithm requires verification of the transitive closure and transitive reduction of the series-parallel graph which can be performed in the time proportional to the size of the input. The recursive structure of the series-parallel graphs can be represented by a decomposing tree (Valdes *et al.*, 1982).

The two-terminal series-parallel graph in fig. 1 is supposed to represent an evacuation network representing the cities by the nodes and the connections between them by the arcs. Each arc *(i, j)* contains the integer capacity u_{ij} and the integer travel time τ_{ij}, for example, the arc from S^+ to A has 10 units capacity and 2 unit travel time. The following property plays a central role in obtaining the min-cost solution with earliest arrival property without taking the negative cycles into consideration. Let $P\left[i,j\right]$ be the sub-graph of series-parallel graph G including all paths from a vertex i to a vertex j in G. In this representation $P\left[i,j\right]$ corresponds to the smallest subtree including all arcs e with $h\left(e\right)=j$ or $t\left(e\right)=i$. Thus $P\left[i,j\right]$ is the inclusion-wise maximal (series or parallel) composition having i and j as its terminals. Any $S^+ - S^-$ path using an arc of this sub-graph must use both i and j.

Bein *et al.* (1985) presented a greedy algorithm that solves the minimum cost flow problem on series-parallel graphs which can be implemented in strongly polynomial time $O\left(\left|V\right|\left|E\right|+\left|E\right|\log\left|E\right|\right)$ for all flow values upto the maximal flow value $f_{\max}$. Their algorithm starts with zero flow, uses a greedy approach and iteratively finds the currently cheapest path P_k with cost $c\left(P_k\right)=\sum_{e\in P_k}\tau_e$ and assigns the flow value $f\left(P_k\right)$ to it. Their augmenting path algorithm does not use backward arcs. The cheapest path capacity has been totally used by this successive shortest paths computing algorithm before flow has to be sent along any other expensive path. It is remarkable that the special minimum cost flow problem on series-parallel graphs constructed from a tree by adding one source S^+ to all arcs with $t\left(e\right)=S^+$ and $h\left(e\right)=v$ for all nodes ν without predecessors of it can be solved in $O\left(\left|E\right|\log\left|E\right|\right)$ steps (Brucker 1982).

Combining Algorithm 1 and greedy algorithm of Bein *et al.* (1985), Steiner (2009) and Ruzika *et al.* (2011) presented Algorithm 2 for series-parallel graphs which solves the MDF and EAF problems in polynomial time although their algorithm does not look for negative directed cycles which was not necessary for obtaining minimum cost flow in this special class of graphs. The latter modification rejects a path if its cost exceeds the given time horizon. In fact an inclusion of a path from the sink to the source with cost –(*T*+ 1) in Ford and Fulkerson (1958) has the same meaning.

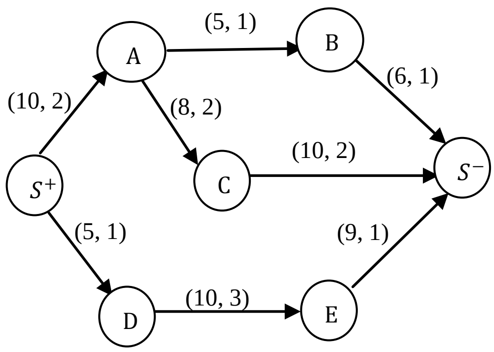

Figure 1. Series-parallel graph

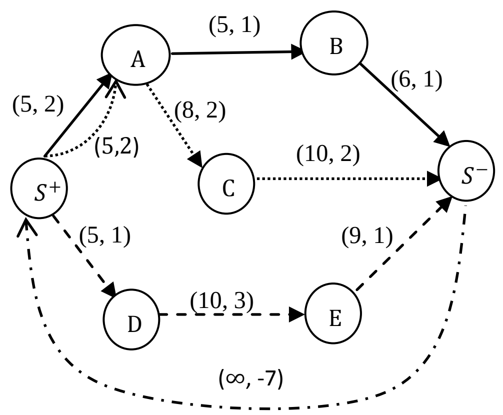

Figure 2. Minimum cost flow on series-parallel graphs

Algorithm 2: Minimum Cost Circulation Flows (MCCF) on TTSP-Graphs

***Input:*:** TTSP-graph $G = (V, E, T)$ with cost $c_{ij} \in Z^+$ and capacity $u_{ij} \in Z^+$ for each arc $(i, j) \in E$. An additional arc (S^-, S^+) with capacity $u_{S^-,S^+} = \infty$ and cost $c_{S^-,S^+} = -(T+1)$

for all $(i, j) \in E$ do
$x_{ij} = 0, k = 0$
end for
while there exists a path connecting S^+ and S^- in G do
$k = k + 1$
Find a minimum cost path P_k and the corresponding cost $c(P_k)$ value.
Expand P_k to a circulation C^k by directing the flow that flows from S^- through P_k *to* S^+ using arc (S^-, S^+).
If $c(P_k) - (T+1)$ then
$f_k = \min\{u_{ij} | (i, j) \in \mathcal{C}^k\}$
else
Stop the algorithm.
end if
for all $(i, j) \in C^k$ do
$u_{ij} = u_{ij} - f_k$
If $u_{ij} = 0$ then $E = E \setminus \{(i, j)\}$
end for
end while

Output: Minimum cost paths P_k, corresponding costs $c(P_k)$, lengths f_k of P_k and also output the last value k^* of the counting variable k for which a flow was sent along the shortest $S^+ - S^-$ path P_{k^*}.

Example 1. Consider fig. 1. In order to find a minimum cost flow on TTSP-graph, the greedy algorithm of Bein *et al.* (1985) uses only forward arcs from the source node to sink node. It is assumed that the flow would be only from the source to sink but not backward. The travel time τ_{ij} has been used as cost $c_{i,j}$. Using the algorithm of Bein *et al.* (1985), the first minimal cost path is $P_1 = (S^+, \mathrm{A}, \mathrm{B}, \mathrm{S}^-)$ with total costs $c(P_1) = 4$ and the maximum flow value $f(P_1) = 5$. Now by updating the capacities we get $u_{S^+,A} = 5, u_{A,B} = 0$ and $u_{B,S^-} = 1$. Then the arc (A, B) is deleted. In the next iteration, we get second minimal cost path $P_2 = (S^+, D, E, S^-)$ with total costs $c(P_2) = 5$ and the maximum flow value $f(P_2) = 5$. The update of the capacities gives $u_{S^+,D} = 0, u_{D,E} = 5$ and $u_{E,S^-} = 4$. Arc (S^+, D) is deleted. Similarly, the next iteration gives the third minimal cost path $P_3 = (S^+, \mathrm{A}, \mathrm{C}, S^-)$ with total costs $c(P_3) = 6$ and the maximum flow value $f(P_3) = 5$. The update of the capacities gives $u_{S^+,A} = 0, u_{A,C} = 3$ and $u_{C,S^-} = 5$. Arc (S^+, A) is deleted. Since there exists no longer path from S^+ to S^- the algorithm terminates.

According to MCCF Algorithm 2, add an additional arc (S^-, S^+) with infinite capacity and $-(T+1)$ transit time in fig. 2. Consider the MDF problem with T = 6. Let the first directed cycle be $\mathcal{C}^1 = (S^+, A, B, S^-, S^+)$ with cost $c(\mathcal{C}^1) = 4 - 7 = -3$ and flow is $f(\mathcal{C}^1) = 5$. nsider the residual network $G_1(x)$. We find $\mathcal{C}^2 = (S^+, D, E, S^-, S^+)$ with cost $c(\mathcal{C}^2) = 5 - 7 = -2$ and flow is $f(\mathcal{C}^2) = 5$ on $G_1(x)$. We update flow x by adding the flow on C^2 and consider the residual network $G_2(x)$. Then again we find another dicycle $\mathcal{C}^3 = (S^+, A, C, S^-, S^+)$ with cost $c(\mathcal{C}^3) = 6 - 7 = -1$ and the flow is $f(\mathcal{C}^3) = 5$. Since there is no any other directed cycle within the boundary cost $c(P_k) - (T+1) < 0$ the algorithm terminates. We update the flow x which is the MDF, see Theorem 1.
Moreover, note that this flow has the earliest arrival property, see Theorem 2.

Theorem 1. (Ruzika *et al*., 2011) *Algorithm 2 solves the MDF problem for TTSP-graph* $G = (V, E, T)$ *with given capacity* $u_{ij} \in Z^+$ *and transit time* $\tau_{ij} \in Z^+$ *on each arc* $(i, j) \in E$ *optimally.*

Recall that a MCCF solution has minimum cost if and only if the corresponding residual network does not contain a cycle with negative cost (Ahuja *et al*., 1993). Ruzika *et al.* (2011) and Steiner (2009) exploit the property of series-parallel graph and showed that every cycle in the residual network has non-negative cycle. Thus, they proved that Algorithm 2 solves the MCCF problem introduced by Ford and Fulkerson (1958) for the MDF problem. The temporally repeated flow obtained by Algorithm 2 is maximal by Theorem 1. On the other direction, Bein *et al.* (1985) proved an additional stronger result that if Algorithm 2 fails to find a MDF solution, then the underlying graph is not series-parallel.

This MDF solution also satisfies the earliest arrival property by Theorem 2 (Ruzika *et al*., 2011; Steiner, 2009). However, it is not true that every earliest arrival flow on series-parallel graph satisfies the temporarily repeated property. A proof of Theorem 2 is similar to the proof of Theorem 5 second part, in Subsection 3.2.

Theorem 2. (Ruzika *et al*., 2011) *Let* $G = (V, E, T)$ *be a TTSP-graph. Let* x *be the MDF of* G *obtained by applying Algorithm 2. Then,* x *is also an optimal solution to the EAF problem.*

3. Contraflow Problems on Series-Parallel Graph

In this section, we present a polynomial time algorithm solving the earliest arrival contraflow (EACF) problem on series-parallel dynamic network (c.f. Subsection 3.2). First we describe polynomial time maximum dynamic contraflow (MDCF) algorithm of Rebennack *et al.* (2010) on general dynamic network with single-source and single-sink (c.f. Subsection 3.1).

3.1 Maximum Dynamic Contraflow Problem

Arulselvan (2009) and Rebennack *et al.* (2010) study contraflow problems with respect to the computational complexity. Given a directed graph $G = (V, E, T)$ with a single source S^+ and a single sink S^- having travel time $\tau_{ij} \in Z^+$ with $\tau_{ij} = \tau_{ji}$ for $(i, j), (j, i) \in E$ and capacity $u_{ij} \in Z^+$ for each $(i, j) \in E$, the MDCF problem requires to find the maximum amount of flow that can be sent within the given integer time T units from the source to the sink S^- if the direction of the arcs can be reversed at time 0.

Arulselvan (2009) and Rebennack *et al.* (2010) presented strongly polynomial time algorithms for the single-source and single-sink MDCF problem and for the multi-terminal maximum static contraflow problem. The time complexities of these algorithms are $O\left(S_2\left(|V|,|E|\right) + S_3\left(|V|,|E|\right)\right)$ and $O\left(\left(S_1\left(|V|,|E|\right) + S_2\left(|V|,|E|\right)\right)\right)$, respectively, where $S_1\left(|V|,|E|\right) = O\left(|V|^2 . \sqrt{E}\right)$ solves the maximum flow problem, $S_2\left(|V|,|E|\right) = O\left(|V| . |E|\right)$ solves the flow decomposition problem and $S_3\left(|V|,|E|\right) = O\left(|V|^2 . |E|^3 . \log|V|\right)$ solves the minimum cost flow problem. They also prove that the quickest contraflow problem with single-source and single-sink is solvable in polynomial time. By reductions from the problems 3-SAT and from PARTITION, respectively, Kim *et al.* (2008) and Rebennack *et al.* (2010) proved that the MDCF problem remains NP-hard in the strong sense even with two sources and one sink or vice versa.

We modify the MDCF algorithm of Arulselvan (2009) and Rebennack *et al.* (2010) making use of the modified MCCF algorithm of Steiner (2009) and Ruzika *et al.* (2011) (c.f. Algorithm 2) on TTSP- graphs. As a result, we obtained a MDCF solution (c.f. Algorithm 3) on series-parallel graphs which has the earliest arrival property.

Example 2. Consider a dynamic contraflow problem as given in fig. 3(a). In fig. 3(b), the arcs of the network from the source to the sink are reversed towards the outgoing arcs at time zero. The new increased capacities u_{ij} are obtained by adding capacities of both arcs, however the travel times τ_{ij} remain the same. A complication arises on the intersecting paths. For example, which direction of arc (A, B) or (B, A) with added capacity gives the efficient flow through the network? Here, we are choosing the arc (A, B) randomly.

For the MDCF problem, we consider maximum flow paths within time horizon $T = 6$ as shown in fig. 3(c). The paths $s^+ - A - s^-$ and $s^+ - B - s^-$ are sufficient for a maximum flow and the arc (A, B) or (B, A) does not affect for the flow value. Fig. 3(c) represents the temporally repeated flow obtained by Ford and Fulkerson (1958). The MDCF problem has the same solution according to the algorithm of Rebennack *et al.* (2010).

But for the EACF problem, as we need shortest distance paths at all successive time periods, a flow through the arc (A, B) is essential due to the shortest path $s^+(0) - A(1) - B(2) - s^-(4)$ as shown in fig. 3(d) at the beginning. A flow of maximum value 4 is reached to the sink at the earliest time $T = 4$ through this path. But it is not yet guaranteed whether this orientation remains valid for the latter time periods. Note that, a solution with this path structure does not represent a temporally repeated flow since the chain flow $s^+(0) - A(1) - B(2) - s^-(4)$ is only started at times zero and one but not at two, though there would be enough time for that chain flow to reach the sink.

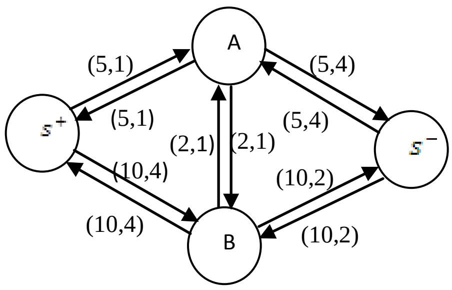

Figure 3(a). Evacuation scenario without contraflow

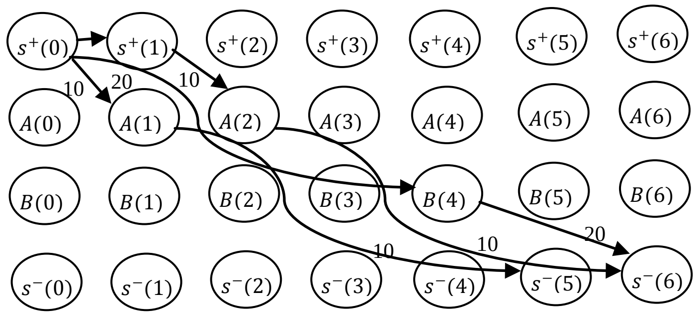

Figure 3(c). Maximum dynamic contraflow for fig. 3(b)

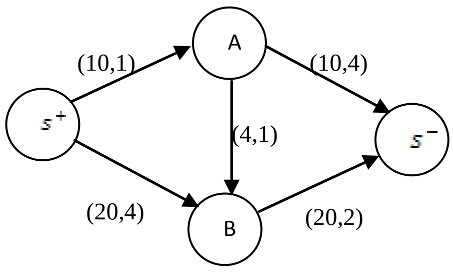

Figure 3(b). Evacuation scenario with contraflow

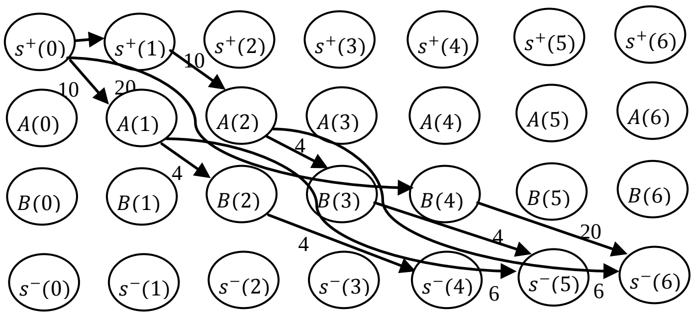

Figure 3(d). A contraflow for fig. 3(a) w.r.t fig. 3(b)

Figure 3. Maximum contraflow and earliest arrival contraflow on general graph

It is observed from this example that the algorithm of Rebennack *et al.* (2010) is unable to find the MDF solution having the earliest arrival property. Moreover, the network given in fig. 3(b) and also with another orientation (B, A) in fig. 3(a) is not a series-parallel graph. Hence the algorithm presented by Rebennack *et al.* (2010) could not find the EACF solution unless the graph is series-parallel. A complication of the EACF problem arises because of the flipping requirements of such intermediate arcs with respect to the time. The dynamic multi-source problem is already NP-hard and an existence of a solution to the dynamic multi-sink problem is uncertain.

Algorithm 3 modifies the algorithm of Rebennack *et al.* (2010) for the MDCF problem on general graphs replacing Step 2 by Algorithm 2 for the earliest arrival flow on series-parallel graphs. Our Algorithm solves the MDCF problem for TTSP-graphs that also satisfy the earliest arrival property.

Algorithm 3: Maximum Dynamic Contraflow (MDCF) on TTSP-Graphs

Step 1: Construct the graph $G' = (V, E', T)$ with arc set defined as $(i, j) \in E'$ if $(i, j) \in E$ or $(j, i) \in E$

The arc capacity function u' is given by $u'_{ij} = u_{ij} + u_{ji}$

For all arcs $(i, j) \in E'$ the transit time is $\tau'_{ij} \left(= \tau'_{ji}\right) = \begin{cases} \tau_{ij} & \text{if } (i, j) \in E \\ \tau_{ji} & \text{Otherwise} \end{cases}$

Step 2: Generate a dynamic temporally repeated flow on transformed series-parallel graph G' with capacity u' and travel time τ'_{ij} using Algorithm 2.

Step 3: Obtain flow decomposition into path and cycle flows of the network of Step 2. Remove the cycles.

Step 4: Arc $(j,i) \in E$ is reversed, if and only if the flow along arc (i,j) is greater than u_{ij} or if there is a non-negative flow along arc $(i,j) \notin E$ and the resulting flow is MDF with the arc reversal for the graph $G = (V,E,T)$.

We give the proofs of Lemma 1 and Theorem 3 for the sake of completeness similar to the results of Rebennack *et al.* (2010) that yields the correctness of Algorithm 3.

Lemma 1. *For TTSP-graph* $G = (V,E,T)$*, the maximum amount of flow in the MDCF problem is less than or equal to the optimal flow in the maximum contraflow for the corresponding time expanded graph* $G_T = (V_T, \mathrm{E}_T)$.

Proof:

Every feasible flow to the MDCF problem has an equivalent feasible flow to the maximum contraflow problem of the time expanded graph of the given series-parallel graph.

Theorem 3. *Algorithm 3 solves the MDCF problem for TTSP- graph* $G = (V,E,T)$ *optimally.*

Proof:

To show the feasibility of Algorithm 3, it is enough to show that only Step 4 is well defined because Steps 1-3 are feasible. After the flow decomposition, the optimal solution outcomes in a set of paths from source to sink and a set of cycles with positive flows obtained from Step 3. The positive flow along all cycles could be cancelled and hence there is no flow along any cycle. Therefore, there is either a flow along arc (i,j) or (j,i) but never in both arcs. This proves that the flow is not greater than the reversed capacities on all the arcs at all time units.

Next, we show the optimality. Depending on the feasibility we conclude that every feasible MDF of the dynamic flow problem in the transformed series-parallel graph $G' = (V, E', T)$ is feasible to the MDCF in the series- parallel graph $G = (V,E,T)$, i.e.,

$$\left[G = (V, E)\right]_{MDCFopt} \geq \left[G' = (V\ E')\right]_{MDFopt} \quad (6)$$

According to Lemma 1, we have $\left[G = (V, E)\right]_{MDCFopt} \leq \left[G_T = (V_T\ E_T)\right]_{MDFopt}$.

Now the maximum contraflow in graph $G_T = (V_T, \mathrm{E}_T)$ is equivalent to the maximum flow in the graph $G'_T = (V_T, \mathrm{E}'_T)$, where the arc set E' is defined as $(i,j) \in E'_T$, if $(i,j) \in E_T$ or $(j,i) \in E_T$ and the capacity function u' is given by $u'^{\tau}_{ij} = u'^{\tau}_{ij} + u'^{\tau}_{ji}$. Thus, we have $\left[G_T = (V_T, E_T)\right]_{MCFopt} = \left[G'_T = (V_T, E'_T)\right]_{MFopt}$.

According to the (Ford and Fulkerson 1958) and the MCCF algorithm, the maximum flow in the time expanded graph $G'_T = (V_T, \mathrm{E}'_T)$ can be obtained by a temporally repeated chain flow of the static series-parallel graph $G' = (V, \mathrm{E}')$. This implies that

$$\left[G = (V,E)\right]_{MDCFopt} \leq \left[G'_T = (V_T, E'_T)\right]_{MFopt} = \left[G' = (V, E')\right]_{MDFopt} \quad (7)$$

Combining results (6) and (7), we obtain the result.

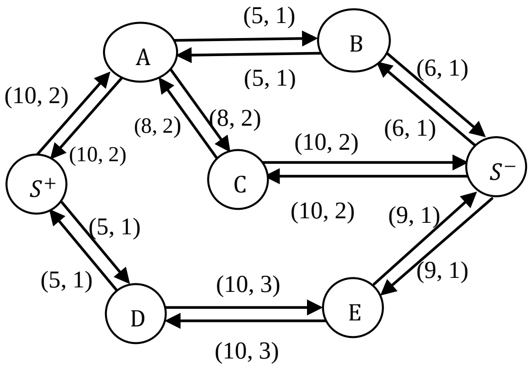

Figure 4(a). Evacuation scenario of TTSP-graph

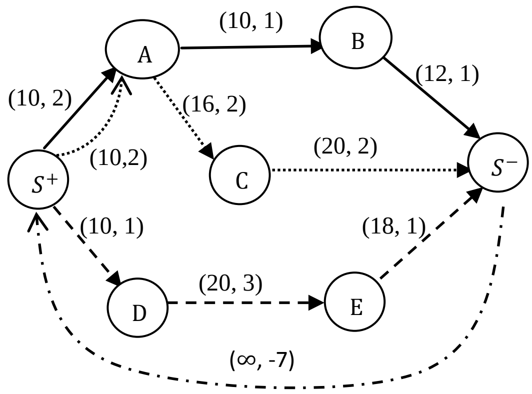

Figure 4(b). Contraflow configuration of TTSP-graph

Example 3. We use the contraflow model to the network given in fig. 4(a). By Step 1 of Algorithm 3, reverse all backward arcs towards the sink S^- as shown in fig. 4(b). The contraflow configuration of this series-parallel graph will give the MDCF solution. The MDF is obtained by solving the minimum cost flow problem using Algorithm 2 as the case without contraflow. Here the MDF value is 30. The flow along the backward arc $\left(S^-, S^+\right)$ is ignored and the flows from S^+ to S^- are decomposed into the directed cycles. Let the first directed cycle be $\mathcal{C}^1 = \left(S^+, A, B, \mathrm{S}^-, \mathrm{S}^+\right)$ with cost $c(\mathcal{C}^1) = 4 - 7 = -3$ and flow is $f\left(\mathcal{C}^1\right) = 10$. Consider the residual network $\mathrm{G}_1^{'}\left(x\right)$ and look for another dicycle. The second directed cycle is $\mathcal{C}^2 = \left(S^+, D, E, S^-, S^+\right)$ with cost $c(\mathcal{C}^2) = 5 - 7 = -2$ and flow is $f\left(\mathcal{C}^2\right) = 10$. We update flow x by adding the flow on C^2 and again consider the residual network $\mathrm{G}_2^{'}(x)$. Then we find another directed cycle $\mathcal{C}^3 = \left(S^+, A, C, S^-, S^+\right)$ with cost $c(\mathcal{C}^3) = 6 - 7 = -1$ and the flow is $f\left(\mathcal{C}^3\right) = 10$. Since there is no any other directed cycle within the boundary cost $c(P_k) - \left(T + 1\right) < 0$ the algorithm terminates. The MDCF solution is represented in fig. 5.

The complexity in case of TTSP-graph has been improved in comparison to the general graphs.

Theorem 4. *Algorithm 3 solves the MDCF problem in strongly polynomial time for TTSP-graphs.*

Proof:

Step 1 and Step 4 are solved in linear time. The flow decomposition in Step 3 can be done in $O\left(\left|V\right|\left|E'\right|\right)$ time. The running time of the Algorithm 2 is $O\left(\left|V\right|\left|E'\right| + \left|E'\right| log \left|E'\right|\right)$ (Ruzika *et al.*, 2011). Hence total complexity of the Algorithm 3 is $O\left(\left|V\right|\left|E'\right| + \left|E'\right| log \left|E'\right|\right)$.

3.2 Earliest Arrival Contraflow Problem

Given a directed TTSP-graph $G = \left(V, E, T\right)$ with single-source $S^+ \in V$ and single-sink $S^- \in V$ having travel time $\tau_{ij} \in Z^+$ on each arc $e \in E$ with $\tau_{ij} = \tau_{ji}$ if $(i, j)(j, i) \in E$ and capacity $u_{ij} \in Z^+$ for each arc $(i, j) \in E$, the EACF problem on series-parallel graphs requires the maximum amount of flow that can be sent in every time period $\theta, 0 \le \theta \le T$ from the source to the sink with arc reversal capability at time zero. The problem of EACF could be formulated similarly on general graphs.

We show that Algorithm 3 also solves the EACF problem for series-parallel graphs.

Theorem 5. *Any MDCF solution induced by Algorithm 3 has earliest arrival property for TTSP-graph.*

Proof:

Algorithm 3 gives a MDCF solution on the transferred series-parallel graph $G' = \left(V, E', T\right)$. The fact that the temporarily repeated MDCF flow obtained by Algorithm 3 has the earliest arrival flow property, follows from Theorem 2 of Ruzika *et al.* (2011). They prove by contradiction that the temporally repeated flow obtained from Algorithm 3 does not improve on its residual graphs proposed by Minieka (1973). This proof makes use of an inclusion-wise minimal parallel components in series-parallel graphs.

Theorem 6. *Algorithm 3 solves the EACF problem optimally in strongly polynomial time for TTSP-graphs.*

A proof of Theorem 6, follows directly from Theorems 3-5, and Algorithm 3. From the definition, all the EACF solutions are the MDCF solutions. Theorem 3 proves that all the MDCF has the earliest arrival property for a series-parallel graph. Therefore, we state

Corollary 1. *Any MDCF solution is also an EACF solution and vice versa for TTSP-graphs.*

Example 4. Fig. 5 represents the time-expanded network of fig. 4(b), which gives an EAF solution. The first 10 flow units arrive to the sink through the path $S^+ - A - B - S^-$ at time 4. Another 20 flow units is added through the paths $S^+ - A - B - S^-$ and $S^+ - D - E - S^-$ at time 5. Another 30 more flow units reach to the sink at time 6 through three paths $S^+ - A - B - S^-$, $S^+ - D - E - S^-$ and $S^+ - A - C - S^-$. Hence 60 flow units has reached at the sink in total up to time six. Note that this flow is temporally repeated because all paths are used continuously to flow from the source to sink and it will be continuous with the increment of the time horizon T. There is a temporally repeated flow obtained by Algorithm 3 and has the earliest arrival property in TTSP-graphs.

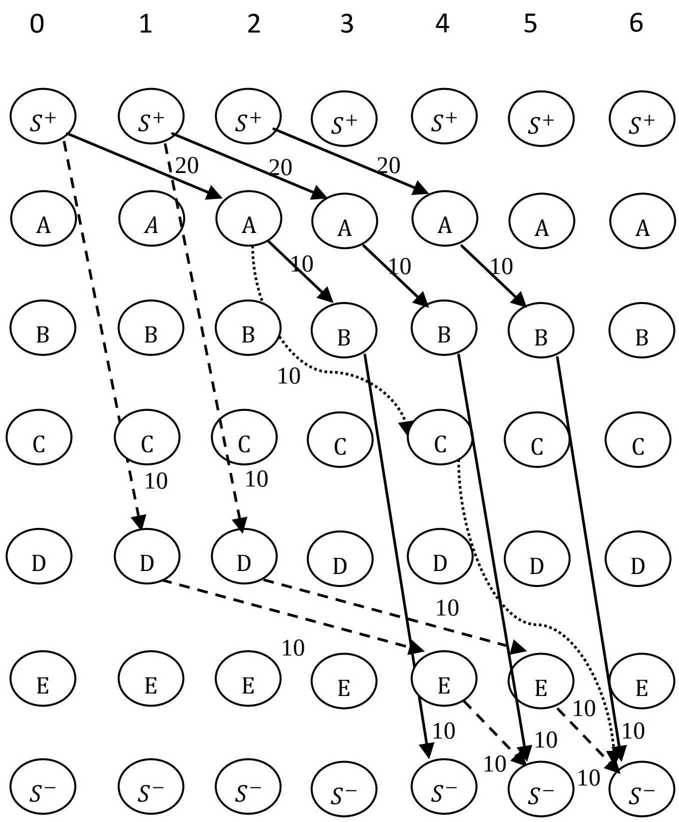

Figure 5. Earliest arrival contraflow for series- parallel graph of fig. 4(b)

4. CONCLUSION

Arulselvan (2009) and Rebennack *et al.* (2010) present a strongly polynomial time algorithm that gives the maximal dynamic contraflow solution for general graphs. Steiner (2009) and Ruzika *et al.* (2011) present a strongly polynomial time algorithm that gives the earliest arrival flow solution on two-terminal series-parallel graphs. Based on their algorithms, we presented a strongly polynomial time algorithm for two-terminal series-parallel graphs that computes the MDCF solution within the time horizon T in a discrete time setting. The solution relies on the temporally repeated flows. The presented algorithm also gives the EACF solution because it yields the existence of temporally repeated flow on series-parallel graphs which has the earliest arrival property. If the algorithm is unable to find the MDCF having the earliest arrival property, it is concluded that the underlying graph is not series-parallel.

To the best of our knowledge, this paper introduces a mathematical modeling of the earliest arrival contraflow problem in evacuation planning.

ACKNOWLEDGMENT

The first author acknowledges the support of Alexander von Humboldt Foundation and Prof. Dr. Horst W. Hamacher for the research stay at the University of Kaiserslautern, Germany. The authors are very grateful to the anonymous referees for their valuable comments and suggestions to improve the paper.

REFERENCES

1. Ahuja, R.K., Magnati, T.L. and Orlin, J.B. (1993). *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, Englewood Cliffs, New Jersey.
2. Arulselvan, A. (2009). *Network Model for Disaster Management*, PhD Thesis, University of Florida, USA
3. Baumann, N. (2007). *Evacuation by earliest arrival flows,* PhD thesis. Department of Mathematics*,* University of Dortmund, Germany.
4. Baumann, N. and Skutella, M. (2009). Solving evacuation problem efficiently: earliest arrival flows with multiple sources. *Mathematics of Operations Research,* 34: 499-512.
5. Bein, W.W., Brucker, P. and Tamir, A. (1985). Minimum cost flow algorithms for series-parallel networks. *Discrete Applied Mathematics*, 10: 117-124.
6. Brucker, P. (1982). Network flows in trees and knapsack problems with nested constraints. *Proceedings on 8th Conference on Graphtheoretic Concepts in Computer Science.*
7. Burkard, R.E., Dlaska, K. and Klinz, B. (1993). The quickest flow problem .*ZOR-Methods and Models of Operations Research*, 37: 31-58.
8. Busacker, R.C. and Saaty, T.L. (1965). *Finite Graphs and Networks.* McGraw-Hill, New York.
9. Ford, L.R. and Fulkerson, D.R. (1958). Constructing maximal dynamic flows from static networks. *Operations Research,* 6: 419-133.
10. Fleischer, L. and Tardos, E. (1998). Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23: 71-80.
11. Ford, L.R. and Fulkerson, D.R. (1962). *Flows in Networks.* Princeton University Press, Princeton, New Jersey.
12. Gale, D. (1959). Transient flows in networks. *Michigan Mathematical Journal,* 6: 59-63.
13. Gross, M., Kappmeier1, J-P.W., Schmidt, D.R., and Schmidt, M. (2012). Approximating Earliest Arrival Flows in Arbitrary Networks. L. Epstein and P. Ferragina (Eds.): *ESA 2012*, LNCS 7501: 551-562.
14. Hamacher, H.W. and Tjandra, S.A. (2002). Mathematical modeling of evacuation problems: a state of the art. In *Pedestrian and Evacuation Dynamics*, 227-266.
15. Hoppe, B. (1995). *Efficient dynamic network flow algorithms*. PhD thesis, Cornell University.
16. Hoppe, B. and Tardos, E. (1994). Polynomial time algorithms for some evacuation problems. *Proc of 5th Ann, ACM-SIAM Symp, on Discrete Algorithms*, 433-441.
17. Hoppe, B. and Tardos, E. (2000). The quickest transshipment problem. *Mathematics of Operations Research*, 25: 36-62.
18. 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.
19. Kim, S. and Shekhar, S. (2005). Contraflow network reconfiguration for evacuation planning: a summary of results. *Proceedings of 13th ACM Symposium on Advances in Geographic Information Systems* (GIS 05), 250-259.
20. Klein, M. (1967). A primal method for minimum cost flows with applications to the assignment and transportation problem. *Management Science,* 14: 205-220.
21. Minieka, E. (1973). Maximal lexicographic and dynamic network flows. *Operations Research,* 21: 517-527.
22. 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.

23. 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.
24. Ruzika, S., Sperber, H. and Steiner, M. (2011). Earliest arrival flows on series- parallel graphs. *Networks*, 10: 169-173.
25. Schmidt, M. and Skutella, M. (2010). Earliest arrival flows in networks with multiple sinks. *Electronics Notes in Discrete Mathematics,* 36: 607-614
26. Steiner, M. (2009). *A Survey on Earliest Arrival Flows and a Study of the Series-Parallel Case*. Diploma Thesis*,* Department of Mathematics, University of Kaiserslautern.
27. Theodoulou, G. and Wolshon, B. (2004). Alternative methods to increase the effectiveness of freeway contraflow evacuation. *Transportation Research Record: Journal of Transportation Research Board,* 1865: 48-56.
28. Tjandra, S.A. (2003). *Dynamic Network Optimization with Application to the Evacuation Problem*. PhD Thesis, Department of Mathematics, University of Kaiserslautern.
29. Tuydes, H. and Ziliaskopoulos, A. (2007). Tabu-based heuristic for optimization of network evacuation contraflow. *Proceedings 85th Annual Meeting of the Transportation Research Board.*
30. Valdes, J., Tarjan, R.E. and Lawler, E.L. (1982). The recognition of series-parallel digraph. *SIAM Journal on Computing*,11: 198-313.
31. Wilkinson, W.L. (1971). An algorithm for universal maximal dynamic flows in a network. *Operations Research,* 19: 1602-1612.