2006

Approximation Algorithms for Survivable Multicommodity Flow Problems with Applications to Network Design

Ajay Todimala
University of Nebraska - Lincoln, ajayt@cse.unl.edu

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Proceedings INFOCOM 2006. 25th IEEE International Conference on Computer Communications.
Digital Object Identifier: 10.1109/INFOCOM.2006.92
Publication Year: 2006 , Page(s): 1 - 12

Approximation Algorithms for Survivable Multicommodity Flow Problems with Applications to Network Design

Ajay Todimala and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln NE 68588-0115 U.S.A.
Email: {ajayt, byrav}@cse.unl.edu

***Abstract*— Multicommodity flow (MF) problems have a wide variety of applications in areas such as VLSI circuit design, network design, etc., and are therefore very well studied. The fractional MF problems are polynomial time solvable while integer versions are $\mathcal{NP}$-complete. However, exact algorithms to solve the fractional MF problems have high computational complexity. Therefore approximation algorithms to solve the fractional MF problems have been explored in the literature to reduce their computational complexity. Using these approximation algorithms and the randomized rounding technique, polynomial time approximation algorithms have been explored in the literature.**

In the design of high-speed networks, such as optical wavelength division multiplexing (WDM) networks, providing survivability carries great significance. Survivability is the ability of the network to recover from failures. It further increases the complexity of network design and presents network designers with more formidable challenges. In this work we formulate the survivable versions of the MF problems. We build approximation algorithms for the survivable multicommodity flow (SMF) problems based on the framework of the approximation algorithms for the MF problems presented in [1] and [2]. We discuss applications of the SMF problems to solve survivable routing in capacitated networks.

Keywords: Multicommodity Flow, Linear Programming, Approximation Algorithms, Network Design.

I. Introduction

Multicommodity flow (MF) problems are used to model a variety of problems in areas such as VLSI circuit design, network design etc. The following are the most popular versions of the MF problems, *maximum multicommodity flow (MMF)* problem, *maximum concurrent flow (MCF)* problem and *minimum cost maximum concurrent flow (MC-MCF)* problem. For the definition of a flow in a network, please see Section II or refer to [29]. Let us define these problems. Given a directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow R^+$ and a set of k commodities with a node pair (s_i, t_i) corresponding to the i^{th} commodity. The MMF problem is to compute a flow for each commodity (s_i, t_i) such that the sum of the flows is maximized. An instance of the MCF problem includes an instance of the MMF problem and a demand $d(i)$ for each commodity. The MCF problem is to determine the maximum value of the parameter $\lambda \leq 1$ such that the flow f_i for each commodity is at least $\lambda d(i)$. An instance of the MC-SMCF problem includes an instance of the MMF problem, edge cost function $s : E \rightarrow R^+$ and a constant B. The objective of the MC-MCF problem, similar to the MCF problem, is to compute a maximum concurrent flow with the additional constraint that its cost is at most B.

The above discussed MF problems are fractional versions where flow values are positive real numbers. The fractional versions of the MF problems are polynomial time solvable. However, the exact algorithms have high computational complexity and are not practical for large problem sizes. Recent research studies have focused on developing efficient approximation schemes for the MF problems. The integer versions of the MF problems are $\mathcal{NP}$-complete. They have applications both in packet-switched and circuit switched network design and more recently in the design of future generation optical wavelength division multiplexing (WDM) networks. MF problems are extensively used as the fundamental problems in the design of circuit switched networks [3], [4] [5] and also in reliable network design [6], [7]. The MF problems are also used to solve the routing and wavelength assignment problem in WDM networks and routing in high-speed capacitated networks.

The authors in [8] and [9] model the routing and wavelength assignment problem in WDM networks

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

with the objective of maximizing the number of connections as a maximum concurrent flow problem and present an ILP formulation. They present algorithms to solve the formulation based on the solution to the fractional linear programming formulation of the MCF problem. The author in [10] presents a probabilistic approach (randomized rounding) to provide a deterministic algorithm to the integral MF problem using the approximation scheme for the fractional MF problem. The authors in [11] and [12] model the routing and wavelength assignment (RWA) problem in multi-fiber WDM networks with limited wavelength conversion as an integer multicommodity flow problem. They use the approximation scheme for the fractional MMF problem presented in [2] and improve the randomized rounding scheme of [10] to provide a deterministic approximation algorithm for the integer MF problem. Therefore providing approximation schemes for fractional MF problems is critical, as these schemes are used to build approximation schemes for integral MF problems, which in turn are used to model the routing problem in high-speed networks.

Survivability, the ability to recover from failures, is a critical issue in the design of optical WDM and high-speed capacitated networks. Survivability further increases the complexity of the design of optical WDM networks. Survivable design of WDM networks under static traffic was studied in [14], [15], [16], [17], [18], [19], [20], and [21]. Most of the current approaches solve the survivable routing problem in WDM networks either by directly solving complex ILP formulations that are extremely time consuming or by using heuristics or meta heuristics such as tabu-search, which do not provide any guarantee on the optimality of the solution obtained. In this work we formulate the survivable versions of the multicommodity flow problems (SMF) that are then used to model survivable routing in WDM networks.

Several variations of the capacitated network design problems have been studied in the literature. The authors in [22] and [23] study a variation of the capacitated network design problem with the objective of installing additional capacity to route all the specified point-to-point traffic demands at minimum cost. The authors in [24] present a survey of models and algorithms for multicommodity capacitated network design problems. Survivable capacitated network design has received little attention. The authors in [25] and [26] present approximation algorithms for survivable static routing using link-based protection for fast restoration in capacitated networks. In link-based protection the traffic is routed around the failed link to support fast restoration but at the expense of lower resource utilization. In this work we formulate survivable versions of the MF problems that are then used to model the static routing problem using shared path protection in capacitated networks.

To the best of our knowledge this is the first work that presents approximation algorithms to survivable multicommodity flow problems. If all possible paths between a given node pair (s, t) are considered valid for pushing flow between (s, t) then even the fractional survivable multicommodity flow problems are $\mathcal{NP}$-complete and also hard to approximate. In this work we assume that the number paths that can be used to push flow between a given node pair is limited and formulate the survivable multicommodity problems accordingly. The motivation for such an assumption stems from the observation that not every path between a given node pair satisfies the QoS constraints. Recent research studies on network design with/without network survivability have presented heuristics that use a limited number of paths such as the K-shortest paths. We present approximation schemes to these limited survivable versions of the multicommodity flow problems.

The rest of the document is organized as follows. In Section II, we define a survivable flow and a survivable multicommodity flow under shared protection in a network. In Section III, we discuss the general idea to build an approximation scheme exploiting primal-dual relationships in linear programming. The primal-dual linear programming formulations and their relationship are discussed in this section. Also in Section III, we present a framework used in formulating the survivable flow problems as a linear program using a limited number of paths, popularly known arc-chain formulations. In Section IV, we present primal and dual arc-chain formulations of the Survivable Maximum Multicommodity Flow (SMMF) problem. We then present an approximation scheme, proof of its approximation and an analysis of its running time. In Sections V and VI, we formulate and present a similar study of the Survivable Maximum Concurrent Flow (SMCF) and Minimum Cost Survivable Concurrent Flow (MC-SMCF) problems respectively. In Section VII, we discuss applications of SMF problems in survivable routing in capacitated networks under shared path protection. In Section VIII, we present the concluding remarks.

II. Survivable Multicommodity Flow Problems

In this section, we define a survivable flow and a survivable multicommodity flow in a graph. A flow between a pair of nodes (s, t) in a graph G is defined as an assignment of values to directed edges of the graph also called edge flows or flows, such that the flow on any edge

does not exceed its capacity and for every node n in the graph except s and t, the amount of incoming flow at the node n is equal to the amount of outgoing flow from the node n. The total outgoing flow from node s is equal to the total incoming flow into node t and is called the flow value. For more information on network flows and related concepts, please refer to [29].

A. Survivable Flow

Given a graph $G = (V, E)$ with edge capacities $c : E \rightarrow R^+$ and a node pair (s, t), a survivable flow from a source node s to a destination node t consists of two flows, a primary flow and a secondary flow such that for every primary flow of value f along a path p from s to t there exists a distinct secondary flow also of value f along a disjoint path q. The value of the survivable flow is the sum of flows f along all such disjoint path pairs.

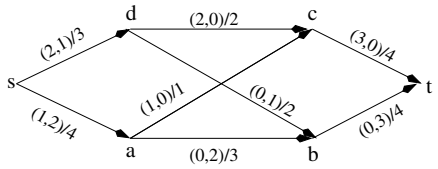

Fig. 1. *Illustration of a survivable flow.*

Fig. 1 illustrates the survivable flow in a graph between node pair (s, t). The primary flow p_e and the secondary flow s_e along an edge e with capacity c_e is represented as follows $(p_e, s_e)/c_e$. For example, as shown in Fig. 1, the flow representation along the edge (s, d) with capacity 3 is $(2, 1)/3$ where the value of the primary and secondary flows on the edge are 2 and 1 units respectively. Fig. 1 shows that a primary flow of value 2 units is pushed along path $\overrightarrow{sdct}$ and the corresponding secondary flow is pushed along path $\overrightarrow{sabt}$. Similarly, a primary and its corresponding secondary flow of value 1 unit is pushed along the path pair $(\overrightarrow{sact}, \overrightarrow{sdbt})$. Fig. 1 shows the resultant flow of pushing the primary and their secondary flows along the path pairs $(\overrightarrow{sdct}, \overrightarrow{sabt})$ and $(\overrightarrow{sact}, \overrightarrow{sdbt})$. The total survivable flow value is 3.

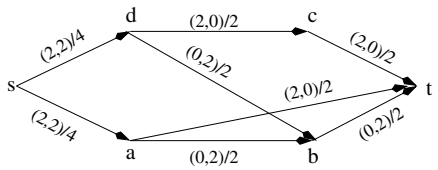

Fig. 2. *Illustration of a survivable flow under shared path protection.*

A survivable flow under shared path protection is defined such that for every pair of primary flows along disjoint paths p_1 and p_2 with flow values f_1 and f_2 respectively, the corresponding secondary flows along a common edge e need only $max\{f_1, f_2\}$ amount of flow. Fig. 2 illustrates a survivable flow under shared protection between node pair (s, t). Let us push a survivable flow of value 2 units along the path pair $(\overrightarrow{sdct}, \overrightarrow{sabt})$. Let us push another survivable flow of value 2 units along path pair $(\overrightarrow{sat}, \overrightarrow{sdbt})$. Since the primary flow along path $\overrightarrow{sdct}$ is disjoint with the primary flow along path $\overrightarrow{sat}$ their corresponding secondary flows can share the capacity on the edge $\overrightarrow{bt}$. Therefore, as shown in Fig. 2, the edge $\overrightarrow{bt}$ has a total secondary flow of only 2 units while it supports both the primary flows, each of value 2 units. Shared protection thus reduces the amount of capacity required on the secondary (protection) paths compared to a dedicated protection scheme. In the rest of the paper, we refer to 'survivable flow under shared path protection' as simply 'survivable flow'. The problem of computing disjoint path pairs under shared path protection can be reduced to the problem of computing a survivable flow with specific optimization criteria.

B. Survivable Multicommodity Flow under Shared Protection

Given a graph $G = (V, E)$ with edge capacities $c : E \rightarrow R^+$ and k node pairs (s_i, t_i). A survivable multicommodity flow under shared protection is such that for any primary flow of value f_1 disjoint with any other primary flow of value f_2 the corresponding units of secondary flows need only $max\{f_1, f_2\}$ units of capacity on a common edge e.

III. Approximation Technique for LP Problems

In this section we briefly describe the general technique of designing approximation algorithms/schemes for a problem based on its linear programming formulation.

A. Primal-dual formulation

Let F_p be a linear programming formulation of an optimization problem π. F_p is called the primal formulation of π. For each primal formulation there exists a corresponding dual formulation F_d. If the primal is a maximization problem then its dual is a minimization problem and vice versa. Given a feasible solution to the primal formulation F_p, a solution for the dual formulation F_d corresponding to the given feasible primal solution can be computed in polynomial time in terms of number of variables and constraints of the primal formulation. For more information on the primal-dual relationship, please refer to [27].

B. Approximation Basis

Let us suppose that primal F_p is a maximization problem and therefore its dual F_d is a minimization problem. One of the important properties of the primal-dual formulations is that the objective value of any feasible solution for F_p is at most the objective value of any feasible solution of its dual. We can conclude from this property that if the objective value obj_p of a feasible solution S_p of primal F_p is equal to the objective value obj_d of a feasible solution S_d of F_d then S_p and S_d are optimal solutions of the corresponding primal and dual formulations. Comparing the objective values of the primal and dual formulations we can measure the closeness of a solution S_p to the optimal solution. Therefore, if S_p and S_d are the feasible solutions of the primal and dual formulations respectively and their corresponding objective values are obj_p and obj_d respectively then the feasible solution for the primal S_p is at most $((obj_d - obj_p)/obj_d)\%$ away from the optimal solution.

C. Survey of Approximation Schemes for MF problems

Based on the above discussed technique, the authors in [28] proposed an approximation scheme to the MMF problem. The author in [2] presents a brief review of the evolution of the approximation schemes for the fractional multicommodity flow problems. The authors in [1] presented approximation algorithms that use shortest paths computations instead of minimum cost flow computations. This work also suggested simple modifications to the approximation algorithms to determine a lower bound on the capacity of edges in the graph such that approximate solutions to integral MF problem can be computed in polynomial time. The algorithms presented in [2] further improve the run-time complexity of the approximation schemes of multicommodity flow problems. The run-time of the approximation algorithm for the maximum multicommodity flow problem presented in this work is independent of the number of commodities.

D. Node-arc and Arc-chain formulations

The multicommodity flow formulations can be formulated using either node-arc or arc-chain formulations. The node-arc formulation constraints include the flow-conservation constraints [29] while arc-chain formulations do not. The arc-chain formulations use a set of paths as input to the formulation while node-arc formulations consider all possible paths (though not provided as explicit input to the formulation). In the following section we present a framework that describes the set of disjoint path pairs supplied as input to the arc-chain formulations of the survivable multicommodity flow formulations.

E. Disjoint-path pair framework for Arc-chain formulations of MF

For each node pair (s_i, t_i) a set of candidate primary and secondary path pairs are already chosen that satisfy the quality of service constraints such as delay, signal strength etc., and denoted by Γ_i. The universal set of path pairs Γ is defined as the union of the set of candidate paths pairs for all the node pairs (s_i, t_i) i.e., $\Gamma = \cup_i \Gamma_i$. The set of path pairs whose secondary path uses the link e is defined as $\Psi_e = \{(p,q)|(p,q) \in \Gamma \wedge e \in q\}$. The set of path pairs Ψ_e are partitioned into a minimal number of sub-sets/partitions such that primary paths of the path pairs belonging to a partition are pair-wise disjoint. Let the set of partitions of Ψ_e be represented as $\Pi(\Psi_e)$. The problem of computing such a partitioning into least number of partitions is NP-hard. This problem can be solved by recursively solving the maximum independent set problem. But the maximum independent set problem is NP-hard and also hard to approximate. Simple heuristics can be used to compute such a partitioning of Ψ_e.

Let us define two parameters based on the partitioning of the set Ψ_e for all $e \in E$ that are later used in the analysis of the approximation schemes for the SMF problems. Let k_1 be the maximum number of disjoint path pairs in any partition $\mathcal{Q}$ of Ψ_e for all edges e in the graph. Formally, $k_1 = max\{|\mathcal{Q}| \mid \forall \mathcal{Q} \in \Pi(\Psi_e) \wedge \forall e \in E\}$ where $|\mathcal{Q}|$ is the size of the set $\mathcal{Q}$. Let k_2 be the maximum number of partitions of Ψ_e for all edges e in the graph G. Formally, $k_2 = max\{|\Pi(\Psi_e)| \mid \forall e \in E\}$.

IV. Survivable Maximum Multicommodity Flow (SMMF)

In this section, we present an approximation algorithm for the survivable maximum multicommodity flow problem. The problem is defined as follows. Given a graph $G = (V, E)$ with edge capacities $c : E \rightarrow R^+$ and k node pairs (s_i, t_i). The problem is to compute a survivable flow for each commodity i under shared protection such that the sum of the flows of all the commodities is maximized.

The flow along a path pair P is denoted by $w(P)$. To push a survivable flow f of value $|f|$ along a path pair $P = (p, q)$ implies pushing a primary flow of value $|f|$ along the primary path p and the supporting secondary flow of value at most $|f|$ along disjoint secondary path q. Notice that for primary flow of value $|f|$ pushed along the primary path p the supporting secondary flow

along path q need not always be equal to $|f|$, for the following reason. The amount of secondary flow that is assigned to an edge e for protecting the primary flow along the path pairs belonging to a partition $\mathcal{Q} \in \Psi_e$ is defined as $b(e, \mathcal{Q})$. Since all the primary paths of the path pairs in a partition $\mathcal{Q}$ are pair-wise disjoint, their secondary flows along secondary paths can share capacity on a common edge e. Therefore, for pushing secondary flow of value $|f|$, equivalent to support the primary flow, along the secondary path q, only the difference flow of value $|w(P) - b(e, \mathcal{Q})|$ is pushed along edge $e \in q$ where $w(P)$ is the total flow along path pair P after pushing a flow of value $|f|$ and $\mathcal{Q} \in \Psi_e$ s. t. $P \in \mathcal{Q}$. This difference flow is called the exclusive flow pushed along edge e for supporting primary flow along the path $p \in P$ s.t. $P \in \mathcal{Q}$ and $\mathcal{Q} \in \Psi_e$. The amount of secondary flow along a link e, $b(e)$, is the sum of the secondary flows assigned to link e for all partitions $\mathcal{Q}$, $\mathcal{Q} \in \Psi_e$. The objective of the primal is to maximize the sum of the surivable flows along all the path pairs in the set Γ. The arc-chain primal formulation $\mathcal{P}_{SMMF}$ of the SMMF problem follows.

$$\text{Maximize} \sum_{P \in \Gamma} w(P)$$

$$\forall e \in E : \sum_{P=(p,q): e \in p} w(P) + \sum_{\mathcal{Q} \in \Psi_e} b(e, \mathcal{Q}) \leq c(e)$$

$$\forall e \in E \wedge \forall P \in \mathcal{Q} \text{ s.t.} \mathcal{Q} \in \Psi_e : w(P) - b(e, \mathcal{Q}) \leq 0$$

$$\forall P \in \Gamma : w(P) \geq 0$$

$$\forall \mathcal{Q} \text{ and } e \text{ s.t. } \mathcal{Q} \in \Psi_e : b(e, \mathcal{Q}) \geq 0$$

The dual of the problem is to assign primary lengths $l^p(e)$ to the edges of the graph for carrying primary flow and secondary lengths $l^s(e, P)$ to the edge e for carrying secondary flow for the path pair P that belongs to set of path pairs $\mathcal{Q} \in \Psi_e$. Let us define $D(l^p) = \sum_e l^p(e)c(e)$ where l^p is the length function. The objective of the dual formulation is to minimize $D(l^p)$. The dual formulation $\mathcal{D}_{SMMF}$ follows.

$$\text{Minimize} \sum_{e \in E} l^p(e)c(e)$$

$$\forall P \in \Gamma : \sum_{e \in p} l^p(e) + \sum_{e \in q} l^s(e, P) \geq 1$$

$$\forall \mathcal{Q} \text{ and } e \text{ s.t. } \mathcal{Q} \in \Psi_e : l^p(e) - \sum_{P \in \mathcal{Q}} l^s(e, P) \geq 0$$

$$\forall e \in E : l^p(e) \geq 0$$

$$\forall e \in E \wedge \forall P \in \mathcal{Q} \text{ s.t.} \mathcal{Q} \in \Psi_e : l^s(e, P) \geq 0$$

The cost of a path pair P, $C(P) = \sum_{e \in p} l^p(e) + \sum_{e \in q} l^s(e, P)$. Let $P^{min}(l^p, l^s)$ be the path pair $P \in \Gamma$ with minimum cost and let $\gamma(l^p, l^s) = C(P^{min}(l^p, l^s))$ be its cost. Now the dual problem is equivalent to finding length functions (l^p, l^s) such that $D(l^p)/\gamma(l^p, l^s)$ is minimized. Since if length function (l^p, l^s) that satisfies the constraints of the dual formulation $\mathcal{D}_{SMMF}$ then the length function $(l^p/\gamma(l^p, l^s), l^s/\gamma(l^p, l^s))$ also satisfies the constraints of the formulation $\mathcal{D}_{SMMF}$ and has lower objective value. Let β be the optimal objective value, therefore by definition $\beta \leq D(l^p)/\gamma(l^p, l^s)$. Let $\gamma_p(l^p)$ be the cost of the primary path of the path pair $P^{min}(l^p, l^s)$, therefore $\gamma(l^p, l^s) \geq \gamma_p(l^p)$. Hence $\beta \leq D(l^p)/\gamma(l^p, l^s) \leq D(l^p)/\gamma_p(l^p)$. Let $\alpha(l^p) = min_i\{dist_i(l^p)\}$ where $dist_i(l^p)$ is the length of the shortest primary path among the path pairs $P \in \Gamma_i$ between node pair (s_i, t_i) with length function l^p. Therefore $\alpha(l^p) \leq \gamma_p(l^p)$. And hence $\beta \leq D(l^p)/\gamma_p(l^p) \leq D(l^p)/\alpha(l^p)$.

A. The SMMF Algorithm

We term the algorithm for solving the maximum multicommodity flow (MMF) problem presented in [1] as the MMF Algorithm. We build our SMMF Algorithm for solving the SMMF problem based on the MMF Algorithm. The SMMF Algorithm is outlined in Figure 3. The length function of all the edges e in the graph is initialized to δ_1, a constant to be determined later. The length function corresponding to the secondary flow that is reserved on edge e for protecting flow along disjoint path pair $P \in \mathcal{Q}$ where $\mathcal{Q} \in \Psi_e$, $l^s(e, P)$, is initialized to δ_2. The SMMF Algorithm iterates steps 4-9 until the stopping condition is satisfied. In one iteration only the flow along the disjoint path pair P is changed. Though the flow along the path pair is increased, actually only the flow along the primary path p of the path pair P is incremented. The exclusive flow pushed on the links e along the secondary path q is only the difference between the required secondary flow of value $w(P)$ and the existing secondary flow value $b(e, \mathcal{Q})$. The sum of all the exclusive secondary flows pushed through edge e by the SMMF Algorithm for protecting the primary flow along path pair P is $b(e, \mathcal{Q}, P)$. Let $b(e, \mathcal{Q})$ be the secondary flow through edge e for protecting the primary flow along the primary paths of the disjoint path pairs in $\mathcal{Q}$. Therefore, $\sum_{P \in \mathcal{Q}} b(e, \mathcal{Q}, P) = b(e, \mathcal{Q})$ and for all path pairs $P \in \mathcal{Q}$, $w(P) \leq b(e, \mathcal{Q})$.

The length of the links along the the primary path p are increased exponentially proportional to the flow increment c. The length of the links along the secondary path is incremented exponentially proportional to exclusive flow increment, $w(P) - b(e, \mathcal{Q})$. The constants ϵ, δ_1 and δ_2 are determined in the analysis of the SMMF Algorithm based on the required approximation guarantee.

SMMF Algorithm

Initialize $\forall e : l^p(e) = \delta_1 \; ; w = 0$
Initialize $\forall e \in E \wedge \forall P \in \mathcal{Q}$ s.t. $\mathcal{Q} \in \Psi_e : l^s(e,P) = \delta_2 \; ; b = 0$
while $P = (p,q) \in \Gamma$ with $l(p) < 1$ exists **do**
  select a path pair $P = (p,q) \in \Gamma$ with $l(p) < 1$
  $c \leftarrow min_{e \in p \vee e \in q}\{c(e)\}$
  $w(P) \leftarrow w(P) + c$
  $\forall e \in p : l^p(e) \leftarrow l^p(e)(1 + \epsilon c / c(e))$
  for $\forall e \in q \wedge \mathcal{Q} \in \Psi_e \; s.t. \; P \in \mathcal{Q}$ **do**
    $b(e,\mathcal{Q}) \leftarrow max\{w(P), b(e,\mathcal{Q})\}$
    if $(w(P) > b(e,\mathcal{Q}))$ **then**
      $l^s(e,P) \leftarrow l^s(e,P)(1 + \epsilon \frac{(w(P) - b(e,\mathcal{Q}))}{c(e)})$
    end if
  end for
end while

Fig. 3. The outline of the SMMF Algorithm.

B. The Analysis

The following two lemmas of the MMF Algorithm also hold for the SMMF Algorithm.

Lemma 1: If f_t is the flow computed by the SMMF Algorithm at the end of t iterations when the stopping condition is satisfied then it holds that $\frac{\beta}{f_t} \leq \frac{\epsilon}{\ln(\delta_1 n)^{-1}}$

Lemma 2: Let e be any edge in the graph and if the SMMF Algorithm stops at the end of t iterations when the stopping condition is satisfied then the primary flow through edge e, $w(e)$, is at most $c(e) \log_{1+\epsilon} \frac{1+\epsilon}{\delta_1}$.

Lemma 3: Let e be any edge in the graph and if the SMMF Algorithm stops at the end of t iterations when the stopping condition is satisfied then $b(e,\mathcal{Q})$ is at most $c(e) k_1 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2}$.

Proof: For every $c(e)$ units of the exclusive secondary flow pushed along the link e for protecting flow along the primary path $P \in \mathcal{Q}$ and $\mathcal{Q} \in \Psi_e$ the length $l^s(e,P)$ is incremented by a factor of at least $1+\epsilon$. If at the end of t iterations $b_t(e,\mathcal{Q},P)$ is the total exclusive secondary flow pushed along link e for protecting flow along primary path $P \in \mathcal{Q}$ and $\mathcal{Q} \in \Psi_e$ then $l^s_t(e,P) \geq l^s_0(e,P)(1+\epsilon)^{\frac{b_t(e,\mathcal{Q},P)}{c(e)}} = \delta_2(1+\epsilon)^{\frac{b_t(e,\mathcal{Q},P)}{c(e)}}$. Let us define $l^s(e,\mathcal{Q}) = \sum_{P \in \mathcal{Q}} l^s(e,P)$. Therefore,

$$\begin{aligned} l^s_t(e,\mathcal{Q}) &= \sum_{P \in \mathcal{Q}} l^s_t(e,P) \\ &\geq \delta_2 \sum_{P \in \mathcal{Q}} (1+\epsilon)^{\frac{b_t(e,\mathcal{Q},P)}{c(e)}} \\ &\geq \delta_2 k_1 (1+\epsilon)^{\frac{b_t(e,\mathcal{Q})}{k_1 c(e)}} \end{aligned}$$

From the proof of Lemma 2, we know that $l^p(e)$ is at most $1+\epsilon$. Imposing the constraint of the dual problem that $l^p(e) \geq l^s(e,\mathcal{Q})$. Therefore $1+\epsilon \geq \delta_2 k_1 (1+\epsilon)^{\frac{b_t(e,\mathcal{Q})}{k_1 c(e)}}$ leading to conclude that $b(e,\mathcal{Q})$ is at most $c(e) k_1 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2}$. ■

Lemma 4: The SMMF problem has a feasible flow of value $f_t / \left(\log_{1+\epsilon} \frac{1+\epsilon}{\delta_1} + k_1 k_2 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2} \right)$.

Proof: From Lemma 2 and Lemma 3 the total amount of flow through link e is at most $c(e) \log_{1+\epsilon} \frac{1+\epsilon}{\delta_1} + c(e) k_1 k_2 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2}$ where k_2 is the maximum number of partitions of set Ψ_e.

Therefore scaling the flow by $\log_{1+\epsilon} \frac{1+\epsilon}{\delta_1} + k_1 k_2 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2}$ we get a feasible flow. ■

Theorem 1: The SMMF Algorithm is a $2(1+\omega)$ approximation algorithm for the SMMF problem.

Proof: Let γ be the ratio of the dual and primal solutions, then

$$\begin{aligned} \gamma &\leq \frac{\beta}{f_t} \left(\log_{1+\epsilon} \frac{1+\epsilon}{\delta_1} + k_1 k_2 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2} \right) \\ &= \frac{\epsilon \left(\log_{1+\epsilon} \frac{1+\epsilon}{\delta_1} + k_1 k_2 \log_{1+\epsilon} \frac{(1+\epsilon)}{k_1 \delta_2} \right)}{\ln(\delta_1 n)^{-1}} \\ &= \frac{\epsilon}{\ln(1+\epsilon)} \left(\frac{\ln \frac{1+\epsilon}{\delta_1}}{\ln(\delta_1 n)^{-1}} + k_1 k_2 \frac{\ln \frac{(1+\epsilon)}{k_1 \delta_2}}{\ln(\delta_1 n)^{-1}} \right) \end{aligned}$$

If $\delta_1 = (1+\epsilon)((1+\epsilon)n)^{-1/\epsilon}$, we get $\frac{\ln \frac{1+\epsilon}{\delta}}{\ln(\delta n)^{-1}} = 1/(1-\epsilon)$. And if $\delta_2 = \frac{(1+\epsilon)}{k_1}(\delta_1 n)^{1/(k_1 k_2 (1-\epsilon))}$, we get $k_1 k_2 \frac{\ln \frac{(1+\epsilon)}{k_1 \delta_2}}{\ln(\delta_1 n)^{-1}} = 1/(1-\epsilon)$. Therefore,

$$\begin{aligned} \gamma \leq \frac{2\epsilon}{(1-\epsilon)\ln(1+\epsilon)} &\leq \frac{2\epsilon}{(1-\epsilon)(\epsilon - \epsilon^2/2)} \\ &\leq 2(1-\epsilon)^{-2} \end{aligned}$$

Let ϵ be chosen such that $1 + \omega \leq (1-\epsilon)^{-2}$, then the SMMF Algorithm is a $2(1+\omega)$ approximation algorithm. ■

C. Running Time

Let c_{min} be the minimum capacity of all the links in the graph. We define the capacity ratio of an edge e, η_e, as the ratio of its capacity and minimum capacity in the graph, $c(e)/c_{min}$. We also define capacity ratio of a graph G, η_G, as the ratio of the sum of capacities of all the edges in the graph and its minimum capacity, $\sum_e c(e)/c_{min}$.

Theorem 2: The number of iterations of the SMMF Algorithm is bound by $\lceil \eta_G \rceil \lceil \frac{1}{\epsilon} \log_{1+\epsilon} n \rceil$.

Proof: For $c(e)$ units of primary flow routed through edge e the primary length increases by at least $1+\epsilon$. From Lemma 2 the amount of primary flow through edge e is at most $c(e) \log_{1+\epsilon} \frac{1+\epsilon}{\delta_1}$. Undoing the flow pushed in the previous iteration which resulted in satisfying the

stopping condition, the amount of flow through edge e is at most $c(e)\log_{1+\epsilon}\frac{1}{\delta_1}$. Therefore the maximum number of iterations needed to route $c(e)\log_{1+\epsilon}\frac{1}{\delta_1}$ amount of primary flow is $\lceil\frac{c(e)}{c_{min}}\log_{1+\epsilon}\frac{1}{\delta_1}\rceil = \lceil\frac{\eta_e}{\epsilon}\log_{1+\epsilon}n\rceil$. In each iteration a primary flow of value at least c_{min} is pushed along at least one edge. Therefore, summation over all the m edges in the graph, we get the upper bound on the number of iterations of the SMMF Algorithm as $\lceil\frac{\eta_{e_1}}{\epsilon}\log_{1+\epsilon}n\rceil + \ldots + \lceil\frac{\eta_{e_m}}{\epsilon}\log_{1+\epsilon}n\rceil \leq \lceil\frac{1}{\epsilon}\log_{1+\epsilon}n\rceil\lceil(\eta_{e_1}+\ldots+\eta_{e_m})\rceil = \lceil\eta_G\rceil\lceil\frac{1}{\epsilon}\log_{1+\epsilon}n\rceil$. ■

In Step 4 of the SMMF Algorithm we need to compute a path pair (p,q) among all the path pairs for every node pair such that $l(p)$ is shortest. The simplest algorithm to compute such a path pair (p,q) assuming that the number of path pairs for a given node pair is an arbitrary constant c takes $\mathcal{O}(kn)$ since the maximum number of edges along any loop less path is limited by the number of nodes n and k is the number of commodities. From Theorem 2 the run-time of the algorithm is $\mathcal{O}(\lceil\eta_G\rceil\lceil\frac{1}{\epsilon}\log_{1+\epsilon}n\rceil kn)$.

V. Survivable Maximum Concurrent Flow (SMCF)

In the following section, we adopt a similar approach to the one described above, for developing an approximation algorithm for the SMCF problem. We are given a graph $G=(V,E)$ with edge capacities $c: E \rightarrow R^+$, k node pairs (s_i,t_i) corresponding to k commodities and a demand $d(i)$ is associated with each commodity i. The problem is to find the largest λ such that there are at least $\lambda d(i)$ units of survivable flow for each commodity i under shared protection. The primal formulation $\mathcal{P}_{SMCF}$ follows.

$$\text{Maximize } \lambda$$
$$\forall e \in E: \sum_{P=(p,q):e\in p} w(P) + \sum_{\mathcal{Q}\in\Psi_e} b(e,\mathcal{Q}) \leq c(e)$$
$$\forall j: \sum_{P\in\Gamma_j} w(P) \geq \lambda d(j)$$
$$\forall e \in E \wedge \forall P \in \mathcal{Q}\ \text{s.t.}\mathcal{Q}\in\Psi_e : w(P) - b(e,\mathcal{Q}) \leq 0$$
$$\forall P \in \Gamma : w(P) \geq 0$$
$$\forall e \in E, \forall \mathcal{Q}\in\Psi_e : b(e,\mathcal{Q}) \geq 0; \lambda \geq 0$$

The dual of the problem is to assign primary lengths $l^p(e)$ to the edges of the graph for carrying primary flow and secondary lengths $l^s(e,P)$ to the edge e for carrying secondary flow for the path pair P that belongs to set of path pairs $\mathcal{Q}\in\Psi_e$. Let us define $D(l^p)=\sum_e l^p(e)c(e)$ where l^p is the length function. The objective of the dual formulation is to minimize $D(l^p)$. The dual formulation $\mathcal{D}_{SMCF}$ follows.

$$\text{Minimize} \sum_{e\in E} l^p(e)c(e)$$
$$\forall P\in\Gamma : \sum_{e\in p} l^p(e) + \sum_{e\in q\wedge\mathcal{Q}\in\Psi_e, s.t. P\in\mathcal{Q}} l^s(e,P) \geq z_j$$
$$\forall e\in E\wedge\forall\mathcal{Q}\in\Psi_e : l^p(e) - \sum_{P\in\mathcal{Q}} l^s(e,P) \geq 0$$
$$\sum_j z_j d(j) \geq 1$$
$$\forall e\in E : l^p(e) \geq 0$$
$$\forall e\in E\wedge\forall P\in\mathcal{Q}\ \text{s.t.}\mathcal{Q}\in\Psi_e : l^s(e,P)\geq 0$$
$$\forall j : z_j \geq 0$$

We know that the cost of a path pair P, $C(P) = \sum_{e\in p} l^p(e) + \sum_{e\in q} l^s(e,P)$. Let $P_j^{min}(l^p,l^s)$ be the path pair $P \in \Gamma_j$ with minimum cost and let $\gamma(l^p,l^s)$ be the sum of the costs of pushing $d(j)$ units of flow for j^{th} commodity along the path pair $P_j^{min}(l^p,l^s)$, for all commodities. Therefore $\gamma(l^p,l^s) = \sum_j d(j)C(P^{min}(l^p,l^s))$. The dual formulation now reduces to finding primary and secondary length functions (l^p,l^s) such that $D(l^p)/\gamma(l^p,l^s)$ is minimized where $D(l^p)=\sum_{e\in E} l^p(e)c(e)$. Let β be the optimal objective value, then $\beta \leq D(l^p)/\gamma(l^p,l^s)$. Let $\gamma_p(l^p)$ be the sum of the costs of pushing $d(j)$ units of flow along the primary path of the path pair $P^{min}(l^p,l^s)$, for all commodities. Therefore $\gamma(l^p,l^s)\geq\gamma_p(l^p)$. Hence $\beta \leq D(l^p)/\gamma(l^p,l^s) \leq D(l^p)/\gamma_p(l^p)$. Let $\alpha(l^p)$ be the sum of costs of pushing $d(j)$ units of flow along the shortest primary path among the path pairs $P\in\Gamma_i$ between node pair (s_i,t_i) with length function l^p. Therefore $\alpha(l^p)\leq\gamma_p(l^p)$. And hence $\beta \leq D(l^p)/\gamma_p(l^p) \leq D(l^p)/\alpha(l^p)$.

A. The Algorithm

In this section, we describe the SMCF Algorithm for approximating the SMCF problem. The outline of the SMCF Algorithm is shown in Fig. 4. The SMCF Algorithm is built based on the MCF Algorithm [1] for the maximum concurrent flow problem that uses shortest path computation instead of min-cost flow computations. The length function of all the edges e in the graph is initialized to $\delta_1/c(e)$, where δ_1 is a constant to be determined later. The length function corresponding to secondary flow that is reserved on edge e for protecting flow along disjoint path pair $P\in\mathcal{Q}$ where $\mathcal{Q}\in\Psi_e$, $l^s(e,P)$, is initialized to $\delta_2/c(e)$. The algorithm now runs in phases and each phase has k iterations. In each iteration i, the SMCF Algorithm survivably routes $d(i)$ units of i^{th} commodity in sequence of steps along the path pair (p,q) with shortest primary path p. It updates

the primary and secondary length functions, $l^p(e)$ and $l^s(e,P)$ accordingly (similar to SMMF Algorithm).

SMCF Algorithm

Initialize $\forall e : l^p(e) = \delta_1/c(e)\ ; w = 0$
Initialize $\forall e \wedge \forall \mathcal{Q} \in \Psi_e \wedge \forall P \in \mathcal{Q} : l^s(e,P) = \delta_2/c(e)\ ; b = 0$
while $(D(l^p) < 1)$ **do**
  for $j = 1 \ldots r$ **do**
    $d'_j \leftarrow d_j$
    while $(D(l^p) < 1) \wedge (d'_j > 0)$ **do**
      select a path pair $P = (p,q) \in \Gamma_j$ where p is shortest
      $c \leftarrow min\{c(e) \mid e \in p \vee e \in q\} \cup \{d'_j\}$
      $d'_j \leftarrow d'_j - c$
      $w(P) \leftarrow w(P) + c$
      $\forall e \in p : l^p(e) \leftarrow l^p(e)(1 + \epsilon c/c(e))$
      for $\forall e \in q \wedge \mathcal{Q} \in \Psi_e\ s.t.\ P \in \mathcal{Q}$ **do**
        $b(e,\mathcal{Q}) \leftarrow max\{w(P), b(e,\mathcal{Q})\}$
        if $(w(P) > b(e,\mathcal{Q}))$ **then**
          $l^s(e,P) \leftarrow l^s(e,P)(1 + \epsilon(w(P) - b(e,\mathcal{Q}))/c(e))$
        end if
      end for
    end while
  end for
end while

Fig. 4. The outline of the SMCF Algorithm.

B. The Analysis

Let $l^p_{i,j-1}(e)$ and $l^s_{i,j-1}(e,P)$ be the primary and secondary length functions at the beginning of j^{th} iteration in the i^{th} phase. Let $l^p_{i,j-1,s-1}(e)$ and $l^s_{i,j-1,s-1}(e,P)$ be the primary and secondary length functions at the beginning of s^{th} step in the j^{th} iteration and i^{th} phase. In the j^{th} iteration d_j units of flow is routed along the shortest path given by primary length function l^p. Let $f_{i,j,s}(e)$ be the primary flow routed through edge e in the s^{th} step of j^{th} iteration and the i^{th} phase. The primary length of edge e is incremented as $l^p_{i,j-1,s}(e) = l^p_{i,j-1,s-1}(e)(1 + \epsilon f_{i,j,s}(e)/c(e))$. Let $b_{i,j-1,s}(e,\mathcal{Q})$ be the exclusive secondary flow pushed through edge e in the s^{th} step of j^{th} iteration and the i^{th} phase for protecting the primary flow along the path pair (p,q) in the set $\mathcal{Q}$. The secondary length function is incremented as follows, $l^s_{i,j-1,s}(e,P) = l^s_{i,j-1,s-1}(e,P)(1 + \epsilon b_{i,j-1,s}(e,\mathcal{Q})/c(e))$. Let $l^p_{i,j-1}$ ($l^p_{i,j-1,0}$) be the primary length function at the beginning of the j^{th} iteration of the i^{th} phase and let $l^p_{i,j}$ be the primary length function at the beginning of the $(j+1)^{th}$ iteration. Note that length function at the end of iteration i is same as the length function at the beginning of the next iteration. Since the primary lengths are monotonically increasing, $l^p_{i,j}(e) = l^p_{i,j-1}(e)(1 + \epsilon f_{i,j}(e)/c(e))$ where $f_{i,j}(e)$ is the primary flow routed through edge e in the j^{th} iteration of the i^{th} phase.

$$\begin{aligned} D(l^p_{i,j}) &= \sum_{e \in E} l^p_{i,j}(e)c(e) \\ &\leq D(l^p_{i,j-1}) + \epsilon d(j) dist_j(l^p_{i,j-1}) \end{aligned}$$

Let the primary length function at the end of k iterations of the j^{th} phase be $l^p_{i,k}$.

$$D(l^p_{i,k}) \leq D(l^p_{i,0}) + \epsilon \sum_{j=1}^{k} d(j) dist_j(l^p_{i,j-1})$$

The following two Lemmas from the analysis of the MCF Algorithm in paper [1] also hold true for the SMCF Algorithm.

Lemma 5: If $\beta \geq 1$ and the SMCF Algorithm terminates in the phase t when $D(l^p_{t,k}) \geq 1$ then $\frac{\beta}{(t-1)} \leq \frac{\epsilon}{(1-\epsilon)\ln\frac{1-\epsilon}{m\delta_1}}$.

Lemma 6: Let e be an edge and the SMCF Algorithm stops in the phase t when the stopping condition is satisfied then the primary flow through edge e is at most $c(e)\log_{1+\epsilon} 1/\delta_1$.

Lemma 7: Let e be an edge and the SMCF Algorithm stops in the phase t when the stopping condition is satisfied then the secondary flow through edge e for protecting primary flow along the path pairs in set $\mathcal{Q}$, $b_{t-1,k}(e,\mathcal{Q})$, is at most $c(e)k_1 \log_{1+\epsilon} 1/k_1\delta_2$.

Proof: Following the same lines of argument as in proof of Lemma 3, If $b_{t-1,k}(e,P)$ is the total exclusive secondary flow pushed through edge e for protecting the primary flow along the path pair $P \in \mathcal{Q}$, $\mathcal{Q} \in \Phi_e$ then

$l^s_{t-1,k}(e,P) \geq l^s_{0,0}(e,P)(1+\epsilon)^{\frac{b_{t-1,k}(e,P)}{c(e)}}$
$= \delta_2/c(e)(1+\epsilon)^{\frac{b_{t-1,k}(e,P)}{c(e)}}$.

Let us define $l^s(e,\mathcal{Q}) = \sum_{P\in\mathcal{Q}} l^s(e,P)$. Therefore,

$$\begin{aligned} l^s_{t-1,k}(e,\mathcal{Q}) &= \sum_{P\in\mathcal{Q}} l^s_{t-1,k}(e,P) \\ &\geq \delta_2/c(e) \sum_{P\in\mathcal{Q}} (1+\epsilon)^{\frac{b_{t-1,k}(e,\mathcal{Q},P)}{c(e)}} \\ &\geq \delta_2 k_1/c(e)(1+\epsilon)^{\frac{b_{t-1,k}(e,\mathcal{Q})}{k_1 c(e)}} \end{aligned}$$

From the proof of Lemma 6, we know that $l^p{}_{t-1,k}(e)$ is at most $1/c(e)$. Imposing the constraint of the dual problem that $l^p{}_{t-1,k}(e) \geq l^s(e,\mathcal{Q})$. Therefore $1/c(e) \geq \delta_2 k_1/c(e)(1+\epsilon)^{\frac{b_{t-1,k}(e,\mathcal{Q})}{k_1 c(e)}}$ leading to conclude that $b_{t-1,k}(e,\mathcal{Q})$ is at most $c(e)k_1 \log_{1+\epsilon} 1/k_1\delta_2$. ■

Lemma 8: The objective of the primal, λ, is at least $(t-1)/\left(\log_{1+\epsilon} 1/\delta_1 + k_1 k_2 \log_{1+\epsilon} 1/\delta_2 k_1\right)$.

Proof: From Lemma 6 and Lemma 7 the total amount of flow through link e is at most $c(e)\log_{1+\epsilon} 1/\delta_1 + c(e)k_1k_2\log_{1+\epsilon} 1/k_1\delta_2$ where k_2 is the maximum number of partitions of set Ψ_e.

Therefore scaling the flow by $\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2$ we get a feasible flow. ■

Theorem 3: The SMCF Algorithm is a $2(1+\omega)$ approximation algorithm for the SMCF problem.

Proof: Let γ be the ratio of the primal and dual solutions, then

$$\begin{aligned}\gamma &\leq \frac{\beta}{t-1}\left(\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2\right)\\ &\leq \frac{\epsilon}{(1-\epsilon)\ln\frac{1-\epsilon}{m\delta}}\left(\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2\right)\\ &= \frac{\epsilon}{(1-\epsilon)\ln(1+\epsilon)}\left(\frac{\ln 1/\delta_1}{\ln\frac{1-\epsilon}{m\delta}} + \frac{k_1k_2\ln 1/k_1\delta_2}{\ln\frac{1-\epsilon}{m\delta}}\right)\end{aligned}$$

If $\delta_1 = (m/(1-\epsilon))^{-1/\epsilon}$, we get $\frac{\ln 1/\delta_1}{\ln\frac{1-\epsilon}{m\delta_1}} = (1-\epsilon)^{-1}$. And if $\delta_2 = \frac{1}{k_1}\left(\frac{m\delta_1}{1-\epsilon}\right)^{\frac{1}{k_1k_2(1-\epsilon)}}$, we get $k_1k_2\frac{\ln 1/k_1\delta_2}{\ln\frac{1-\epsilon}{m\delta_1}} = (1-\epsilon)^{-1}$. Therefore,

$$\gamma \leq \frac{2\epsilon}{(1-\epsilon)^2\ln(1+\epsilon)} \leq 2(1-\epsilon)^{-3}$$

Let ϵ be chosen such that $1+\omega \leq (1-\epsilon)^{-3}$, then the SMCF algorithm is a $2(1+\omega)$ approximation algorithm. ■

C. Running Time

Theorem 4: If $\beta \geq 1$ and the SMCF Algorithm terminates after at most $1 + \frac{\beta}{\epsilon}\left(2\log_{1+\epsilon}\frac{m}{1-\epsilon} + \frac{\epsilon}{1-\epsilon}\log_{1+\epsilon}\frac{1-\epsilon}{m}\right)$ phases.

Proof: By the definition of γ in Theorem 3, we know,

$1 \leq \gamma \leq \frac{\beta}{t-1}\left(\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2\right)$. We get $t \leq 1+\beta\left(\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2\right)$.

Again from the definitions of, δ_1 and δ_2 in Theorem 3, we get, $\log_{1+\epsilon} 1/\delta_1 = \frac{1}{\epsilon}\log_{1+\epsilon}\frac{m}{1-\epsilon}$ and $k_1k_2\log_{1+\epsilon}\frac{1}{k_1\delta_2} = \frac{1}{1-\epsilon}\log_{1+\epsilon}\frac{1-\epsilon}{(m\delta_1)}$
$= \frac{1}{1-\epsilon}\left(\log_{1+\epsilon}\frac{1-\epsilon}{m} + \log_{1+\epsilon}\frac{1}{\delta_1}\right)$. Therefore,

$$\begin{aligned}t &\leq 1+\beta\left(\log_{1+\epsilon} 1/\delta_1 + k_1k_2\log_{1+\epsilon} 1/k_1\delta_2\right)\\ &\leq 1+\frac{\beta}{\epsilon}\left(\log_{1+\epsilon}\frac{m}{1-\epsilon} + \frac{\epsilon}{1-\epsilon}\log_{1+\epsilon}\frac{1-\epsilon}{m} + \frac{\epsilon}{1-\epsilon}\log_{1+\epsilon}\frac{1}{\delta_1}\right)\\ &< 1+\frac{\beta}{\epsilon}\left(2\log_{1+\epsilon}\frac{m}{1-\epsilon} + \frac{\epsilon}{1-\epsilon}\log_{1+\epsilon}\frac{1-\epsilon}{m}\right)\end{aligned}$$

Hence proved. ■

From Theorem 4, note that the number of phases of the SMCF Algorithm depends on β. Therefore run-time of the SMCF Algorithm depends on β. Following the same enhancements discussed in [1] using the technique proposed in [30] to eliminate the dependence of the run-time on β, we get the number of phases of the SMCF Algorithm is $T\log_2 k$ phases where

$$\begin{aligned}T = &\ 2\left\lceil\frac{1}{\epsilon}\left(2\log_{1+\epsilon}\frac{m}{1-\epsilon} + \frac{\epsilon}{1-\epsilon}\log_{1+\epsilon}\frac{1-\epsilon}{m}\right)\right\rceil\\ = &\ 2\left\lceil\frac{1}{\epsilon}\left(\log_{1+\epsilon}\left(\frac{m}{1-\epsilon}\right)^2 + \log_{1+\epsilon}\left(\frac{1-\epsilon}{m}\right)^{\frac{\epsilon}{1-\epsilon}}\right)\right\rceil\\ = &\ 2\left\lceil\frac{1}{\epsilon}\left(\log_{1+\epsilon}\left(\frac{m}{1-\epsilon}\right)^{\frac{-2\epsilon}{1-\epsilon}}\right)\right\rceil\\ = &\ 4\left\lceil\log_{1+\epsilon}\left(\frac{m}{1-\epsilon}\right)\right\rceil\end{aligned}$$

VI. Minimum cost Survivable Maximum Concurrent Flow (MC-SMCF)

In this section, we present an approximation algorithm for the minimum cost survivable maximum concurrent flow (MC-SMCF) problem. Given a graph $G=(V,E)$ with edge capacities $c: E \rightarrow R^+$, edge costs $s: E \rightarrow R^+$, k node pairs (s_i,t_i) and a bound B. The problem is to find the largest λ such that there are at least $\lambda d(i)$ units of survivable flow for each commodity i under shared protection and the cost of the flow is no more than B. The primal formulation $\mathcal{P}_{MC-SMCF}$ of the MC-SMCF problem is very similar to the primal of the SMCF problem with the additional constraint that the cost of the survivable multicommodity flow must be at most B. The primal follows.

Maximize λ

$$\forall e \in E: \sum_{P=(p,q):e\in p} w(P) + \sum_{\mathcal{Q}\in\Psi_e} b(e,\mathcal{Q}) \leq c(e)$$

$$\sum_{\forall e\wedge P: e\in p} w(P)s(e) + \sum_{\forall e\wedge\mathcal{Q}\in\Psi_e} b(e,\mathcal{Q})s(e) \leq B$$

$$\forall j: \sum_{P\in\Gamma_j} w(P) \geq \lambda d(j)$$

$$\forall e \in E \wedge \forall P \in \mathcal{Q}\ \text{s.t.}\mathcal{Q} \in \Psi_e: w(P) - b(e,\mathcal{Q}) \leq 0$$

$$\forall P \in \Gamma: w(P) \geq 0$$

$$\forall e \in E, \forall \mathcal{Q} \in \Psi_e: b(e,\mathcal{Q}) \geq 0; \lambda \geq 0$$

The dual formulation $\mathcal{D}_{MC-SMCF}$ of the MC-SMCF problem is again similar to the dual of the SMCF problem except that we introduce a pseudo-edge with capacity B. As before let $l^p(e)$ and $l^s(e)$ be the primary and secondary length functions of edges in G respectively and ϕ be the

length of the pseudo edge.

$$\text{Minimize} \sum_{e \in E} l^p(e)c(e) + \phi B$$

$$\forall P \in \Gamma : \sum_{e \in p} [l^p(e) + \phi s(e)] + \sum_{e \in q \wedge \mathcal{Q} \in \Psi_e, s.t. P \in \mathcal{Q}} l^s(e, P) \geq z_j$$

$$\forall e \in E \wedge \forall \mathcal{Q} \in \Psi_e : l^p(e) + \phi s(e) - \sum_{P \in \mathcal{Q}} l^s(e, P) \geq 0$$

$$\sum_j z_j d_j \geq 1$$

$$\forall e \in E : l^p(e) \geq 0$$

$$\forall e \in E \wedge \forall P \in \mathcal{Q} \, \text{s.t.} \mathcal{Q} \in \Psi_e : l^s(e, P) \geq 0$$

$$\forall j : z_j \geq 0; \phi \geq 0$$

Let $D(l^p, \phi) = \sum_{e \in E} l^p(e)c(e) + \phi B$. The cost of a path pair P, $C(P) = \sum_{e \in p} [l^p(e) + \phi s(e)] + \sum_{e \in q} l^s(e, P)$. Let $P_j^{min}(l^p + \phi, l^s)$ be the path pair $P \in \Gamma_j$ with minimum cost and let $\gamma(l^p + \phi, l^s)$ be the sum of the costs of pushing $d(j)$ units of flow for j^{th} commodity along the path pair $P_j^{min}(l^p + \phi, l^s)$, for all commodities. Therefore $\gamma(l^p, l^s) = \sum_j d(j)C(P^{min}(l^p, l^s))$. Let β be the optimal objective value, therefore $\beta \leq D(l^p)/\gamma(l^p, l^s)$. Let $\gamma_p(l^p)$ be the sum of the costs of pushing $d(j)$ units of flow for along the primary path of the path pair $P^{min}(l^p, l^s)$, for all commodities. Therefore $\gamma(l^p + \phi, l^s) > \gamma_p(l^p + \phi)$. Hence $\beta \leq D(l^p + \phi)/\gamma(l^p + \phi, l^s) < D(l^p + \phi)/\gamma_p(l^p + \phi)$. Let $\alpha(l^p + \phi)$ be the sum of costs of pushing $d(j)$ units of flow along the shortest primary path among the path pairs $P \in \Gamma_i$. Therefore $\alpha(l^p + \phi) \leq \gamma_p(l^p + \phi)$. And hence $\beta < D(l^p + \phi)/\gamma_p(l^p + \phi) \leq D(l^p + \phi)/\alpha(l^p + \phi)$.

A. The Algorithm and its Analysis

The MC-SMCF Algorithm is outlined in Figure 5. Let $l^p_{i,j-1}(e)$ and $l^s_{i,j-1}(e, P)$ be the primary and secondary length functions and $\phi_{i,j-1}$ be the length function of the pseudo edge at the beginning of j^{th} iteration in the i^{th} phase. Similar to the SMCF Algorithm, the d_j units of flow is routed in j^{th} iteration of any phase along the shortest path given by primary length function l^p. After each step in the iteration the length functions along the primary and secondary paths of the path pair are incremented. Let $l^p_{i,j-1,s-1}(e)$ and $l^s_{i,j-1,s-1}(e, P)$ be the primary and secondary length functions and $\phi_{i,j-1,s-1}$ be the length function of the pseudo edge before the s^{th} step in the j^{th} iteration of the i^{th} phase. Let $f_{i,j,s}(e)$ be the primary flow routed through the edge e in the s^{th} step of the j^{th} iteration and i^{th} phase. The primary length of edge e is incremented as $l^p_{i,j-1,s}(e) = l^p_{i,j-1,s-1}(e)(1 + \epsilon f_{i,j,s}(e)/c(e))$ and pseudo length function is incremented as $\phi_{i,j-1,s} = \phi_{i,j-1,s-1}(1 + \epsilon B_{i,j}/B)$. Let $b_{i,j-1,s}(e, \mathcal{Q})$ be the exclusive secondary flow pushed through edge e for protecting the primary flow along the path pairs in the set $\mathcal{Q}$. The secondary length function is incremented as follows, $l^s_{i,j-1,s}(e, P) = l^s_{i,j-1,s-1}(e, P)(1 + \epsilon b_{i,j-1,s}(e, \mathcal{Q})/c(e))$.

MC-SMCF Algorithm

1: Initialize $\forall e : l^p(e) = \delta_1/c(e)$; $\phi = \delta_1/B$; $w = 0$
2: Initialize $\forall e \wedge \forall \mathcal{Q} \in \Psi_e \wedge \forall P \in \mathcal{Q} : l^s(e, P) = \delta_2/c(e)$; $b = 0$
3: **while** $(D(l^p, \phi) < 1)$ **do**
4: **for** $j = 1 \ldots r$ **do**
5: $d'_j \leftarrow d_j$
6: **while** $(D(l^p, \phi) < 1) \wedge (d'_j > 0)$ **do**
7: select a path pair $P = (p, q) \in \Gamma_j$ with least cost
8: $c \leftarrow min\{c(e) \mid e \in p \vee e \in q\} \cup \{d'_j\}$
9: $K \leftarrow \sum_{e \in q} s(e) min\{c, (b(e, Q) - w(P))\}$
10: $B_j \leftarrow c \sum_{e \in p \vee e \in q} s(e) - K$
11: **if** $B_j > B$ **then**
12: $c \leftarrow c \frac{B}{B(j)+K}$
13: $K \leftarrow \sum_{e \in q} s(e) min\{c, (b(e, Q) - w(P))\}$
14: $B_j \leftarrow B - K$
15: **end if**
16: $d'_j \leftarrow d'_j - c$
17: $w(P) \leftarrow w(P) + c$
18: $\forall e \in p : l^p(e) \leftarrow l^p(e)(1 + \epsilon c/c(e))$
19: $\phi \leftarrow \phi(1 + \epsilon B_j/B)$
20: **for** $\forall e \in q \wedge \mathcal{Q} \in \Psi_e \; s.t. \; P \in \mathcal{Q}$ **do**
21: $b(e, \mathcal{Q}) \leftarrow max\{w(P), b(e, \mathcal{Q})\}$
22: **if** $(w(P) > b(e, \mathcal{Q}))$ **then**
23: $l^s(e, P) \leftarrow l^s(e, P)(1 + \epsilon(w(P) - b(e, \mathcal{Q})/c(e)))$
24: **end if**
25: **end for**
26: **end while**
27: **end for**
28: **end while**

Fig. 5. The outline of the MC-SMCF Algorithm.

Based on the analysis of the MC-MCF Algorithm in [1] and then following a similar analysis as in Section V-B leads to the equivalent approximation guarantee for the MC-SMCF Algorithm.

Theorem 5: The MC-SMCF Algorithm is a $2(1 - \epsilon)^{-3}$ approximation algorithm for the MC-SMCF Problem.

VII. Integrality

A multicommodity flow has integrality f_g if the flow of every commodity on every edge is a non-negative integer multiple of f_g. A multicommodity flow problem formulated using arc-chain linear programming

formulation has integrality f_g if the flow of every commodity on every path considered for that commodity is a non-negative integer multiple of f_g. In this section, we address the consequences of such integrality constraints on the three survivable MCF problems. We also explore the applications of the SMMF and SMCF problems to the problem of survivable routing in capacitated networks.

A. The SMMF Problem

The SMMF Algorithm can be modified to compute an approximation of the fractional multicommodity flow computed with an integrality. Instead of pushing c units of flow, if we push q units of flow where $q \leq c(e)$ for all $e \in E$. Let call this algorithm the SMMF-I Algorithm computes a multicommodity flow of integrality $\frac{q\epsilon}{\log_{1+\epsilon} n}$ [1]. The following Theorem is proved in [1].

Theorem 6: If all the edges in the graph have capacity at least $\frac{\log_{1+\epsilon} n}{\epsilon}$ then there exists a $2(1+\omega)$-approximation to integral maximum multicommodity flow.

B. The SMCF Problem

In this section we discuss the modification suggested in [1] to compute the integrality of the concurrent flow problem. Instead of routing c units of flow computed in Step 8 of the SMCF Algorithm we route q units where q is selected such that q is less than the capacity of the minimum capacity edge in G and all the demands can be expressed as a non-negative integral multiple of q. The primary and secondary length functions are incremented corresponding to the q units of flow routed. Let us call this modified algorithm the SMCF-I Algorithm. The approximation analysis of the SMCF-I Algorithm is similar to the SMCF Algorithm. However, it has worse computational complexity. The number of phases for an instance of the SMCF problem is same for the SMCF and SMCF-I Algorithms. However the increase in the complexity is due to the fact that the number of steps in a iteration of any phase of the SMCF-I Algorithm increases depending upon the value of q. The proof of the following Theorem follows directly from the proof of the integrality of the MCF Algorithm in [1].

Theorem 7: If q is selected such that all the demands $d(i)$ are integral multiples of q and is at most the capacity of the minimum capacity link in G then the SMCF-I Algorithm computes a $2(1-\epsilon)^{-3}$-approximation to the maximum concurrent flow and has integrality $\frac{q\epsilon}{\log_{1+\epsilon} m/(1-\epsilon)}$.

Corollary 1: If all the edges in the graph have capacity at least $\frac{\log_{1+\epsilon} m/(1-\epsilon)}{\epsilon}$ and all demands are integral multiples of $\frac{\log_{1+\epsilon} m/(1-\epsilon)}{\epsilon}$ then there exists a $2(1+3\epsilon)$-approximation algorithm to integral maximum concurrent flow problem.

C. Applications to Survivable Routing in Capacitated Networks

Let us model the survivable pre-provisioning of static routing in capacitated networks as a multicommodity flow problem. We are given a capacitated network and a set of k path pairs corresponding to k commodities. The problem to maximize the sum of the survivable multicommodity flow. This problem can be modeled as an SMMF problem. If the flow is not required to be integral then the SMMF Algorithm can be used to compute a $2(1+3\epsilon)$ approximate solution. If the multicommodity flow is required to be integral, then ϵ can be selected such that the capacity of the minimum capacity edge is at least $\frac{\log_{1+\epsilon} n}{\epsilon}$. Then the SMMF-I Algorithm can be used to compute a $2(1+3\epsilon)$-approximate integral multicommodity flow. Alternately, a lower bound on the minimum capacity edge in the graph is $\frac{\log_{1+\epsilon} n}{\epsilon}$ for $2(1+3\epsilon)$-approximate integral multicommodity flow.

We are given a capacitated network and traffic demands between a set of path pairs. The problem is to survivably provision a maximum fraction λ of all the demands. The problem can be modeled as an SMCF problem. The capacitated network is represented as a graph $G = (V, E)$ with capacities $c(e)$ for each edge $e \in E$. The traffic demand as $d(i)$ between node pair (s_i, t_i) as i^{th} commodity. If the capacities and demands are real numbers the SMCF Algorithm can be directly used to compute approximate solution. The SMCF-I Algorithm can be used to compute an $2(1+3\epsilon)$ approximate solution with integrality $\frac{q\epsilon}{\log_{1+\epsilon} m/(1-\epsilon)}$.

VIII. Conclusions

In this work we formulate and study the fundamental problems used to model and solve survivable routing in multi-fiber wavelength division multiplexing network design and capacitated network design. We define a survivable flow under shared protection in a directed graph. We formulate survivable versions of multicommodity flow (MF) problems, namely, survivable maximum multicommodity flow (SMMF), survivable maximum concurrent flow (SMCF) and minimum cost maximum survivable concurrent flow (MC-SMCF) problems. Assuming that the choice of paths that are used to push a flow between any node pair in the graph is limited, we present approximation algorithms for the fractional SMF problems.

We modeled the survivable routing problem in capacitated networks as an SMF problem. We discuss the modifications to the approximation algorithms to the fractional SMF problems presented in this paper to solve the problem of survivable routing in capacitated networks.

REFERENCES

[1] N. Garg and J. Konemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," In 39^{th} annual IEEE Symposium on foundations of computer science, pp. 300-309, 1998.

[2] L. K. Fleischer, "Approximating fractional multicommodity flows independent of the number of commodities," SIAM J. Discrete Math. vol. 13, no. 4, pp. 505-520, 2000.

[3] M. J. Fischer, D. A. Garbin, T. C. Harris, and J. E. Knepley, "Large scale communications networks design and analysis," *Int. J. Manage. Sci.*, vol. 6, pp. 331-340, 1978.

[4] J. E. Knepley, "Minimum cost design for circuit switched networks," Defense Commun. Agency Syst., Eng. Facility, Reston, VA, Tech. Rep. AD-A014 101, July 1973.

[5] R. R. Srikanta Kumar, "On optimal flows in circuit switched networks," in Proc. Globecom '84, pp.27.3.1-27.3.5, 1984.

[6] B. Sanso, M. Gendreau, and F. Soumis, "An algorithm for network dimensioning in under reliability considerations," *Ann. Operations Res.*, vol. 36, pp. 263-274, 1992.

[7] A.Girard, and B. Sanso, "Multicommodity flow models, failure propagation, and reliable loss network design," *IEEE/ACM Transactions on Networking*, vol. 6, no. 1, Feb. 1998.

[8] R. M. Krishnaswamy and K. N. Sivarajan, "Algorithms for routing and wavelength assignment and topology design in optical network," Ph. D. dissertation, Dept. of Electrical Commun. End., Indian Institute of Science, Banglore, India, Nov. 1998.

[9] R. M. Krishnaswamy and K. N. Sivarajan, "Algorithms for routing and wavelength assignment based on solutions of LP-relaxations," *IEEE Communication Letters*, vol. 5, no. 10, pp. 435-437, Oct. 2001.

[10] P. Raghavan, "Probabilistic construction of deterministic algorithm: Approximating packing integer programs," *Journal of Computer and Systems Sciences*, vol. 38, pp. 683-707, 1994.

[11] D. Coudert and H. Rivano, "Lightpath assignment for multifibers WDM optical networks with wavelength translators," In IEEE Globecom'02, vol. 3, pp. 2686 - 2690, Taiwan, 17-21 Nov. 2002.

[12] M. Bouklit, D. Coudert, J-F. Lalande, C. Paul, and H. Rivano, "Approximate multicommodity flow for WDM networks design," In Proceedings of the 10th International Colloquium on Structural Information and Communication Complexity (SIROCCO '03), Carleton Scientific 17, pp. 43-56, 2003.

[13] A. Ferreira, S. Prennes, A. W. Richa, H. Rivano, and N. Stier Moses, "Models, complexity and algorithms for the design of multifiber WDM networks," In ICT'03, Papeete, French Polynesia, February 2003.

[14] O. Crochat, and J. -Y. Le Boudec, "Design protection for WDM optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1158-1165, Sept. 1998 .

[15] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Networks Part I: Protection," in Proc. IEEE INFOCOM, pp. 744-751, Mar. 1999.

[16] Y. Miyao and H. Saito,"Optimal design and evaluation of survivable WDM transport networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 11901198, Sept. 1998.

[17] B. V. Caenegem, W. V. Parys, F. De Turck, and P. M. Demeester, "Dimensioning of survivable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 11461157, Sept. 1998.

[18] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration," Bell Labs Tech. J., pp. 5883, Jan.Mar. 1999.

[19] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," in IEEE INFOCOM, vol. 2, pp. 910918, Mar. 1998.

[20] R. R. Iraschko and W. D. Grover, "A highly efficient path-restoration protocol for management of optical network transport integrity," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 5, pp. 779-794, Oct. 2000.

[21] X. Yang , L. Shen and B. Ramamurthy, "Survivable lightpath provisioning in WDM mesh networks under shared path protection and signal quality constraints," in *IEEE/OSA Journal of Lightwave Technology*, Apr. 2005.

[22] D. Bienstock, O. Gunluk, "Capacitated network design - polyhedral structure and computation," *INFORMS Journal on Computing*, vol. 8, pp. 243-259, 1996.

[23] O. Gunluk, "A branch-and-cut algorithm for capacitated network design problems," *Math. Programming*, vol. 86, pp. 17-39, 1999.

[24] B. Gendron, T. G. Crainic and A. Frangioni, "Multicommodity capacitated network design," In B. Sans'o and P. Soriano (eds), Telecommunications Network Planning, pp. 1-19, Kluwer, Norwell, MA, 1998.

[25] M. Alicherry, R. Bhatia and Y.-C. Wan, "Designing networks with existing traffic to support fast restoration." APPROX, 2004.

[26] M. Alicherry and R. Bhatia, "Pre-provisioning networks to support fast restoration with minimum over-build," INFOCOM, 2004

[27] H. A. Taha, "Operations research: An introduction," Third Edition, MacMillan Publishing Co., Inc., New York, 1982.

[28] F. Shahrokhi and D. Matula, "The maximum concurrent flow problem," J. ACM, vol. 37, no. 2, pp. 318-334, 1990.

[29] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, "Network flows: Theory, Algorithms, and Applications," Prentice Hall, New Jersey, 1993.

[30] S. A. Plotkin, D. Shmoys, and E. Tardos,"Fast approximation algorithms for fractional packing and covering problems," *Mathematics of Operations Research*, vol. 20, pp. 257-301, 1995.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.