**Final version appears in Networks, 25 (1995), pp. 19-30.**

# Multicommodity Flows in Simple Multistage Networks

E.S. Elmallah and J. Culberson
Department of Computing Science
University of Alberta, Edmonton, Alberta

**Abstract**

This paper considers the integral multicommodity flow problem on directed graphs underlying two classes of multistage interconnection networks. In one direction, we consider 3-stage networks. Using existing results on $(g, f)$-factors of bipartite graphs, we show sufficient and necessary conditions for the existence of a solution when the network has at most 2 secondary switches. In contrast, the problem is shown to be NP-complete if the network has 3 or more secondaries. In a second direction, we introduce a recursive class of networks that includes multistage hypercubic networks (such as the omega network, the indirect binary n-cube, and the generalized cube network) as a proper subset. Networks in the new class may have arbitrary number of stages, moreover, each stage may contain identical switches of any arbitrary size. The notion of extra-stage networks is extended to the new class, and the problem is shown to have polynomial time solutions on $r$-stage networks where $r = 3$, or $r \geq 3$ and each link has a unit capacity. The latter result implies an efficient algorithm for deciding admissible permutations on conventional extra-stage hypercubic networks. In contrast, we show that the multicommodity flow problem is NP-complete on extra-stage networks, even if $r = 6$, each link has an integral capacity $\leq 3$, and all flow demands are equal.

## 1 Introduction

Multicommodity flow problems arise naturally in the analysis of circuit-switching communication networks where a collection of end-to-end connections are to be routed simultaneously. With some connections already in progress at some instant, the state of the network is captured by a graph $G = (V, E)$, where each communication link $e \in E$ has a positive residual capacity $c(e)$. The required connections are specified by a set of pairs of vertices $\{(s_i, t_i) : 1 \leq i \leq k\}$, with connection $(s_i, t_i)$ requiring $q(s_i, t_i)$ ($q_i$ for short) units of bandwidth. In the integral multicommodity flow (*MCF*) problem, $c$ and $q$ are integer-valued. The problem is to compute a set of flow functions $\{f_i : E \to Z_0^+ : 1 \leq i \leq k\}$ such that (a) for each $e \in E$, $\sum f_i(e) \leq c(e)$, (b) for each vertex $v \notin \{s_i, t_i\}$, flow $f_i$ is conserved at $v$, and (c) for each $i$, $1 \leq i \leq k$, the net flow into $t_i$ under $f_i$ is at least $q_i$.

Our objective in this paper is to investigate the complexity of the problem on directed graphs underlying certain classes of multistage interconnection networks (*MINs*), commonly constructed from arrays of space-division or time-division multiplexers (for example, crossbar switches). An $r$-stage MIN then corresponds to a directed graph $G$ with stages $(V_1, V_2, \cdots, V_r)$ of vertices, and links connecting vertices in one stage to vertices in the next stage; the required connections form a subset of $V_1 \times V_r$. Networks of this type have been studied extensively in the context of designing telephone switching systems, communication networks, as well as high performance parallel and distributed multiprocessor systems (see, for example, [3, 8, 13, 23, 26]). In particular, numerous interesting results (to mention a few, [2, 14, 17, 18, 21, 25]) exist on special cases of the problem where $c(e) = 1$ for each link, and the traffic is restricted to permutation connections or broadcast connections. In contrast, not as much seems to be known about the general problem on various classes of MINs. In this paper we aim at bridging the above gap by considering the problem on 3-stage networks (sections 2 and 3), and another hierarchy of classes of networks that includes extra-stage hypercubic networks (examples of standard hypercubic networks include the omega network [15], the generalized cube network [22], and the indirect binary n-cube [21]) as special subclasses (sections 4, 5, and 6).

Before proceeding further, it is perhaps worthwhile reviewing some general results that are valuable in directing the search for efficient exact algorithms for the MCF problem. From a complexity point of view, Even et al. [6] have shown that the 2-commodity MCF problem is NP-complete, even if each edge has a unit capacity. Although the result serves well in pruning a nontrivial portion of the search space, it is based on constructing directed graphs with relatively large diameters, and with multiple paths between the terminals of interest. In contrast, the classes of networks considered here have either small diameters, or regular structure that offer only two paths between any pair of terminals (the paths may share an initial or a terminal segment, but otherwise link-disjoint).

A second result of interest, due to Okamura and Seymour [19], deals with planar graphs where the terminals of interest lie on a common face of the graph. With $X \subseteq V$, $\delta(X)$ denoting the set of edges with one end point in $X$ and the other in $V - X$, $D(X) = \{i : |\{s_i, t_i\} \cap X| = 1\}$, and $c, q$ are integer-valued, [19] shows that integral flows exist if and only if for every subset $X \subseteq V$, $\sum_{e \in \delta(X)} c(e) - \sum_{i \in D(X)} q_i$ $(\geq 0)$ is an even integer. This implies integral solutions if $c$ and $q$ are even-integer-valued, and each cut's capacity is at least as large as the cut's demand. Hassin [10] gives efficient procedures for constructing such a flow if one exists. Conversely, if no feasible solution exists, the argument identifies a violating set $X$ of vertices. Finding such a set $X$ is useful in cases where additional bandwidth can be allocated dynamically to increase link capacities to accommodate a given workload. Analogous to the above cut-demand conditions, we examine in sections 3 and 5 the sufficiency of the following set of conditions for directed multistage networks:

**Set [CD]:** For every subset $X \subseteq V_1$ and $Y \subseteq V_r$: $q(X,Y) \geq \delta(X,Y)$, where $q(X,Y) = \sum_{(s_i,t_i) \in X \times Y} q_i$, and $\delta(X,Y)$ is the minimum capacity of a directed cut separating $X$ from $Y$.

## 2 3-stage Networks

Networks in this class have a simple structure that is often used as the basis of constructing other networks with various properties. Thus, providing a good starting point for analysis. Here, the network is modeled by a tripartite graph $G = (X \cup Y \cup Z, E_{XY} \cup E_{YZ})$, on a set $X$ of primary switches, a set $Y$ of secondary switches, and a set $Z$ of tertiary switches. Each switch corresponds to an independent subnetwork that is capable of switching all incoming traffic, regardless of its volume or distribution. Common graph terminology is used and will be introduced as required. To start, we need the following few notations. If $S$ is a subset of vertices or edges of a graph $G$, then $G[S]$ denotes the subgraph of $G$ induced by $S$. In addition, if $f$ is a real function on $S$, we let $f(S) = \sum_{x \in S} f(x)$. Well known terminology of switching networks is also used. Thus, a network with $N$ inputs (outputs) is rearrangeable if it admits all $N!$ permutation connections between the inputs and outputs.

Clos (see, for example, [3]) pioneered the discovery of a class of 3-stage rearrangeable networks. In its general form, a Clos network is constructed from rearrangeable switches such that each primary is connected to every secondary, and each secondary is connected to every tertiary. Thus, $G[E_{XY}]$ and $G[E_{YZ}]$ are complete bipartite graphs. When routing permutations is of concern, it is well known that Clos's result can be derived from Vizing edge-coloring theorem (e.g., see [4] or [5]). Specifically, for a given permutation to be routed, consider the bipartite graph $G_D = (X \cup Z, E_D)$ of flow demands, where $(x, z) \in E_D$ if $q(x, z) = 1$. The maximum degree of any vertex in $G_D$ is $|Y|$, hence, by Vizing theorem, $E_D$ can be properly colored using at most $|Y|$ colors. Thus, all demands can be assigned to the set $Y$ of secondaries without conflicts, proving that the network is rearrangeable. Fast parallel algorithms for routing in networks that can be derived recursively from 3-stage networks appear in [16].

At an arbitrary instant, however, some links of the network may be busy serving connections already in progress, and it is no longer true that all of the $|Y|$ secondaries are accessible from each primary or tertiary. Consequently, all of the above techniques are potentially inapplicable. Nevertheless, one may anticipate that minor topological variations of the fully connected Clos network may give rise to routing problems that can be solved efficiently. In Theorem 2.1 we support the above intuitive remark for two restricted cases. Furthermore, the result shows that conditions in [CD] are both necessary and sufficient for the existence of a solution. The proof uses an elegant characterization of sufficient and necessary conditions for solving the following problem. Given two integer-valued functions $f$ and $g$ on the vertex set of a bipartite graph $G$ such that $0 \leq g(x) \leq f(x) \leq deg_G(x)$, for each $x$ in $G$. Find a $(g, f)$-factor of $G$; that is, a spanning subgraph $F$ such that $g(x) \leq deg_F(x) \leq f(x)$ for each vertex $x$ in $G$. Using the new notation $a \dot{-} b = max(0, a - b)$ employed in [11], the result can be stated as follows.

**Theorem HHKL** [11]: A bipartite graph $G$ has a $(g, f)$-factor if and only if for every set $S$ of vertices $\sum_{x \notin S} g(x) \dot{-} deg_{G-S}(x) \leq f(S)$.

Efficient algorithms for solving several factoring problems appear in [12] and the references therein. Our proof applies the above result on the demand graph $G_D = (X \cup Z, E_D)$, having $q(x, z)$ parallel edges between $x$ and $z$ for each $q(x, z) > 0$.

**Theorem** 2.1: Let $G = (X \cup Y \cup Z, E_{XY} \cup E_{YZ})$ be a 3-stage network satisfying one of the following conditions. Then the MCF problem ($c$ and $q$ are as in section 1) has a solution if and only if conditions in [CD] hold.

1. $|Y| \leq 2$.
2. $|X| \leq 2$ and $G[E_{YZ}]$ is a complete bipartite graph (by symmetry, $|Z| \leq 2$ and $G[E_{XY}]$ is a complete bipartite graph), and $c$ is constant.

Proof of 1:

Part 1 follows easily if $|Y| = 1$, so let $Y = \{a, b\}$. Note that the MCF problem has a solution if and only if $G_D$ has a $(g, f)$-factor $G_a$ (corresponding to demands routed through the secondary $a$), where $f(v) = c(v, a)$, and $g(v) = deg_{G_D}(v) - c(v, b)$ for every $v \in G_D$. So that the remaining demands form a second bipartite graph $G_b$ where $deg_{G_b}(v) \leq c(v, b)$ for every vertex $v$. To derive a contradiction, let $(G, c, q)$ be a counter instance of the MCF problem with the fewest number of vertices. That is, $(G, c, q)$ satisfies conditions [CD], yet there is no feasible solution. In $G$, every primary or tertiary $v$ is connected to both secondaries $a$ and $b$. Otherwise, one can route all demands of $v$ via its unique adjacent secondary, and subsequently reduce the capacities, and delete $v$. The resulting instance forms a smaller counterexample.

By Theorem [HHKL], there exits a *violating* set $S$ of minimum cardinality such that $\sum_{v \notin S} g(v) \dot{-} deg_{G_D - S}(v) > f(S)$. The cardinality of $S$ implies that it is a subset of either $X$ or $Z$, but not both. Without loss of generality, assume that $S \subseteq X$. It is then easy to verify that the vertices of $X - S$ have zero contribution to the summation on the left hand-side. In contrast, let $Z' \subseteq Z$ be the set $\{z : \; g(z) \dot{-} deg_{G_D - S}(z) > 0\}$ that contributes positively to the sum. With $z \in Z'$ and $\beta_z$ denoting the flow requirements $q(X - S, z) = \sum_{x \in X - S} q(x, z)$, the violating inequality can be written as:

$$\sum_{z \in Z'} (deg_{G_D}(z) - c(z, b) - \beta_z) > f(S) = \sum_{x \in S} c(x, a).$$

Equivalently,

$$\sum_{z \in Z'} (deg_{G_D}(z) - \beta_z) > \sum_{x \in S} c(x, a) + \sum_{z \in Z'} c(z, b).$$

The left hand-side of the above inequality corresponds to $q(S, Z')$, and the right hand-side corresponds to the capacity of an $(S, Z')$-cut, a contradiction. ■

Proof of 2:

The claim follows easily if $X$ has one primary. So, assume that $X = \{x_1, x_2\}$. By the above proof, it suffices to show a flow equivalent instance $(G', c', q')$ with exactly 2 secondaries, denoted $y_s$ and $y_d$. With $N(x)$ denoting the neighbors of an arbitrary vertex $x$, let $Y_{1,2} = N(x_1) \cap N(x_2)$, $Y_1 = N(x_1) \setminus N(x_2)$, and $Y_2 = N(x_2) \setminus N(x_1)$. In the new instance, $X' = X$, $Y' = \{y_s, y_d\}$, $Z' = Z$, and $q' = q$. In addition, for each vertex $x_i \in X$ set $c'(x_i, y_d) = c(x_i, Y_i)$, and $c'(x_i, y_s) = c(x_i, Y_{1,2})$. Moreover, for each vertex $z_i \in Z$ set $c'(z_i, y_d) = c(z_i, Y_1 \cup Y_2)$, and $c'(z_i, y_s) = c(z_i, Y_{1,2})$. Since $c$ is constant and $E_{YZ}$ is a complete bipartite graph, it then follows that $G$ admits the demands of $q$ if and only if $G'$ admits $q'$. ■

# 3 NP-completeness

Theorem 2.1(1) has an application in section 6; we now show that it is a tractable problem at the edge of an intractability cliff.

**Theorem** 3.1: The MCF problem is NP-complete on 3-stage networks with 3 secondaries, and unit-valued functions $c$ and $q$.

In the special case (where $c = 1$), the MCF problem is equivalent to the following restricted edge-coloring problem on bipartite graphs, denoted $BREC$ thereafter. Given a bipartite graph $G = (A \cup B, E)$, a set $L$ of colors, and for each vertex $x$ a subset $L(x)$ of permissible colors. Does there exist a coloring function $color : E \rightarrow L$ such that for each edge $xy \in E$, $color(xy) \in L(x) \cap L(y)$, and no two edges incident with the same vertex have the same color?

The reduction is from the Not-All-Equal 3-SAT (problem [LO3], page 259, [7]) where $U$ is a set of variables and $C$ is a collection of clauses over $U$ such that each clause $c \in C$ has $|c| = 3$. We ask whether there exists a truth assignment for $U$ such that each clause in $C$ has at least one true literal and at least one false literal. To show Theorem 3.1 above, it suffices to show the following.

**Lemma** 3.1: N-A-E 3-SAT $\propto$ BREC with exactly 3 colors.

*Proof:*

Let $(U, C)$ be an instance of the N-A-E 3-SAT, we construct an instance $(G, L)$ of the BREC problem where $G$ has $|U|$ *variable-gadgets* and $|C|$ *clause-gadgets*, and $L = \{n, t, f\}$. Figure 3.1 illustrates the caterpillar-like structure of a typical variable-gadget $G_i$ associated with variable $u_i$. The head edge $h_i$ of the gadget has two permissible colors: $t$ and $f$. The outlets of the gadget are the leaves labeled $u_{i,j}$, $j \geq 0$. An outlet $u_{i,j}$ is called *odd* or *even* depending on the index $j$. If literal $u_i$ appears $p$ times, and literal $\overline{u_i}$ appears $q$ times in $C$, then the gadget is extended to supply $2\max(p, q)$ outlets.

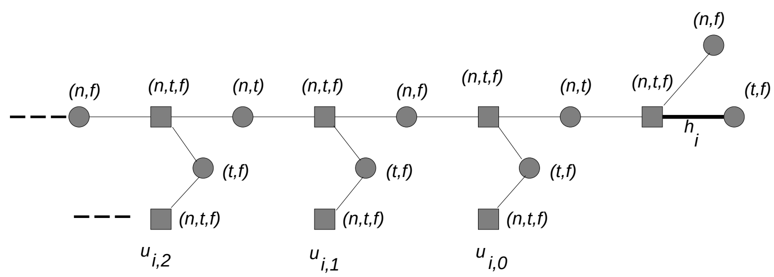



Figure 3.1. A variable gadget $G_i$.

The gadget satisfies the following property: if $h_i$ is colored $x \in \{t, f\}$ then all edges incident with the even outlets have the same color $x$, and all edges incident with the odd outlets have the complement color $\overline{x}$.

In addition, for each clause $C_j \in C$, $G$ has a vertex $C_j$ with $L(C_j) = \{n, t, f\}$. If literal $u_i \in C_j$ then vertex $C_j$ is made adjacent to an odd outlet in $G_i$ using the new edge $e = u_i C_j$. Otherwise, (if $\overline{u_i} \in C_j$) then $C_j$ is made adjacent to an even outlet using the new edge $e = \overline{u_i} C_j$. By the above property of variable-gadgets, it follows that:

**Property 1**: $color(u_i C_j) \in \{n, color(h_i)\}$, and $color(\overline{u_i}, C_j) \in \{n, \overline{color(h_i)}\}$.

It is easy to verify that $G$ is a bipartite graph with *square* nodes on one side and clause gadgets plus *circle* nodes on the other side. Now, suppose that N-A-E 3-SAT has a truth assignment $val : U \rightarrow \{t, f\}$ in which each clause has at least one true literal and one false literal. Then the following coloring scheme is feasible:

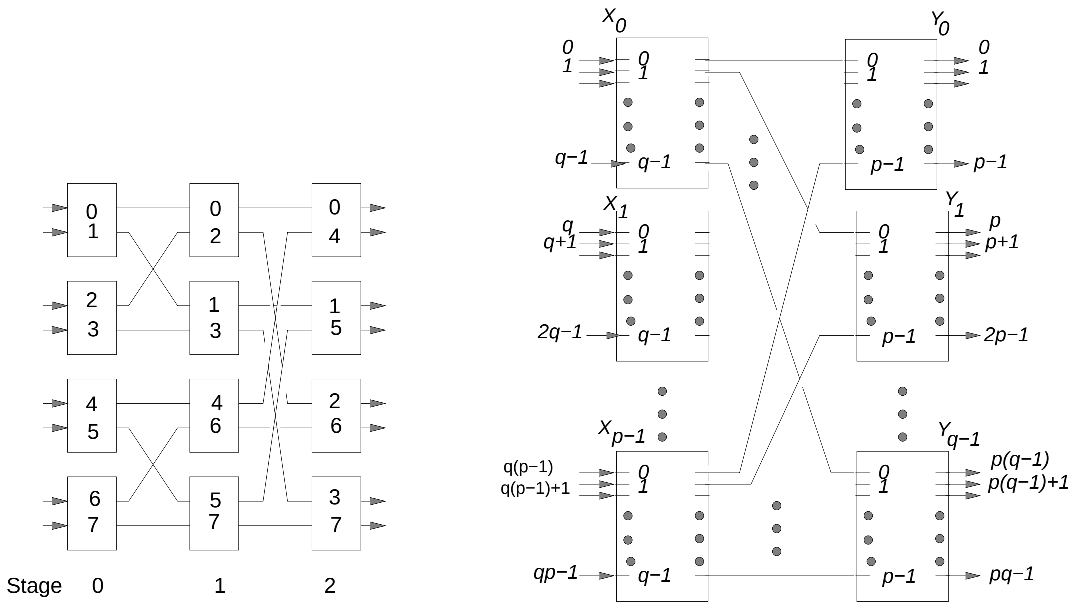
1. For every $u_i \in U$, set $color(h_i) = val(u_i)$.

2. For each clause $C_j$ identify a true literal, say $u_0$ (or $\overline{u_0}$), a false literal, say $u_1$ (or $\overline{u_1}$), and let $u_3$ be the third literal in $C_j$. For $u_i \in \{u_0, u_1\}$, set $color(u_i C_j) = val(u_i)$ if $u_i \in C_j$. Otherwise (if $\overline{u_i} \in C_j$), set $color(\overline{u_i} C_j) = \overline{val(u_i)}$. Finally, set $color(u_3 C_j) = n$.

Conversely, suppose that $(G, L)$ has a feasible coloring. Set $val(u_i) = color(h_i)$, for each $u_i \in U$. Now, each clause vertex $C_j$ has two edges $e_t$ and $e_f$ colored $t$ and $f$, respectively. For each $e \in \{e_t, e_f\}$, if $e$ is labeled $u_i C_j$ (for some $u_i$) then, by convention, literal $u_i \in C_j$. By Property 1, $color(u_i C_j) = color(h_i) = val(u_i)$. Likewise, if $e = \overline{u_i} C_j$ then $color(\overline{u_i} C_j) = \overline{color(h_i)} = \overline{val(u_i)}$. In both cases, variable $u_i$ gives a true literal in $C_j$ if $e = e_t$, and a false literal if $e = e_f$. This completes the proof. ∎

## 4 A Recursive Class of Multistage Networks

Hypercubic MINs form a class of cost-effective self-routing networks that include the omega network [15], the generalized cube network [22], and the indirect binary n-cube [21]. Parker [20], and independently, Wu and Feng [24], have shown that the inverse omega network, the indirect binary n-cube, a restricted version of the SW-banyan networks [9] are topologically equivalent. Hence, results on any of the above networks can be extended to the others. Here, we choose the indirect binary $n$-cube network ($\mathit{IBC}$, for short) as a representative of the above class, and introduce a recursive class that generalizes the $\mathit{IBC}$ networks. The recursive class, denoted $\mathit{GIBC}$, removes the restriction of using $2 \times 2$ switches, and allows switches of any arbitrary size to be used. The resulting proofs are more graph-theoretic, with a limited component based on manipulating network addresses at the binary level. In section 5, we define the class of extra-stage $\mathit{GIBC}$ networks and show two cases where the MCF problem can be solved efficiently.




(a) (b)

Figure 4.1. (a) an 8-input $\mathit{IBC}$ network. (b) a 2-stage network.

To set a framework, we need some definitions. The binary representation of a positive integer $x$, $x < 2^n$, is denoted $(x_{n-1}, \cdots, x_0)_2$. The *fan-in* of a MIN refers to the number of inputs (outputs) of the network. An $\mathit{IBC}$ [21] of fan-in $N = 2^n$ (see, for example, Figure 4.1a) has $(\lg N)$ stages numbered $0, 1, \cdots, n-1$ from the input side to the output side. The topology can be described using the *cube* functions: $cube_i(x) = (x_{n-1} \cdots \overline{x_i} \cdots x_0)_2$, $0 \le i \le n-1$, as follows. At each stage $0 \le \sigma \le n-1$ of the network, each $2 \times 2$ switch is identified by a unique pair of addresses $(x, cube_\sigma(x))$, where bit $x_\sigma = 0$. Address $x$ is then associated with the upper input and output ports, and the second address $cube_\sigma(x)$, with bit $x_\sigma = 1$, is associated with the lower ports. Each internal link in the network joins an output port of one stage to the unique input port of the next stage having the same address.

Central to the structure of the new class is the topology of *ordered full-access* 2-stage networks. Figure 4.1b illustrates a typical network with $p$ primaries $(X_0, \cdots, X_{p-1})$, and $q$ secondaries $(Y_0, \cdots, Y_{q-1})$, where each primary is a rearrangeable switch with $q$ inputs, and each secondary is a rearrangeable switch with $p$ inputs. Two coordinate systems (<u>local</u> and <u>global</u>) are used to reference the inputs and outputs of switches at any stage. In both systems, output addresses within each switch are assumed to be in one-to-one correspondence with input addresses. Hence, for any system, it suffices to fix input labels at each stage. In the local system, the inputs within each switch are ordered sequentially $(0, 1, 2, \cdots)$ from top to bottom. In contrast, the global system associates a unique address with each input of any given stage, moreover, each link joins an output port of one stage to the unique input port of the next stage with the same address. In Figure 4.1b, local labels appear inside the switches, and global labels appear on both sides of the network. Using local coordinates, one may then refer to the $j$th input (or output) of a switch $X_i$ by $X^-_{i,j}$ (respectively, $X^+_{i,j}$). The intermediate links of the network then corresponds to the set: $\{(X^+_{i,j}, Y^-_{j,i}) |\ 0 \le i \le p-1,\ 0 \le j \le q-1\}$.

The class of *GIBC* networks is defined recursively as follows: any rearrangeable switch or any ordered full-access 2-stage network is a *GIBC*. Furthermore, assume that $G$ is a *GIBC* network with a stage $\sigma$ having switches of fan-in $n_\sigma$ each, and $H$ is a 2-stage *GIBC* network of fan-in $n_\sigma$. Then replacing each switch of stage $\sigma$ by a copy of $H$ gives a *GIBC* network $G'$ of the same fan-in and larger number of stages. Here, we say that $G$ derives $G'$ ($G \vdash G'$) in one step. The notation $G \overset{*}{\vdash} G'$ (or $G \overset{+}{\vdash} G'$) implies that $G'$ is obtained from $G$ in zero (respectively, one) or more steps. Note that $G$ has more permutation power than $G'$. With $n_\sigma$ denoting the fan-in of each switch in stage $\sigma$ of an $r$-stage *GIBC* network, define the profile of $G$ as the vector $(n_0, n_1, \cdots, n_{r-1})$. In case where a symbol, say $X_\sigma$ (possibly with additional subscripts), is used to designate the switches at stage $\sigma$ then the extended profile of the network is a vector $(X_0 : n_0, X_1 : n_1, \cdots, X_{r-1} : n_{r-1})$ of name-size pairs.

That *IBC* networks are properly contained in *GIBC* networks is shown next (where the notation "$\cong$" refers to the isomorphism operator).

**Lemma** 4.1: Let $G$ be a 2-stage *GIBC* network of fan-in $2^n$, $n \ge 2$, and let $H$ be an *IBC* of the same fan-in then $G \overset{*}{\vdash} H$.

Proof:

By induction on $n$. The lemma holds by definition for $n = 1$. So, assume that it holds for all integers less than $n > 2$. Let $(X : 2^a, Y : 2^b)$, $2^{a+b} = 2^n$, be $G$'s extended profile. By the induction hypothesis, one can construct a new network $G'$ by replacing each switch $X_i$ (or $Y_j$) by an *IBC* $X'_i$ (respectively, $Y'_j$), of the corresponding fan-in. Local labels of inputs within each switch $X_i$ (or $Y_j$) induces a labeling of all internal ports within the substitute network $X'_i$ (or $Y'_j$). We now define a global labeling of $G'$ (using addresses of length $a + b$ bits), and use the new labels to prove that $G' \cong H$.

1. For each *IBC* $X'_i$, $0 \le i \le 2^b - 1$, change every occurrence of local address $0 \le j \le 2^a - 1$ to a global address $i2^a + j$. Thus, all addresses in $X'_i$ agree in the most significant $b$ bits. Clearly, in $X'_i$ each switch at stage $0 \le \sigma \le a - 1$ connects two inputs whose addresses differ only in bit $\sigma$. Consequently, the leftmost $a$ stages of $G'$ are identical to their counterparts in $H$.

2. For each *IBC* $Y'_j$, $0 \le j \le 2^a - 1$, change every occurrence of local address $0 \le i \le 2^b - 1$ to the global address $i2^a + j$. Hence, addresses in $Y'_j$ agree in the least significant $a$ bits. In $Y'_j$ each switch at stage $0 \le \sigma \le b - 1$ connects two inputs whose addresses differ only in bit $\sigma + a$, proving that the rightmost $b$ stages of $G'$ are identical to their counterparts in $H$.

Finally, it is easy to verify that each intermediate link $(X^+_{i,j}, Y^-_{j,i})$ in $G$ connects an output with global address $i2^a + j$ to an input of the next stage, with the same address. ■

Lemma 5.1 in the next section introduces a key property used in deriving polynomial time MCF algorithms on extra-stage networks. An important ingredient of the proof is the following *uniqueness* (up to isomorphism) result shown in Lemmas 4.2 and 4.3 below. The proofs use the following $k$-way shuffle function: suppose that $k$ divides $n$ evenly, then a $k$-shuffle of the sequence $(0, 1, \cdots, n-1)$ is obtained by dividing the sequence into $n/k$ subsequences: $(i, i+1, \cdots, i+k-1)$, $i = 0, k, 2k, \cdots, \frac{n}{k} - 1$, and then constructing a new sequence by taking the first element of each subsequence, then the second element of each subsequence, and so forth. The location of an integer $\alpha$, $0 \leq \alpha \leq n-1$, in the new sequence is given by: $k$-shuffle($\alpha$)= $(\alpha \bmod \mathrm{k})\frac{\mathrm{n}}{\mathrm{k}} + \mathrm{int}(\frac{\alpha}{\mathrm{k}})$, where $\mathrm{int}(\frac{\alpha}{k}) = \lfloor \frac{\alpha}{k} \rfloor$.

**Lemma** 4.2: For any three integers $x, y$, and $z$ there exits a unique 3-stage *GIBC* network $H$ with profile $(x, y, z)$.
Proof:

By definition, $H$ can be only derived from one of the following 2-stage networks: $G_X$ with extended profile $(X\!:\!x, [YZ]\!:\!yz)$, or $G_{XY}$ with extended profile $([XY]\!:\!xy, Z\!:\!z)$. Denote by $H_X$ and $H_{XY}$ the two possible networks of extended profile $(X\!:\!x, Y\!:\!y, Z\!:\!z)$ that can be derived from $G_X$ and $G_{XY}$, respectively. We show that $H_X \cong H_{XY}$ by proving that a global labeling of the latter can be obtained from a global labeling of the earlier by applying the $x$-shuffle function to the secondaries (hence, we use a straightforward, but rather tedious, method of translating addresses between local coordinates of the 4 networks $G_X$, $H_X$, $G_{XY}$, and $H_{XY}$). In particular, we examine the following two types of edges:

1. In $G_X$, let $e = (X^+_{i,j}, [YZ]^-_{j,i})$ be an edge, where $i < yz$ and $j < x$. In $H_X$, $e$ is incident with switch $Y_\alpha$, where $\alpha = jz + \mathrm{int}(i/y)$. Hence, the coordinates of $e$ in $H_X$ are $(X^+_{i,j}, Y^-_{\alpha, i \bmod \mathrm{y}})$. Now, consider the other derivation $G_{XY} \vdash H_{XY}$. Module $[XY]_{\mathrm{int}(i/y)}$ in $G_{XY}$, when expanded, yields the distinguished switch $X_i$ in network $H_{XY}$. The offset of that distinguished switch, relative to the top of $[XY]_{\mathrm{int}(i/y)}$, is $(i \bmod \mathrm{y})$. Hence, within $[XY]_{\mathrm{int}(i/y)}$ we consider the edge $e' = (X^+_{i \bmod \mathrm{y}, \mathrm{j}}, Y^-_{j, i \bmod \mathrm{y}})$. In $H_{XY}$, $e'$ hits switch $Y_\beta$, where $\beta = \mathrm{int}(i/y)x + j$. Thus, the coordinates of $e'$ in $H_{XY}$ are $(X^+_{i,j}, Y^-_{\beta, i \bmod \mathrm{y}})$. With $i < yz$, $j < x$, and $\beta \leq xz$ secondaries of $H_X$, it then follows that $\mathrm{x{-}shuffle}(\beta) = \alpha$, proving that $e = e'$.

2. In $G_{XY}$, let $e = ([XY]^+_{i,j}, Z^-_{j,i})$, where $i < z$ and $j < xy$. In $H_{XY}$, e is incident to switch $Y_\alpha$, where $\alpha = ix + \mathrm{int}(j/y)$. Hence, the coordinates of $e$ in $H_{XY}$ are $(Y^+_{\alpha, j \bmod \mathrm{y}}, Z^-_{j,i})$. Now, consider the other derivation $G_X \vdash H_X$. Module $[YZ]_{\mathrm{int}(j/y)}$ in $G_X$, when expanded, yields the distinguished switch $Z_j$ in network $H_X$. The offset of that distinguished switch, relative to the top of $[YZ]_{\mathrm{int}(j/y)}$, is $(j \bmod \mathrm{y})$. Hence, within $[YZ]_{\mathrm{int}(j/y)}$, we consider the edge $e' = (Y^+_{i, j \bmod \mathrm{y}}, Z^-_{j \bmod \mathrm{y}, \mathrm{i}})$. In $H_X$, $e'$ is incident with switch $Y_\beta$, where $\beta = \mathrm{int}(j/y)z + i$. Thus, the coordinates of $e'$ in $H_X$ are $(Y^+_{\beta, j \bmod \mathrm{y}}, Z^-_{j,i})$. With $i < z$, $j < xy$, and $\alpha < xz$ secondaries of $H_{XY}$, it follows that $\mathrm{x{-}shuffle}(\alpha) = \beta$, proving that $e = e'$.

This completes the proof. ■

**Lemma** 4.3: For any sequence $\vec{n} = (n_0, n_1, \cdots, n_{r-1})$ of $r$ positive integers, there exists a unique *GIBC* network with profile $\vec{n}$.
Proof:

The lemma holds for $r \leq 2$, by definition of *GIBC* networks, and for $r = 3$, by Lemma 4.2. We now show that it holds for all $r > 3$. To derive a contradiction, let $r > 3$ be the smallest integer for which the lemma fails. That is, there exist two non-isomorphic $r$-stage networks $H'$ and $H''$ with the same profile $\vec{n}$. Denote by $G'$ and $G''$ the two $(r-1)$-stage networks that derive $H'$ and $H''$, respectively. In addition, assume that the derivation $G' \vdash H'$ expands stage $i$ of $G'$ to yield stages $i$ and $i+1$ of $H'$. Likewise, let $j$ be the index of the stage in $G''$ affected by the derivation $G'' \vdash H''$. Without loss of generality, we may assume that $i \leq j$. We distinguish the following cases.

1. **Case: $i = j$.** Here, $G'$ and $G''$ have the same profile. By the choice of $r$, $G' \cong G''$. Consequently, $H' \cong H''$, a contradiction.

2. **Case: $(i+1) < j$.** By the choice of $r$, there exists a unique $(r-2)$-stage network $G$ with profile $(n_0, \cdots, n_{i-1}, n_i n_{i+1}, \cdots, n_{j-1}, n_j n_{j+1}, \cdots, n_{r-1})$ that derives both $G'$ and $G''$. Thus, the first $j$ stages of $G''$ and $H'$ are identical, moreover, the last $r-(i+2)$ stages of $G'$ and $H''$ are identical. Thus, $H' \cong H''$, a contradiction.

3. **Case: $i+1 = j$.** Again, consider the unique $(r-2)$-stage network $G$ with profile $(n_o, \cdots, n_{i-1}, n_i n_{i+1} n_{i+2}, \cdots, n_{r-1}$ that derives both of $G'$ and $G''$. By Lemma 4.2, each switch of size $n_i n_{i+1} n_{i+2}$ (at the $i$th stage of $G$) expands to a unique 3-stage network of profile $(n_i, n_{i+1}, n_{i+2})$. Hence, $H' \cong H''$, a contradiction.

This completes the proof. ■

## 5 Cutsets of Generalized Extra-Stage Networks

A mincut of a directed graph underlying a MIN is a minimal set of edges whose removal destroys all directed paths between some set of inputs $I$, and another set of outputs $O$. A wide mincut is one where $I$ contains all inputs that can reach any edge in the cutset, and $O$ contains all outputs the are reachable from any edge in the cutset.

Extra-stage $IBC$ networks (denoted $IBC_+$) are obtained by adding a new stage (of $2 \times 2$ switches) with $N = 2^n$ inputs to the input side of an $IBC$ of the same fan-in. The resulting network is of type $IBC_{+k}$ if the new stage implements the $cube_k$ function, for some $0 \leq k \leq n-1$. See, for example, Figure 5.1a. Variations of this type of networks have been studied in the context of designing fault-tolerant networks (see, for example, the survey in [1]). Thus, any possible $IBC_+$ network has exactly two paths from any input $x$ to any output $y$. By convention, a type-0 path (denoted $P^0$, with additional subscripts) leaves the extra stage from an upper output port, and a type-1 path (denoted, $P^1$) leaves from a lower output port. Consequently, networks of this type are rich with mincuts of size $\leq 2$.

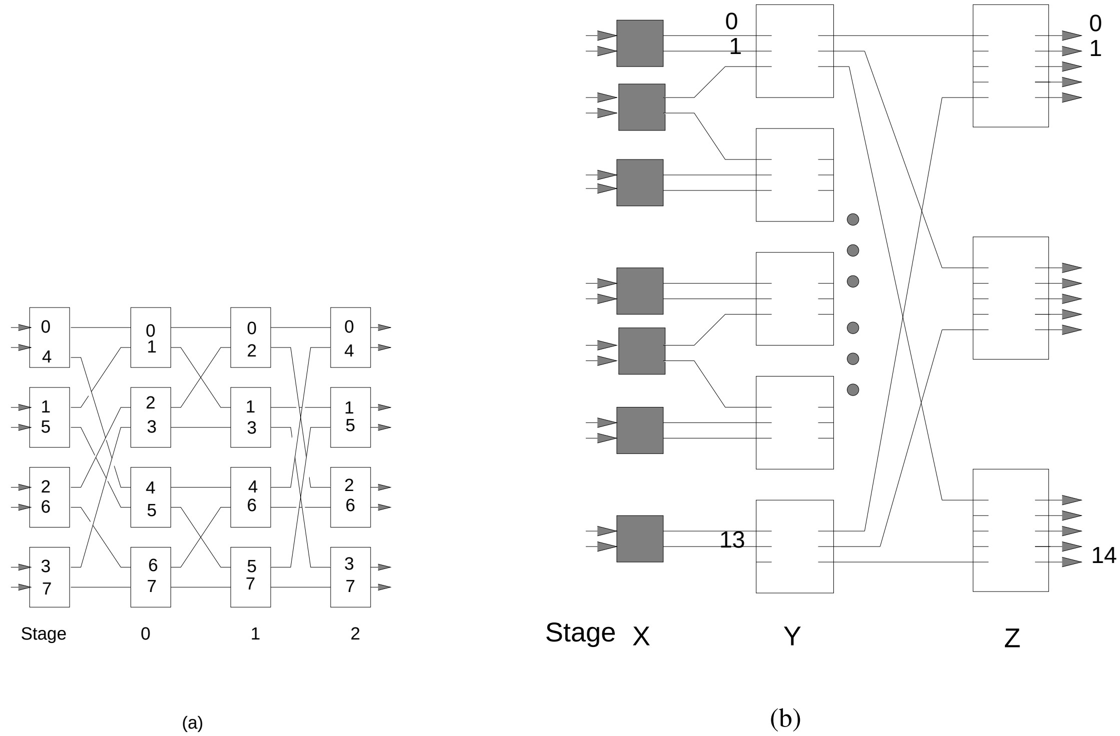



Figure 5.1. (a) A network of type $IBC_{+2}$.
(b) $GIBC_+$ network with profile $(X:2, Y:3, Z:5)$ and spread= 1.

We extend the above notion to $GIBC$-type networks in the following way. Let $G$ be one with profile $\vec{n} = (n_0, \cdots, n_{r-1})$, fan-in $N = n_0 \cdots n_{r-1}$, and inputs with global addresses $0, 1, \cdots, N-1$ from top to bottom. Call an integer $d$ valid for $\vec{n}$ if $d \leq N/2$ and the smaller of $d$ and $n_0$ divides the larger evenly. If $d$ is valid for $\vec{n}$ then connecting a new stage of $N/2$ $(2 \times 2)$-switches to the input stage of $G$ so that each input switch has two inputs with global addresses $(i, i+d)$ results in an extra-stage $GIBC$ network $G'$. The integer $d$ is called the spread of the extra-stage in $G'$. See, for example, Figure 5.1b. In a typical $IBC_{+k}$ network, $n_0 = 2$ divides the spread factor $2^k \leq N/2$, hence, $GIBC_+$ networks includes the class of $IBC_+$ networks as a proper subset. We now present a key structural property of the class $GIBC_+$ networks.

**Lemma** 5.1: Let $H$ be a $GIBC$ network with an edge $e$. Then

1. either $e$ is a wide mincut or there exists a unique edge $e^*$ in the same stage as $e$ such that $\{e, e^*\}$ is a wide mincut, and
2. if $\{e, e^*\}$ is a wide mincut separating inputs $\{s_1, s_2\}$ from outputs $\{t_1, t_2\}$, and $e \in P^0_{s_1,t_1}$ (or $P^1_{s_1,t_1}$) then $e \in P^0_{s_2,t_2}$ (respectively, $P^1_{s_2,t_2}$)

Proof of 1:

Let $(W:2, X_0:n_0, \cdots, X_{r-1}:n_{r-1})$ be $H$'s extended profile. If $e \in E_{W,X_0}$ then set $e^*$ to be the unique edge incident to the same input switch as $e$, and the lemma follows immediately. Else, $e \in E_{X_i,X_{i+1}}$, $0 \leq i \leq r-2$. Let $n_x = n_0 \cdots n_i$ and $n_y = n_{i+1} \cdots n_{r-1}$. By Lemma 4.3, there exits a unique 2-stage $GIBC$ network $G$ with an extended profile $(X:n_x, Y:n_y)$, that derives the rightmost $r$ stages of $H$. The edge $e$ can be uniquely identified in $G$ by the global addresses of its two end-points. For convenience, however, we use local coordinates in $G$ to refer to $e$. Thus, one may assume that $e = (X^+_{i,j}, Y^-_{j,i})$, for some $i < n_y$ and $j < n_x$.

Consider the extra-stage network $G'$ obtained by augmenting $G$ with an input stage of (2×2)-switches with the same spread factor $d$ as in $H$. Since mincuts of $G'$ are also mincuts of $H$, it then suffices to proof the lemma for $G'$. To this end, we examine the connected components of the subgraph $G'[W \cup X]$ (induced on the first two stages of $G'$), in each of the following cases.

1. Case: $n_x = \alpha d$, for some even integer $\alpha$. Each component of $G'[W \cup X]$ has exactly one $X$-switch. Hence, $e$ is a directed cut separating $X_i$ from $Y_j$.

2. Case: $n_x = \alpha d$, for some odd integer $\alpha$. Each component of $G'[W \cup X]$ has exactly two consecutive $X$-switches sharing $d$ input switches. Let $X_{i'}$ be the other switch in the same component as $X_i$. Set $e^*$ to the edge $(X^+_{i',j}, Y^-_{j,i'})$. No other edge in the same stage links the shared input switches to switch $Y_j$.

3. Case: $n_x$ divides $d$ evenly. Each component of $G'[W \cup X]$ has exactly two $X$-switches, each one can be reached from the same set of input switches. As in the above case, let $X_{i'}$ be $X_i$'s sibling, and set $e^* = (X^+_{i',j}, Y^-_{j,i'})$.

Proof of 2:

follows easily from the above argument. ■

We, henceforth, say that $e$ and $e^*$ are conjugate edges only if they lie in the same stage, and $\{e, e^*\}$ forms a wide mincut. Moreover, an edge $e$ is called essential for a flow demand $q(s,t)$ if $e \in P^0_{s,t} \cap P^1_{s,t}$. Based on the above property, we now present two well-solved cases of the MCF problem.

**Theorem** 5.1: The MCF problem (with $c$ and $q$ as in section 1) can be solved efficiently on any $GIBC_+$ network $H$, where

1. $H$ is a 3-stage network, or

2. $H$ has any possible number of stages, and $c$ and $q$ are unit-valued.

Proof of 1:

Let $(X\!:\!2, Y\!:\!y, Z\!:\!z)$ denote $H$'s profile. By Lemma 5.1, the subnetwork $H[X \cup Y]$ induced on the first two stages of $H$ is a disjoint union of connected components, each containing at most 2 switches of type $Y$. Label such components $G_0$, $G_1$, etc. (in any order). With $V(G_i)$ denoting the switches in the $i$th component, and $H_i = H[V(G_i) \cup Z]$ denoting the subnetwork of $H$ induced on $V(G_i)$ union the set of tertiaries, the claim then follows by applying Theorem 2.1(1) to each subnetwork $H_i$ independently.

Proof of 2:

We first outline a simple polynomial time algorithm that accepts the problem instance if there exists a feasible solution, and rejects, otherwise (the algorithm is not optimized for running time, rather its structure simplifies the proof).

[0] set $L = \{(s,t) : \; q(s,t) = 1\}$.

[1] For each demand $q(s,t) = 1$, allocate all essential edges $P^0_{s,t} \cap P^1_{s,t}$ to the demand. If any such edge has been previously allocated then reject.

[2] Repeat until no more demands can be removed from the current set $L$: for each demand $(s,t) \in L$, examine each edge $e \in P^0_{s,t} \cup P^1_{s,t}$. If $e$ is nonessential to $q(s,t)$ (that is, $e \notin P^0_{s,t} \cap P^1_{s,t}$), but it has been previously allocated to another demand, then allocate every edge on the path passing through the conjugate edge $e^*$ to $q(s,t)$. If any edge on that path has been previously allocated then reject. Delete $(s,t)$ from $L$, and iterate again.

[3] Denote the current set $L$ by $L'$. Construct a conflict graph $R$ on the set $\{v_{s,t} : \ (s,t) \in L'\}$ of vertices. Two vertices (demands) $v_{s,t}$ and $v_{s',t'}$ are adjacent in $R$ only if $P^0_{s,t} \cap P^0_{s',t'}$ contains an edge $e$ that is in nonessential for either $q_{s,t}$ or $q_{s',t'}$. Thus, by Lemma 5.1(2), the conjugate edge $e^* \in P^1_{s,t} \cap P^1_{s',t'}$. If $R$ is a bipartite graph with partitions $V_0$ and $V_1$ then assign each demand in $V_0$ (or $V_1$) to a type-0 (respectively, type-1) path, and accept. Else, reject.

To show the correctness, first, assume that step 3 accepts. It then suffices to show that demands in $L'$ are assigned to link-disjoint paths. To this end, observe the following:

1. If step 3 allocates a path $P_{s,t}$ for a demand $(s,t) \in L'$ then, prior to executing step 3, every edge $e \in P_{s,t}$ is either free, or has been allocated in step 1 to $q(s,t)$. Otherwise (i.e., if $e$ has been allocated to another demand) then step 2 would have removed $(s,t)$ from $L$.

2. By definition of the graph $R$, all paths assigned to demands in the same partition (either $V_0$ or $V_1$) are link-disjoint.

3. For any two vertices $v_{s,t} \in V_0$ and $v_{s',t'} \in V_1$, the two assigned paths $P^0_{s,t}$ and $P^1_{s',t'}$ are link-disjoint. If not, then let $e = P^0_{s,t} \cap P^1_{s',t'}$. Now, $e$ is nonessential for either demands (otherwise, step 2 would have allocated a path to one of them and, subsequently, deleted the other from $L$). Thus, $e$ is part of a wide mincut separating $(s,s')$ from $(t,t')$. By Lemma 5.1(2), $e \in P^0_{s',t'}$, contradicting the above claim that $e$ is nonessential for $(s',t')$.

In the other direction, suppose that the algorithm rejects. This happens in step 1 if an edge $e$ is essential for two or more demands. Since $c$ is unit-valued, the problem has no feasible solution. Likewise, in step 2, all path-allocation decisions are enforced. Failing to allocate a path to a demand implies the existence of a cutset whose demands exceed its capacity. Finally in step 3, suppose that $R$ has an odd cycle. Then any assignment of the demands in the cycle to paths (of type-0 or type-1) assigns two paths of the same type to two adjacent vertices. By definition of $R$, such an assignment maps two demands to two paths sharing at least one edge. Thus no solution exists. ■

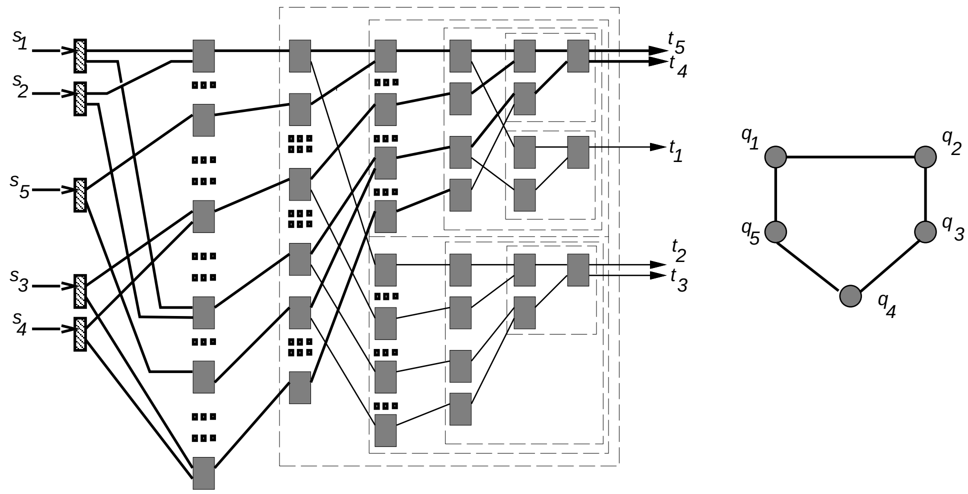



Figure 5.2. An instance satisfying conditions [CD] with no feasible solution.

By Theorem 2.1(1), conditions [CD] are sufficient for problems in part 1 of the above theorem to have feasible solutions. In contrast, Figure 5.2 shows an instance with 5 demands where conditions [CD] hold, yet step 3 of the above algorithm rejects. In the given instance, every edge (or flow demand) has a unit value, and each mincut has size 2 (hence, any wide mincut has an even number of edges). The observation follows by inspecting cuts of size 2 and 4. In particular, if the network has a violating cut of size 4 (i.e., one that disconnects all paths for the 5 demands) then it has a violating cut of size 2. However, not cut of size 2 disconnects more than 2 demands.

# 6 The Complexity Result

The complexity of the MCF problem on $r$-stage $GIBC_+$ networks seems to be open for $r = 4$ and $5$. The following theorem deals with the case when $r \geq 6$.

**Theorem** 6.1: The MCF problem ($c$ and $q$ are as in section 1) is NP-complete on 6-stage $GIBC_+$ networks where $q$ is unit-valued, and $c(e) \leq 3$ for any edge.

Proof:

Let $(U, C)$ be an instance of the 3-satisfiability problem on $|U| = n$ variables, and $|C| = m$ clauses, where each variable appears at most 3 times in $C$, and each clause has no more than 3 variables. The proof constructs an instance $(G, c, q)$ of the MCF problem on a 6-stage $GIBC_+$ network $G$, with the following demand sets:

1. $QC = \bigcup_{j=0}^{m-1} QC_j$: demands associated with clauses in $C$, where $QC_j = \{q(s_{j,k}, t_{j,k}) = 1 : u_k$ or $\overline{u}_k$ appears in $C_j\}$. Thus, $|QC| \leq 3m$. For simplicity, we use the abbreviation $q_{j,k}$ to denote $q(s_{j,k}, t_{j,k})$.

2. $QU = \bigcup_{k=0}^{n-1} QU_k$: demands associated with variables in $U$, where $QU_k = \{q(s_k, t_k) = 1, q(s_k, \overline{t}_k) = 1\}$. That is, $|QU| \leq 2n$. Again, for convenience, we use the simplified notation $q_k$ and $\overline{q}_k$ for the two demands above, respectively.

The topology of $G$, explained later, forces any feasible assignment of paths to the above flow requirements to satisfy the following conditions:

[c1] for each clause $C_j$, at least one demand, say $q_{j,k}$, is assigned a path of type 0.

[c2] for each variable $u_k$, exactly one of the two flow demands $q_k$ or $\overline{q}_k$ is assigned a path of type 0.

[c3] If demand $q_{j,k}$ is a assigned a type 0 path, and literal $u_k \in C_j$ then

[a] $q_k$ and any other demand $q_{j',k}$, where literal $u_k \in C_{j'}$, are assigned paths of type 0, and

[b] $\overline{q}_k$ and any demand $q_{j',k}$, where literal $\overline{u}_k \in C_{j'}$, are assigned paths of type 1.

Conversely, if $q_{j,k}$ is as above (i.e., assigned a path of type 0) and literal $\overline{u}_k \in C_j$ then demands in [b] are assigned to paths of type 0, and these in [a] are assigned paths of type 1.

Assuming that [c1], [c2], and [c3] hold in $G$, one may verify that any feasible assignment of flow requirements can be used to construct a truth assignment $val : U \rightarrow \{t, f\}$ that satisfies each clause in $C$. The strategy is to set $val(u_k) = 1$ if $q_k$ is assigned a type 0 path (cf. condition [c2]). Else, if $\overline{q}_k$ is assigned to a type 0 path then set $val(u_k) = 0$. Condition [c1] then implies that at least one demand $q_{j,k}$ uses a type 0 path, and condition [c3] ensures that literal $u_k$ (or $\overline{u}_k$) satisfies $C_j$.

In the remaining part, we describe the structure of $G$, and show that it satisfies the above conditions. Subsequently, we show that any solution to the given instance $(U, C)$ of the 3-SAT problem implies a solution to the instance $(G, c, q)$ of the MCF problem. The network $G$ has the extended profile $(A:2, V:3, W:n, X:m, Y:2, Z:2)$. The rightmost 5 stages of $G$ forms a *GIBC* network, denoted $H_5$. We derive $H_5$ from a 2-stage network $H_2$ with profile $([VW]:3n, [XYZ]:4m)$, having the following types of gadgets.

**Clause Gadgets:**

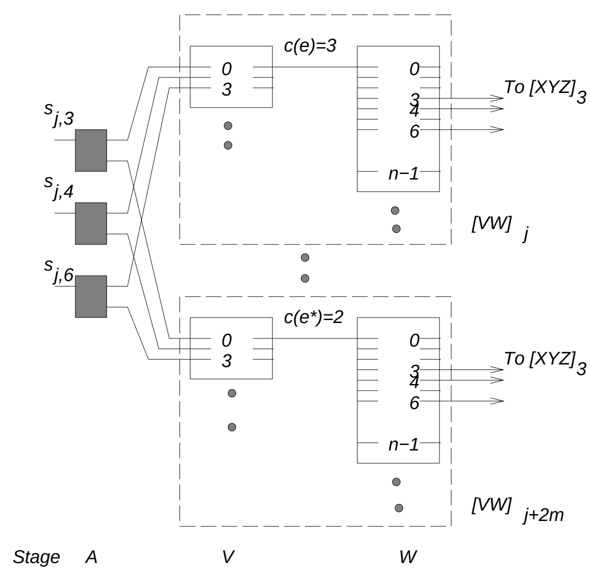



Figure 6.1 A clause gadget for $C_j = \{u_3, u_4, \overline{u}_6\}$.

For each clause $C_j$, the corresponding gadget is formed by the two switches $[VW]_j$ and $[VW]_{j+2m}$ of network $H_2$, plus a set of 3 switches from stage $A$. Figure 6.1 illustrates a typical gadget corresponding to a hypothetical clause $C_j = \{u_3, u_4, \overline{u}_6\}$. Note the inputs $s_{j,3}$, $s_{j,4}$, and $s_{j,6}$ required for the flow requirements

$q_{j,3}$, $q_{j,4}$, and $q_{j,6}$, respectively. Condition [c1] is guaranteed by assigning the critical capacities $c(e) = 3$ and $c(e^*) = 2$, for the two distinguished edges $e$ and $e^*$ shown in the gadget. Any other edge has a default capacity of one unit.

### The Auxiliary Gadget

The gadget is formed by switches $[VW]_m$ and $[VW]_{3m}$ of network $H_2$, plus $n$ switches from stage $A$, as shown in Figure 6.2. For $0 \le k \le n-1$, the gadget contains an input $s_k$ required by the two demands $\{q_k, \overline{q}_k\} \in QU_k$. It is easy to verify that the structure satisfies condition [c2], above.

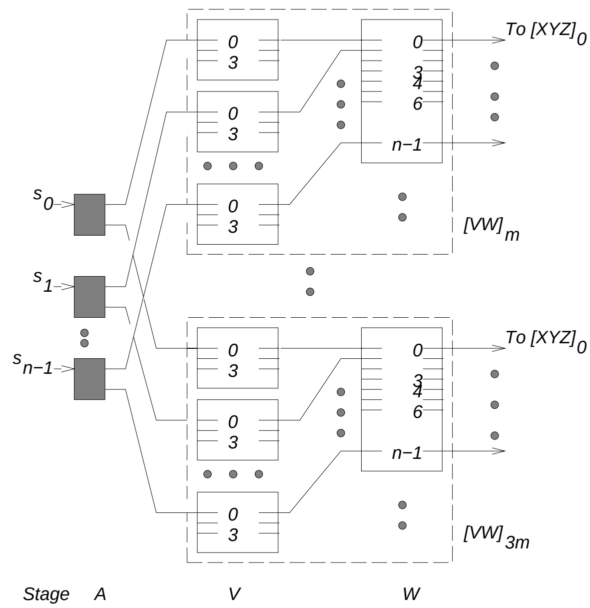



Figure 6.2 The auxiliary gadget for $|U| = n$ variables.

### Variable Gadgets

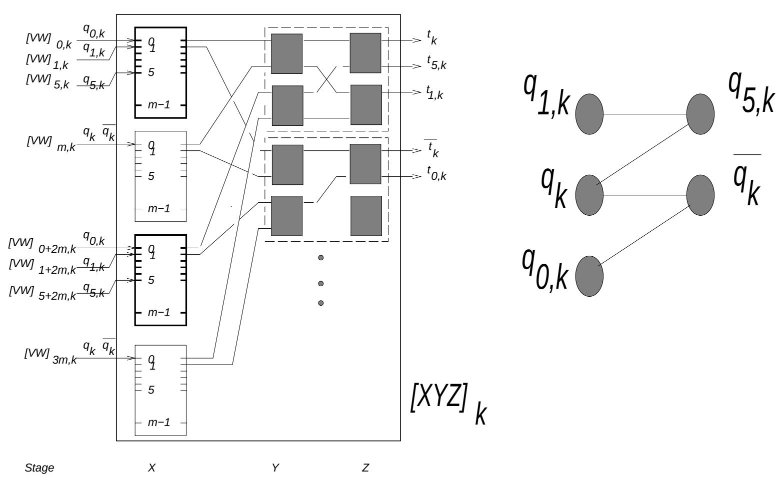



Figure 6.3 A variable gadget where $u_k \in C_0$ and $C_1$,

$\overline{u}_k \in C_5$, and the corresponding conflict graph.

For each variable $u_k \in U$, the corresponding gadget is formed by part of switch $[XYZ]_k$. The detailed structure depends on the distribution of literals $u_k$ and $\overline{u}_k$ among the clauses in $C$. Figure 6.3 illustrates a typical structure, assuming that literal $u_k$ appears in $C_0$ and $C_1$, and the complement $\overline{u}_k \in C_5$. In the figure, only the relevant part of the switch is shown, and each of the relevant input links is labeled with the possible flow demands that can pass through that link. Each link in the gadget has a unit capacity; this implies the conflict graph (shown on the right) on the set $\{q_{0,k}, q_{1,k}, q_{5,k}\} \cup \{q_k, \overline{q}_k\}$ of vertices. Hence, condition [c3] applies for this distribution.

By exchanging outputs $t_k$ and $\overline{t}_k$, we account for the case where literal $\overline{u}_k$ appears twice, say in $C_0$ and $C_1$, and literal $u_k$ appears once, say in $C_5$. The remaining distribution, where literal $u_k$ appears exactly once, say in $C_1$, and literal $\overline{u}_k$ appears exactly once, say in $C_5$, the terminal $t_{0,k}$ is no longer needed, but the remaining structure remains intact. In all of the above cases, one may verify that condition [c3] holds.

It is routine to verify that the above construction can be done in polynomial time. We now show that any satisfying assignment $val : U \rightarrow \{t, f\}$ yields a feasible solution to the above instance of the MCF problem. This can be done as follows:

[1] For each variable $u_k$, where literal $u_k$ satisfies some clause in $C$ (that is, $val(u_k) = 1$), assign $q_k$ and any other flow $q_{j,k}$, where literal $u_k \in C_j$, to paths of type 0. Assign $\overline{q}_k$ and the remaining demands of type $q_{*,k}$ to paths of type 1.

[2] (Converse of [1]) if literal $\overline{u}_k$ satisfies some clause (thus, $val(u_k) = 0$) then assign $\overline{q}_k$ and any other flow $q_{j,k}$, where $\overline{u}_k \in C_j$, to paths of type 0. Assign $q_k$ and the remaining requirements of type $q_{*,k}$ to a path of type 1.

[3] For each of the remaining variables, use the assignment of step [1].

Then at least one demand in each clause gadget will pass through the link with capacity 3, furthermore, the assignments are admissible by the variable gadgets. This concludes the proof. ■

The above proof can be easily extended to extra-stage *IBC* networks, hence.

**Corollary** 6.1: The MCF problem ($c$ and $q$ are as in section 1) is NP-complete on networks of type $IBC_+$, where $q$ is unit-valued, and $c(e) \leq 3$ for any edge.

## Acknowledgment

This research is supported by NSERC Canada under grant number OGP 36899 (ESE) and OGP 8053 (JC).

## References

[1] G. ADAMS III, D. AGRAWAL, AND H. SIEGEL, *A survey and comparison of fault-tolerant multistage interconnection networks*, Computer, 20 (1987), pp. 14–27.

[2] S. ARORA, T. LEIGHTON, AND B. MAGGS, *On-line algorithms for path selection in a nonblocking network*, in Proceedings of the Annual ACM Symposium on Theory of Computing, ACM, 1990, pp. 149–158.

[3] V. BENEŠ, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York, 1965.

[4] C. BERGE, *Graphs and Hypergraphs*, North-Holland-American Elsevier, 1973.

[5] J. BONDY AND U. MURTY, *Graph Theory with Applications*, McMillan Press, New York, 1976.

[6] S. EVEN, A. ITAI, AND A. SHAMIR, *On the complexity of timetable and multicommodity flow problems*, SIAM J. Comput., 5 (1976), pp. 691–703.

[7] M. GAREY AND D. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.

[8] A. GIRARD, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley, 1990.

[9] L. GOKE AND G. LIPOVSKI, *Banyan networks for partitioning multiprocessor systems*, in Proceedings of the 1st Annual Symposium on Computer Architecture, 1973, pp. 21–28.

[10] R. HASSIN, *On multicommodity flows in planar graphs*, Networks, 14 (1984), pp. 225–235.

[11] K. HEINRICH, P. HELL, D. KIRKPATRICK, AND G. LIU, *A simple existence criterion for (g < f)-factors*, Discrete Math., 85 (1990), pp. 313–317.

[12] P. HELL AND D. KIRKPATRICK, *Algorithms for degree constrained graph factors of minimum deficiency*, Journal of Algorithms, 14 (1993), pp. 115–138.

[13] J. HUI, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, 1990.

[14] D. KIRKPATRICK, M. KLAWE, AND N. PIPPENGER, *Some graph-colouring theorems with applications to generalized connection networks*, SIAM Journal Algebraic and Discrete Methods, 6 (1985), pp. 576–582.

[15] D. LAWRIE, *Access and alignment of data in an array processor*, IEEE Transactions on Computers, 24 (1975), pp. 1145–1155.

[16] G. LEV, N. PIPPENGER, AND L. VALIANT, *A fast parallel algorithm for routing in permutation networks*, IEEE transactions on Computers, 30 (1981), pp. 93–100.

[17] G. MASSON AND Y. YANG, *Nonblocking broadcast switching networks*, IEEE Transactions on Computers, 40 (1991), pp. 1005–1015.

[18] D. NASSIMI AND S. SAHNI, *Parallel permutation and sorting algorithms and a new generalized connection network*, Journal of the ACM, 29 (1982), pp. 642–667.

[19] H. OKAMURA AND P. SEYMOUR, *Multicommodity flows in planar graphs*, Journal of Combinatorial Theory, Series B, 31 (1981), pp. 75–81.

[20] D. PARKER JR., *Notes on shuffle/exchange-type switching networks*, IEEE Transactions on Computers, C-29 (1980), pp. 213–222.

[21] M. PEASE III, *The indirect binary n-cube microprocessor array*, IEEE Transactions on Computers, 26 (1977), pp. 458–473.

[22] H. SIEGEL, W. NATION, C. KRUSKAL, AND L. NAPOLITANO, *Using the multistag cube network topology in parallel supercomputers*, Proceedings of the IEEE, 77 (1989), pp. 1932–1953.

[23] H. J. SIEGEL, *Interconnection Networks for Large-Scale Parallel Processing*, McGraw-Hill Series in Computer Organization and Architecture, 1990.

[24] C. WU AND T. FENG, *On a class of multistage interconnection networks*, IEEE Transactions on Computers, C-29 (1980), pp. 694–702.

[25] ——, *The universality of the shuffle-exchange network*, IEEE Transactions on Computers, C-30 (1981), pp. 324–332.

[26] ——, *Interconnection Networks for Parallel and Distributed Processing*, The Computer Society of the IEEE, 1984.