On the L_∞-norm of Extreme Points for Crossing Supermodular Directed Network LPs

Harold N. Gabow *

Abstract

We discuss extensions of Jain's framework for network design [8] that go beyond undirected graphs. The main problem is finding an approximate minimum cost cover of a crossing supermodular function by a set of directed edges. We show that iterated rounding gives a factor 3 approximation, where factor 4 was previously known and factor 2 was conjectured. Our bound is tight for the simplest interpretation of iterated rounding. We also show that iterated rounding has unbounded approximation ratio when the problem is extended to mixed graphs.

1 Introduction

Iterated rounding, introduced by Jain [8], has taken its place as a general technique for approximating solutions to integer linear programs. Jain's original framework was for undirected network design problems. This paper investigates the directed case. The approximation guarantee for the directed case was known to be at most 4 but conjectured to be 2, just like the undirected case. We prove an upper bound of 3. We prove a matching lower bound, at least for the simplest version of iterated rounding. Specifically, we show that 1/3 is the tight lower bound on the L_∞-norm of an extreme point of the directed cut-covering LP for a crossing supermodular requirement function.

Previous Work

Jain showed that iterated rounding achieves a factor 2 approximation for the undirected Steiner network problem [8, 13]. Iterated rounding remains the only way known to achieve even an $O(1)$-approximation for this problem. More generally Jain gave a factor 2 approximation for any undirected network design problem whose cut requirements are given by a weakly supermodular function.

Subsequently a number of iterated rounding algorithms were presented for network design problems dealing with vertex connectivity. They are all based on the setpair covering LP. Fleischer et.al. achieve a 2-approximation for element connectivity as well as $\{0, 1, 2\}$ vertex connectivity, both on undirected graphs [6]. These are the best results known for these problems. Cheriyan et.al. [1] give an $O(\sqrt{m})$-approximation for setpair covering, which extends to an $O(n/\sqrt{n-k})$-approximation for k-vertex connectivity on both directed and undirected graphs [2] (a recent improvement is given in [10]).

Detouring a little, let *k-EC* stand for "k-edge connected" and let *k-ECSS* stand for "k-edge connected spanning subgraph". A problem of interest is finding a minimum cost k-ECSS. We will also refer to the problem of finding a minimum cost k-EC augmentation (this is the same problem,

*Department of Computer Science, University of Colorado at Boulder, Boulder, Colorado 80309-0430, USA. e-mail: hal@cs.colorado.edu

except the cost 0 edges are not recorded as part of the solution). Khuller and Vishkin give 2-approximations for these two problems, on both directed and undirected graphs [9]. Their work generalizes the algorithm for the case $k = 1$ by Frederickson and Ja'Ja' [5].

Most relevant to our results is the work of Melkonian and Tardos [11]. [11] generalizes Jain's framework to directed graphs. Jain's weakly supermodular cut requirement function becomes a crossing supermodular function. This makes the model less general but it still includes problems of interest: It includes the minimum cost directed k-ECSS problem and hence more sophisticated variations such as making a directed hypergraph k-edge connected by adding a minimum cost set of directed graph edges. (See Section 2.)

[11] proves that iterated rounding achieves an approximation ratio of 4. Melkonian and Tardos conjectured that the true approximation ratio is 2 (like the undirected case) and supported this conjecture with experimental evidence.

In addition to the iterated rounding algorithm, [11] proves an approximation ratio of 2 for their problem by using a different algorithm: It consists of two applications of an algorithm of Frank [3] which gives an exact solution in the special case of an intersecting supermodular cut requirement function. The resulting approximation algorithm is the generalization of [5, 9].

In the experiments this 2-approximation gave poorer results than iterated rounding, e.g., the 2-approximation averaged 25% above optimum while iterated rounding averaged within 1.5% of optimum (more details are in [11]). Such results, plus the success of iterated rounding on other problems, motivate our further study of the directed case.

Our Results

The only known method of proving a performance bound for iterated rounding is Jain's original result. It says iterated rounding gives a $1/\epsilon$-approximation when the LP under study is guaranteed to have optimum extreme points $\vec{x}$ with $\|\vec{x}\|_\infty \geq \epsilon$. [11] conjectured that supermodular directed cut problems have $\epsilon = 1/2$. We prove a tight bound of $\epsilon = 1/3$. More precisely for the cut covering LP of any supermodular directed cut problem, every nonzero extreme point $\vec{x}$ has $\|\vec{x}\|_\infty \geq 1/3$. Furthermore the simplest supermodular directed cut problem of interest, minimum cost k-EC augmentation, has an infinite family of instances where the unique optimum extreme point is $\vec{\frac{1}{3}}$. ($\vec{\frac{1}{3}}$ is our notation for a vector with all coordinates equal to 1/3. This result holds for any integer $k \geq 1$. The proof is based on a "generic" example which should allow it to extend to other supermodular cut problems.) The simplest version of iterated rounding gives approximation ratio exactly 3 on these instances (for both minimum cost k-EC augmentation and minimum cost k-ECSS). We leave open the possibility that a more sophisticated rounding strategy achieves approximation ratio 2. However new proof techniques will be required for such a result.

The reason we can prove a tight upper bound is that we concentrate more on linear independence than graph properties (starting with Proposition 3.6).

The last part of this paper briefly examines the supermodular cut problem on mixed graphs. Here the solution may use both directed and undirected edges. The main finding is negative: Looking again at minimum cost k-EC augmentation, for any integer $k > 1$ there is an infinite family of instances where the unique optimum extreme point is $\vec{\frac{2}{n}}$, for n the number of vertices. But for minimum cost 1-EC augmentation and generalizations like the hypergraph augmentation problem mentioned above, 1/4 is the tight lower bound for $\|\vec{x}\|_\infty$, $\vec{x}$ an extreme point of the cut covering LP. (So iterated rounding is a 4-approximation.)

Section 2 defines the supermodular directed cut problem and gives several examples. Section 3 proves the L_∞ lower bound, implying that iterated rounding achieves approximation ratio 3.

Section 4 shows our bound is tight, even for mincost k-ECSS. Section 5 gives the mixed graph results, with the analysis of 1-ECSS in Appendix A.

Terminology

Throughout this paper n and m denote the number of vertices and number of edges of the given graph $G = (V, E)$, respectively. In all graphs, each edge has a multiplicity that specifies the number of copies. (m ignores multiplicities.) Self-loops are not allowed.

We usually denote a path by listing its vertices, e.g., u, v, w. Sometimes for clarity we include the edges, e.g., $u\,(\alpha)\,v, w$ where the edge uv has been previously designated as α.

For $S \subseteq V$, G/S denotes the graph G with the vertices of S contracted. (Retain any parallel edges that may be created.)

For $S, T \subseteq V$, $\delta(S, T)$ denotes the number of edges from $S - T$ to $T - S$, and $\rho(S, T)$ denotes $\delta(T, S)$.

A set $S \subseteq V$ is *k-edge connected* if no two vertices of S are separated by a cut of $< k$ edges, i.e., any set of vertices X with $S \cap X, S - X \neq \emptyset$ has $\rho(X) \geq k$. Equivalently any two distinct vertices $s, t \in S$ are joined by k edge-disjoint st-paths.

2 Crossing Supermodular Directed Cut Problems

Sets $S, T \subseteq V$ *cross* if each of the four sets $S - T, T - S, S \cap T$ and $V - (S \cup T)$ is nonempty. A family $\mathcal{C}$ of subsets of V is *crossing* if $S, T \in \mathcal{C}$ with S and T crossing implies $S \cap T, S \cup T \in \mathcal{C}$. In this case a function $f : \mathcal{C} \to \Re$ is *crossing supermodular* if $f(S) + f(T) \leq f(S \cap T) + f(S \cup T)$ for every two crossing sets $S, T \in \mathcal{C}$.

Let $G = (V, E)$ be the given graph. We use the vector space $\Re^m$, where each dimension corresponds to an edge of E. An "overarrow" $\vec{\ }$ signifies a vector. In particular we use this notation:

$\vec{x}$ is a vector of values x_e for each $e \in E$. For $F \subseteq E$, $\vec{x}_F$ denotes the vector $\vec{x}$ with every component outside of F zeroed.

For $e \in E$, $\vec{e}$ denotes the unit vector having $x_e = 1$.

For $S \subseteq V$, $\vec{\delta}(S)$ ($\vec{\rho}(S)$) denotes the *boundary vector* of edges leaving (entering) S, i.e., the entry for e is the weight of e if e leaves (enters) S and 0 otherwise.

For $r \in \Re$, $\vec{r}$ denotes the vector with every entry equal to r.

A *(crossing) supermodular (directed) cut problem* is specified by a digraph $G = (V, E)$, a crossing family $\mathcal{C} \subseteq 2^V$ with crossing supermodular function $f : \mathcal{C} \to \mathbf{Z}$, and three vectors in $\Re^m_+$, an edge cost vector $\vec{c}$ and upper and lower bound vectors $\vec{u}$ and $\vec{l}$ respectively. Without loss of generality we assume $\vec{l} \leq \vec{u}$, $\vec{l}$ and $\vec{u}$ are integral vectors, and we allow infinite upper bounds. We seek a minimum cost set of edges respecting the upper and lower bounds such that every set $S \in \mathcal{C}$ has in-degree at least $f(S)$.

We give several supermodular cut problems that are of interest. The simplest is finding a minimum cost directed k-ECSS ($\mathcal{C} = 2^V - \{\emptyset, V\}$, $f \equiv k$). Most of the examples derive from this problem.

(a) Consider a "host" graph $H = (W, F)$, a subset of vertices $V \subseteq W$, and a set of "candidate" edges $E \subseteq V \times V$. A *k-EC internal augmentation (of V)* is a subset $E' \subseteq E$ with V k-edge connected in the graph $(W, E' \cup F)$. (Here E' and F are multisets; equivalently the weight of an edge e in $E' \cup F$ is the sum of its weights in E' and in F).

In the *mincost directed k-EC internal augmentation problem* we are given a directed host graph, a set of vertices V, plus candidate directed edges E and their costs. We seek a minimum cost subset of E that is a k-EC internal augmentation of V.

(b) (This is a special case of (a).) Recall that in a directed hypergraph, each hyperedge consists of a set of head vertices and a set of tail vertices [12, p.769]. (A vertex is allowed to be both a head and a tail.) The natural definitions of path, in-degree and k-edge connectivity apply to directed hypergraphs. It is easy to see that Menger's Theorem holds: For any two distinct vertices s, t, the greatest number of edge-disjoint st-paths equals the smallest number of hyperedges whose deletion destroys every such path. (This is proved by modelling a directed hypergraph by a directed graph.) A *graph-edge augmentation* of a hypergraph consists of the hypergraph plus a number of (ordinary graph) edges. A *k-EC graph-edge augmentation* makes the hypergraph k-edge connnected.

In the *mincost k-EC graph-edge augmentation problem for directed hypergraphs* we are given a directed hypergraph plus a set of (ordinary) candidate directed edges E and their costs. We seek a minimum cost subset of E that, when added to the hypergraph makes it k-edge connected.

(c) Any crossing supermodular function remains crossing supermodular when we increase its value on a singleton set arbitrarily. Similarly for complements of singleton sets. Hence we can add arbitrary lower bound constraints on the in-degree and out-degree of all vertices. As an example, consider the problem of finding a minimum cost directed spanning tree (i.e., all vertices must be reachable from a given root r). This can be done in polynomial time. Adding lower bounds on the out-degree of each vertex makes the problem NP-hard (it contains the directed Hamiltonian path problem). This problem is a supermodular directed cut problem.

(d) Example (c) can be generalized as follows. In a given directed graph, for each vertex v let $\mathcal{I}_v$ denote an intersecting family of subsets of edges directed to v, and let $i_v : \mathcal{I}_v \to \mathbf{Z}$ be an intersecting supermodular function. Define $\mathcal{O}_v$ and o_v similarly. We can enhance any supermodular directed cut problem by requiring that the solution contain at least $i_v(S)$ $(o_v(S))$ edges of S for every $S \subseteq \mathcal{I}_v$ $(S \subseteq \mathcal{O}_v)$ and every $v \in V$. As a simple example the given edges may be colored red or green and we require a minimum number of edges of each color in each direction at each vertex.

Note that we cannot hope for a constant factor approximation algorithm for the supermodular directed cut problem if f is a general function. For instance the directed Steiner tree problem is modelled by such a problem with $f(S)$ equal to 1 if S contains a terminal but not the root, else 0. However this function is not supermodular.

We use the following "cut covering" linear program, which results from dropping the integrality constraints from the supermodular directed cut problem. Recall our conventions for vectors, e.g., the inner product $\vec{x} \cdot \vec{\rho}(S)$ equals the in-degree of S using edge weights given by $\vec{x}$.

$$(LP) \qquad \begin{array}{llcll} \text{minimize} & \vec{x} \cdot \vec{c} & & & \\ \text{subject to} & \vec{x} \cdot \vec{\rho}(S) & \geq & f(S) & S \in \mathcal{C} \\ & \vec{x} & \leq & \vec{u} & \\ & \vec{x} & \geq & \vec{l} & \end{array}$$

Our approximation algorithm is iterated rounding. Although any version suffices for our upper bounds we need to be specific for the lower bound examples. We use *round-the-highest* [11]: Find an optimum extreme point $\vec{x}$. Fix the value of all components x_e that are largest (i.e., equal to $\|\vec{x}\|_\infty$) to their final value $\lceil x_e \rceil$. Modify (LP) to the residual problem (i.e., replace $f(S)$ by $f(S) - \lceil x_e \rceil$ if e enters S). Repeat the procedure until all values are fixed.

As shown in [8, 13], if any nonzero extreme point for any instance of (LP) has L_∞-norm at least ϵ then iterated rounding achieves a $1/\epsilon$ approximation ratio.

3 Lower Bound for Extreme Points

This section proves the L_∞ lower bound:

Theorem 3.1 *Any nonzero extreme point $\vec{x}$ to (LP) has $\|\vec{x}\|_\infty \geq 1/3$. Hence iterated rounding achieves approximation ratio 3 for the supermodular directed cut problem.*

Consider an arbitrary extreme point $\vec{x}$ to (LP). By definition the m components of $\vec{x}$ are uniquely determined by some set of m constraints that have been changed to equalities. Let F be the set of edges that are not determined by their bounds, $F = \{e : x_e \notin \{l_e, u_e\}\}$. The vector $\vec{x}_F$ is determined by the degree constraints, i.e., for some family $\mathcal{L}$ of subsets of V, the vector of unknowns $\vec{x}_F$ is the unique solution to the linear system

$$\vec{x}_F \cdot \vec{\rho}(S) = g(S), \quad S \in \mathcal{L}$$

where $g(S) = f(S) - \vec{x}_{E-F} \cdot \vec{\rho}(S)$ is a positive integer for every $S \in \mathcal{L}$. Combining arguments of Jain [8] and Frank [4], one can show that $\mathcal{L}$ can be taken as laminar. Specifically Melkonian and Tardos proved the following:

Lemma 3.2 ([11]) *There is a laminar family of sets $\mathcal{L} = \mathcal{I} \cup \mathcal{O}$ such that $\vec{x}_F$ is the unique solution for $\vec{y}_F$ in the linear system*

$$\begin{array}{rcll} \vec{y}_F \cdot \vec{\rho}(S) & = & i(S) & S \in \mathcal{I} \\ \vec{y}_F \cdot \vec{\delta}(S) & = & o(S) & S \in \mathcal{O} \end{array} \qquad (1)$$

where each right-hand side $i(S), o(S)$ is a positive integer.

Note that $\vec{y}$ is a vector in $\Re^m$, although only the values in F are relevant. The lemma holds for (LP) with an arbitrary crossing supermodular function f. As expected, a set of $S \in \mathcal{I}$ ($S \in \mathcal{O}$) corresponds to a set S ($V - S$) with a tight constraint in (LP). We will use the terms *$\mathcal{I}$-set* and *$\mathcal{O}$-set* with the obvious meaning. Note that a set can be both.

We define a rooted forest $\mathcal{F}$ that represents the laminar family $\mathcal{L}$. The nodes of $\mathcal{F}$ are partitioned into two sets $\mathcal{I}_\mathcal{F}$ and $\mathcal{O}_\mathcal{F}$. They represent the $\mathcal{I}$-sets and $\mathcal{O}$-sets of $\mathcal{L}$, respectively. Thus $\mathcal{F}$ has $|\mathcal{I}| + |\mathcal{O}|$ nodes. A set belonging to $\mathcal{I} \cap \mathcal{O}$ gives rise to two nodes.

Throughout the discussion we identify each forest node and its set. This may cause slight ambiguities but they will be resolved from the context. As an example if S is a node of $\mathcal{F}$, saying $S \in \mathcal{I}_\mathcal{F}$ means that S is being treated as a node while $S \in \mathcal{I}$ means it is being treated as a set. In the latter context we will speak of a vertex in S. On the other hand when it is entirely clear that S is a node we may use the simpler notation $S \in \mathcal{I}$.

If S is a set in $\mathcal{I} \cap \mathcal{O}$ let S_p and S_c denote the two corresponding nodes. Define S_p to be the parent of S_c in $\mathcal{F}$. We do not specify which node belongs to $\mathcal{I}_\mathcal{F}$ and which to $\mathcal{O}_\mathcal{F}$ – this detail is irrelevant.

We complete the definition of $\mathcal{F}$ by specifying the remaining parent-child relations in $\mathcal{F}$: The children of a node S are the nodes T corresponding to the maximal proper subsets of S in $\mathcal{L}$. In this definition replace S by S_c if $S \in \mathcal{I} \cap \mathcal{O}$, and similarly replace T by T_p if $T \in \mathcal{I} \cap \mathcal{O}$. This completes the definition of $\mathcal{F}$.

A node of $\mathcal{F}$ is a *leaf*, *chain node* or *branching node* depending on whether it has 0, 1 or > 1 children, respectively. Chain nodes have a further classification: A chain node is a *1-node* if it belongs to the same family $\mathcal{I}_\mathcal{F}$, $\mathcal{O}_\mathcal{F}$ as its (unique) child. If not (its child belongs to the opposite family) it is a *2-node*. For example if $S \in \mathcal{I} \cap \mathcal{O}$ then S_p is a 2-node; S_c may be any type of node (including a 2-node).

The overall proof strategy is Jain's [8, 13]: Assume for the sake of contradiction that no edge has value $\geq 1/3$, i.e., $0 < x_e < 1/3$ for every edge e. (We can ignore edges with $x_e = 0$.) We will assign two ends of an edge to each node of $\mathcal{F}$. Each end will be assigned at most once, and at least two ends will remain unassigned. This contradicts the fact that the number of edges equals the number of nodes of $\mathcal{F}$ (by Lemma 3.2).

We define a notion of "availability" to guide the assignment of ends. Each node of $\mathcal{F}$ will be assigned ends that are available to it. Intuitively we will show that ends available to the leaves of $\mathcal{F}$ pay for the leaves and the branching nodes, while ends available to the 1-nodes pay for themselves. The 2-nodes pay for themselves, but their available ends can come from other 2-nodes or branching nodes. The most involved part of the argument is the analysis of 2-nodes. For this reason we postpone all details concerning 2-nodes and begin by discussing the other nodes.

Consider a chain node S with unique child A. Let e be an edge with an end in $S - A$. We'll call e "p-directed" if it's oriented consistent with S's family $\mathcal{I}$ or $\mathcal{O}$ and "c-directed" if it's oriented consistent with A's family. In precise terms e is *p-directed* (*c-directed*) if either

(*i*) $S \in \mathcal{I}_\mathcal{F}$ ($A \in \mathcal{I}_\mathcal{F}$) and e enters S or A,

or (*ii*) $S \in \mathcal{O}_\mathcal{F}$ ($A \in \mathcal{O}_\mathcal{F}$) and e leaves S or A.

Note that in (*i*) only one of the two possibilities holds, i.e., e cannot enter both S and A, since it has an end in $S - A$. Similarly in (*ii*). Also for a 1-node the notions of p-directed and c-directed agree, while for a 2-node they disagree.

Let vertex v be an end of edge e and let S be a node of $\mathcal{F}$. Below we define when v, as an end of e, is available to S. For simplicity we will abbreviate this to "v is available to S" except when it is necessary to refer to e. The definition for 1-nodes is illustrated in Fig.1(a). Note that availability to 2-nodes will be defined later, here we give the definition for the other cases.

Definition 3.3 *End v of edge e is* available *to node S of $\mathcal{F}$ if (i) or (ii) holds:*

(i) When S is a leaf, $v \in S$. Furthermore either $S \in \mathcal{I}_\mathcal{F}$ and e enters S or symmetrically, $S \in \mathcal{O}_\mathcal{F}$ and e leaves S.

(ii) When S is a 1-node, $v \in S - A$ for A the child of S. Furthermore e is p-directed.

Clearly v, as an end of e, is available to at most one node.

Lemma 3.4 *The number of ends available to leaves of $\mathcal{F}$ exceeds twice the number of leaves and branching nodes.*

Proof: A leaf $S \in \mathcal{I}_\mathcal{F}$ is entered by ≥ 4 fractional edges. (If not, $\vec{x} \cdot \vec{\rho}(S)$ a positive integer makes $x_e \geq 1/3$ for some entering edge e.) Let $\mathcal{F}$ have ℓ leaves. Hence $\geq 4\ell$ edge ends are available to leaves. $\mathcal{F}$ has $< \ell$ branching nodes, since any rooted forest has fewer branching nodes than leaves. The lemma amounts to $4\ell = 2(2\ell)$. □

The rest of the proof is based on linear independence of $\mathcal{L}$. Specifically the set of vectors $\{\vec{\rho}(S) : S \in \mathcal{I}\} \cup \{\vec{\delta}(S) : S \in \mathcal{O}\}$ is linearly independent.

Lemma 3.5 *Each 1-node has at least two available ends.*

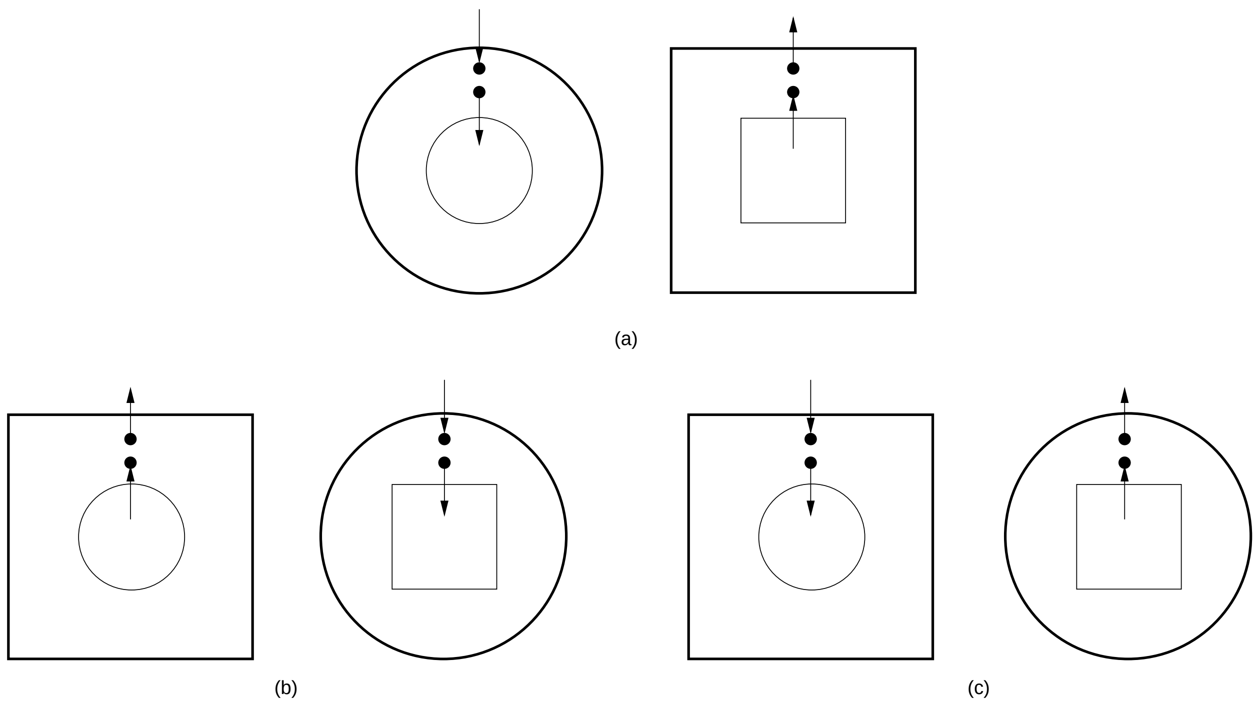

Figure 1: Chain nodes with their unique child. Circles represent $\mathcal{I}$-sets and squares $\mathcal{O}$-sets. The chain node is outermost and drawn heavier. (a) 1-nodes and their available edges. (b)–(c) 2-nodes and their corresponding available edges, when the 2-node is (b) the root, (c) a leaf.

Proof: Consider a 1-node S with child A, where *wlog* $S, A \in \mathcal{I}_{\mathcal{F}}$. The vector $\vec{a} = \vec{\rho}(S) - \vec{\rho}(A)$ has entries 0 and ± 1. Every edge e with $a_e = \pm 1$ has an end available to S: If $a_e = 1$ then e enters S but not A. If $a_e = -1$ then e enters A but not S. In both cases e is p-directed and has an end in $S - A$ available to S.

Now we need only show $\vec{a}$ has ≥ 2 nonzero entries. $\vec{a} \neq \vec{0}$ by linear independence. $\vec{x} \cdot \vec{a}$ is a signed sum of fractions x_e that is an integer (the in-degrees $\vec{x} \cdot \vec{\rho}(S)$ and $\vec{x} \cdot \vec{\rho}(A)$ are integral by Lemma 3.2). Hence there cannot be only 1 nonzero a_e. $\square$

Analysis of 2-nodes

Our theorem follows from Lemmas 3.4–3.5 if we show the 2-nodes pay for themselves. We begin with two simple facts that form the core of the proof:

Proposition 3.6 *(i) For any sets $A \subseteq S \subseteq V$, $\vec{\delta}(S) = \vec{\delta}(A) + \vec{\delta}(S - A, V - S) - \vec{\delta}(A, S - A)$. Similarly for $\vec{\rho}(S)$.*

(ii) For any $S \subseteq V$ partitioned into sets A_i, $i = 1, \ldots, r$, $\vec{\delta}(S) - \vec{\rho}(S) = \sum_{i=1}^{r} \vec{\delta}(A_i) - \vec{\rho}(A_i)$.

Proof: (ii) follows from the well-known fact that $\rho - \delta$ is a modular function. For a direct proof observe that an edge from A_i to A_j has a coefficient of zero on the right-hand side, since it has coefficient one in both $\vec{\delta}(A_i)$ and $\vec{\rho}(A_j)$. $\square$

We start the analysis by defining a subtree $\mathcal{F}_S$ for each node S of $\mathcal{F}$. $\mathcal{F}_S$ is the minimal subtree of $\mathcal{F}$ having S as its root and each leaf either a leaf of $\mathcal{F}$ or a 2-node. As a simple example, $\mathcal{F}_S$ is the single node S when S is a leaf or a 2-node.

For any 2-node S we define a second subtree $\mathcal{F}^*_S$. $\mathcal{F}^*_S$ is defined like $\mathcal{F}_S$ except S must be interior. More precisely $\mathcal{F}^*_S$ is the minimal subtree of $\mathcal{F}$ having S as its root and each leaf either a leaf of $\mathcal{F}$ or a 2-node other than S.

The various trees $\mathcal{F}^*_S$ can overlap: A 2-node S occurs at the root in $\mathcal{F}^*_S$, and also as a leaf in $\mathcal{F}^*_T$ for T the first 2-node that is a proper ancestor of S (if such an ancestor exists). It is easy to see these are the only possible overlaps.

We now complete the definition of availability. This part of the definition is illustrated in Fig.1(b)–(c).

Definition 3.7 *End v of edge e is* available *to the 2-node S if*

(i) v (as the end of e) is not available to any leaf or 1-node;

(ii) the deepest node of $\mathcal{F}$ that contains v, say X, belongs to $\mathcal{F}^*_S$;

(iii) if X is a 2-node then either $X = S$ and e is p-directed, or $X \neq S$ and e is c-directed.

We assert that v, as an end of e, is available to at most one node S of $\mathcal{F}$. This is clear from (i)–(ii) above unless X, the deepest node of $\mathcal{F}$ containing v, belongs to two trees $\mathcal{F}^*_S$. In that case X is a 2-node and the trees are $\mathcal{F}^*_X$ and $\mathcal{F}^*_S$ for some $S \neq X$. Since X is a 2-node edge e cannot be both p-directed and c-directed (as in Fig.1(b)–(c)). Hence (iii) makes v available to only one of X and S.

The analysis of 2-nodes uses Proposition 3.6 to derive a nontrivial linear equation for $\vec{\rho}(S)$ ($\vec{\delta}(S)$) when S is a 2-node in $\mathcal{I}_\mathcal{F}$ ($\mathcal{O}_\mathcal{F}$). This equation will immediately give the two edge ends that pay for S. To prove the equation we will require similar equations for $\vec{\delta}(S)$ ($\vec{\rho}(S)$) when S is an arbitrary node in $\mathcal{I}_\mathcal{F}$ ($\mathcal{O}_\mathcal{F}$). Notice the switch, e.g., when S is a 2-node in $\mathcal{I}_\mathcal{F}$ the first equation is for $\vec{\rho}(S)$ and the second is for $\vec{\delta}(S)$. We specify the desired equations in two steps: First we describe the equation for 2-nodes. Then we give a similar description for the arbitrary node case. The equation for 2-nodes is called an "r-expansion" since it is used at the root of $\mathcal{F}^*_S$. The equation for arbitary nodes is an "n-expansion" since it is used at all nonroots of $\mathcal{F}^*_S$.

We use this additional notation: If $\mathcal{T}$ is a subtree of $\mathcal{F}$ then $\bar{\mathcal{T}}$ denotes the tree $\mathcal{T}$ extended by including the (unique) child of each 2-node leaf of $\mathcal{T}$. For instance Fig.1(b) illustrates $\bar{\mathcal{F}}_S$ for a 2-node S.

Definition 3.8 *Let S be a 2-node. If $S \in \mathcal{I}_\mathcal{F}$ ($\mathcal{O}_\mathcal{F}$) an* r-expansion for $\vec{\rho}(S)$ ($\vec{\delta}(S)$) *is an equation whose left-hand side is $\vec{\rho}(S)$ ($\vec{\delta}(S)$) and whose right-hand side is*

$$\sum_{A\in O} \sigma_A \vec{\delta}(A) + \sum_{B\in I} \sigma_B \vec{\rho}(B) + \sum_{e\in F} \sigma_e \vec{e} \tag{2}$$

where

(i) σ_A, σ_B and σ_e are integers;

(ii) O (I) is a set of nodes contained in $\bar{\mathcal{F}}^*_S \cap \mathcal{O}_\mathcal{F} - S$ ($\bar{\mathcal{F}}^*_S \cap \mathcal{I}_\mathcal{F} - S$);

(iii) F is a set of edges, each having an end that is available to S.

Two aspects of this definition deserve comment. In actuality the σ factors are simply signs, $\pm$. Our main result only requires the weaker fact that the σ's are integral. We discuss the additional properties of an expansion after proving the main theorem.

Second consider the fact that condition (ii) excludes node S from O and I. So for example if $S \in \mathcal{O}_\mathcal{F}$ the trivial expansion $\vec{\delta}(S) = \vec{\delta}(S)$ is outlawed. But suppose S corresponds to a set in

$\mathcal{I} \cap \mathcal{O}$. In more detail take $S \in \mathcal{O}_{\mathcal{F}}$ with unique child $A \in \mathcal{I}_{\mathcal{F}}$, where S and A correspond to the same set of $\mathcal{L}$. (ii) allows I to contain A, e.g., an expansion can have the form $\vec{\delta}(S) = \vec{\rho}(A) + \ldots$. Since $\vec{\rho}(A)$ is identical to $\vec{\rho}(S)$ this amounts to $\vec{\delta}(S) = \vec{\rho}(S) + \ldots$. This of course is fine, since the equation will still be nontrivial.

Example Consider three chain nodes, $O, O' \in \mathcal{O}_{\mathcal{F}}$, $I \in \mathcal{I}_{\mathcal{F}}$, with I the unique child of O and O' the unique child of I. So O and I are 2-nodes. It is possible that I represents the same set as O or O'. The tree $\mathcal{F}^*_O$ consists of root O and leaf I; $\bar{\mathcal{F}}^*_O$ consists of O, I and leaf O'. The r-expansion of $\vec{\delta}(O)$ will derive from Proposition 3.6(i) and have the form $\vec{\delta}(O) = \vec{\delta}(O') + \sum \sigma_e \vec{e}$.

To extend Definition 3.8 let S be an arbitrary node (possibly a 2-node). Let $\hat{S}$ be the first proper ancestor of S that is a 2-node, if such exists. If $S \in \mathcal{I}_{\mathcal{F}}$ ($S \in \mathcal{O}_{\mathcal{F}}$) an *n-expansion for* $\vec{\delta}(S)$ *(*$\vec{\rho}(S)$*)* is defined just like Definition 3.8 changing (ii)–(iii) to the following:

(ii') $O \subseteq \bar{\mathcal{F}}_S \cap \mathcal{O}_{\mathcal{F}}$ and $I \subseteq \bar{\mathcal{F}}_S \cap \mathcal{I}_{\mathcal{F}}$;

(iii') F is a set of edges, each having an end that is available to $\hat{S}$.

This definition also merits several comments. First, (ii') allows $\vec{\rho}(S)$ ($\vec{\delta}(S)$) to occur on the right-hand side if $S \in \mathcal{I}_{\mathcal{F}}$ ($S \in \mathcal{O}_{\mathcal{F}}$). This makes sense since we are expanding $\vec{\delta}(S)$ ($\vec{\rho}(S)$).

Second, suppose S corresponds to a set in $\mathcal{I} \cap \mathcal{O}$, with $S \in \mathcal{I}_{\mathcal{F}}$ and its unique child $A \in \mathcal{O}_{\mathcal{F}}$. Although A does not belong to $\mathcal{F}_S$ it does belong to $\bar{\mathcal{F}}_S$. Hence the expansion $\vec{\delta}(S) = \vec{\delta}(A)$ is allowed, even though it amounts to the identity $\vec{\delta}(S) = \vec{\delta}(S)$. We allow these trivial expansions since they are just the first step in constructing the nontrivial expansions of Definition 3.8.

Finally note that (iii') is vacuous if $\hat{S}$ does not exist (the proof does not use this case).

Here is the main lemma.

Lemma 3.9 *Consider a node* $S \in \mathcal{I}_{\mathcal{F}}$. $\vec{\delta}(S)$ *has an n-expansion, and if* S *is a 2-node then* $\vec{\rho}(S)$ *has an r-expansion. Symmetrically for a node* $S \in \mathcal{O}_{\mathcal{F}}$.

Proof: The proof is by induction on the height of S in $\mathcal{F}$. The inductive assertion consists of the equation for $\vec{\delta}(S)$ or $\vec{\rho}(S)$ with right-hand side (2), plus properties (i) –(ii) below. The two properties simply restate the remaining restrictions from Definition 3.8 and its extension.

The notation $\hat{S}$ was introduced for n-expansions as the first proper 2-node ancestor of S. To unify the two cases take $\hat{S}$ to be S for r-expansions.

Consider any vector $\vec{e}$ occuring in (2), i.e., $e \in F$. We restate properties (iii) and (iii') of the definitions:

(i) Edge e has an end v available to $\hat{S}$.

Next we treat (ii) of Definition 3.8 and (ii'). These statements are essentially syntactic and require no explicit verification, except for excluding the term $\vec{\rho}(S)$ in an r-expansion for $\vec{\rho}(S)$: In the following O and I refer to the sets of (2).

(ii) In an r-expansion for S, $S \notin O \cup I$.

We organize the induction in three cases. We always assume that we seek an expansion for $\vec{\delta}(S)$. This implies $S \in \mathcal{I}_F$ for an n-expansion and $S \in \mathcal{O}_F$ for an r-expansion. The argument for an expansion for $\vec{\rho}(S)$ is symmetric.

Case 1. S is a leaf of $\mathcal{F}$. A leaf only has an n-expansion, so we are assuming $S \in \mathcal{I}_{\mathcal{F}}$. $\mathcal{F}_S$ consists of the single node S. The desired equation is

$$\vec{\delta}(S) = \sum \vec{e}$$

where e ranges over all edges leaving S. Property (i) holds since the tail of an edge leaving S is not available to S (as $S \in \mathcal{I}_\mathcal{F}$) so it is available to $\hat{S}$. (ii) is vacuous.

Case 2. S is a chain node. Let A be the unique child of S. We will sometimes break the analysis into three subcases. Since we seek an expansion of $\vec{\delta}(S)$ these subcases exhaust all possibilities:

(a) S is a 1-node. So $A, S \in \mathcal{I}_\mathcal{F}$ and we seek an n-expansion.

(b) S is a 2-node in $\mathcal{I}_\mathcal{F}$. So $A \in \mathcal{O}_\mathcal{F}$ and again we seek an n-expansion. The subtree $\mathcal{F}_S$ consists of the single node S. Also $\hat{S} \neq S$.

(c) S is a 2-node in $\mathcal{O}_\mathcal{F}$. So $A \in \mathcal{I}_\mathcal{F}$ and we seek an r-expansion. The subtree of interest is $\mathcal{F}^*_S$, and $\hat{S} = S$.

Proposition 3.6(i) applies to any chain node S and gives

$$\vec{\delta}(S) = \vec{\delta}(A) + \sum \pm \vec{e} \tag{3}$$

where $\sum \pm \vec{e}$ has e ranging over all edges with coefficient 1 in $\vec{\delta}(S - A, V - S)$ or $\vec{\delta}(A, S - A)$.

We show the sum $\sum \pm \vec{e}$ satisfies property (i) for v chosen as the end of e in $S - A$: First suppose S is a 1-node. End v of e is not available to the 1-node S (e is not p-directed since $S \in \mathcal{I}_\mathcal{F}$). Clearly this implies v is not available to any leaf or 1-node. This makes v available to $\hat{S}$.

If S is a 2-node then obviously v is not available to any leaf or 1-node. If $S \in \mathcal{I}_\mathcal{F}$ then $S \neq \hat{S}$ and e is c-directed, so v is available to $\hat{S}$. If $S \in \mathcal{O}_\mathcal{F}$ then $S = \hat{S}$ and e is p-directed, so again v is available to $\hat{S}$. This establishes (i).

We complete the analysis of Case 2 in two subcases. First suppose S is a 2-node in $\mathcal{I}_\mathcal{F}$. Then (3) is the desired expansion for S. The term $\vec{\delta}(A)$ is allowed in the expansion since $A \in \bar{\mathcal{F}}_S \cap \mathcal{O}_\mathcal{F}$. (If S belongs to $\mathcal{I} \cap \mathcal{O}$ we are deriving the equation $\vec{\delta}(S) = \vec{\delta}(A)$, i.e., $\vec{\delta}(S) = \vec{\delta}(S)$, but as already noted this is fine.)

The remaining possibilities are that S is a 1-node, or a 2-node in $\mathcal{O}_\mathcal{F}$. Both alternatives have $A \in \mathcal{I}_\mathcal{F}$. To get the desired expansion for $\vec{\delta}(S)$, take (3) and replace $\vec{\delta}(A)$ by its n-expansion given by induction. (This expansion exists since $A \in \mathcal{I}_\mathcal{F}$.)

Property (i) holds by induction. For (ii) suppose S is a 2-node in $\mathcal{O}_\mathcal{F}$. The right-hand side of the expansion of $\vec{\delta}(A)$ contains boundary vectors $\vec{\delta}(C)$, $\vec{\rho}(C)$ only for nodes $C \in \bar{\mathcal{F}}_A$. Clearly C is not node S. (Note that if A represents the same set as S, it is fine to have $\vec{\rho}(A)$ on the right-hand side.)

Case 3. S is a branching node. A branching node only has an n-expansion so we are assuming $S \in \mathcal{I}_\mathcal{F}$. Let the children of S be A_i, $i = 1, \ldots, r$ $(r > 1)$. Each A_i may belong to $\mathcal{I}_\mathcal{F}$ or $\mathcal{O}_\mathcal{F}$. Define A_0 to be the set $S - \cup_{i=1}^r A_i$. It is possible that A_0 is empty, and of course A_0 does not belong to $\mathcal{L}$.

Proposition 3.6(ii) applied to the partition A_i, $i = 0, 1, \ldots, r$ gives

$$\vec{\delta}(S) = \vec{\rho}(S) + \sum_{i=1}^{r} \vec{\delta}(A_i) - \vec{\rho}(A_i) + \sum \pm \vec{e} \tag{4}$$

where $\sum \pm \vec{e}$ has e ranging over all edges incident to A_0. This sum satisfies (i): Let v be the end of e in A_0. Clearly v is not owned by any leaf or 1-node, so v is available to $\hat{S}$.

To get the desired expansion for S, start with (4). Then for $i = 1, \ldots, r$, if $A_i \in \mathcal{I}_\mathcal{F}$ ($A_i \in \mathcal{O}_\mathcal{F}$) replace $\vec{\delta}(A_i)$ ($\vec{\rho}(A_i)$) by its n-expansion, respectively.

Note that property (i) continues to hold by induction. (ii) is vacuous. □

Observe from the proof that each term $\vec{e}$ occurring in an expansion has $|\sigma_e| \le 2$. This is clear since $\vec{e}$ can be introduced into the expansion at most twice, once for each of its ends.

We can now show how to pay for the 2-nodes. Consider a 2-node S. By symmetry assume $S \in \mathcal{O}_{\mathcal{F}}$. Recall that we wish to show at least two ends are available to S. In the r-expansion of $\vec{\delta}(S)$ each boundary vector is either $\vec{\delta}(C)$ for C an $\mathcal{O}$-set distinct from S, or $\vec{\rho}(C)$ for C an $\mathcal{I}$-set. But the boundary vectors for $\mathcal{L}$ are linearly independent. (If $S \in \mathcal{O}_{\mathcal{F}}$ corresponds to the same set of $\mathcal{L}$ as its child $A \in \mathcal{I}_{\mathcal{F}}$ then the term $\vec{\rho}(A) \equiv \vec{\rho}(S)$ may occur in the expansion. This is fine.) We conclude the expansion involves at least one edge vector term $\vec{e}$.

Next we show there cannot be exactly one such term. Multiply the expansion by $\vec{x}$. The term $\vec{x} \cdot \vec{\delta}(S)$ is the in-degree of S *wrt* weights $\vec{x}$. So it is integral. Similarly all the boundary terms $\vec{x} \cdot \sigma_C \vec{\delta}(C)$, $\vec{x} \cdot \sigma_C \vec{\rho}(C)$ are integral. If there is only one edge vector term $\vec{e}$, we have deduced that $\vec{x} \cdot \sigma_e \vec{e} = \sigma_e x_e$ is integral. But then $|\sigma_e| \le 2$ implies x_e is half-integral, contrary to our initial assumption. This shows the expansion contains at least two vectors $\vec{e}$. So we have the desired two ends available to S.

Theorem 3.1 has been proved. We conclude by mentioning an additional property of expansions.

Corollary 3.10 *Each integer $\sigma_A, \sigma_B, \sigma_e$ occurring in an expansion equals ± 1.*

Remark For example an edge vector $\vec{e}$ occurs as $+\vec{e}$ or $-\vec{e}$, even if e has two ends available to $\hat{S}$.

Proof: The corollary is established by an induction similar to the lemma. We simply state the inductive assertion, leaving the verification to the reader. In addition to the corollary we assert two properties. The first pertains only to n-expansions (it is vacuous for an r-expansion):

(i) A boundary vector involving S occurs on the right-hand side of an expansion only with a plus sign.

For example (i) means that if $S \in \mathcal{I}_{\mathcal{F}}$ and $\vec{\rho}(S)$ occurs on the right-hand side of the expansion of $\vec{\delta}(S)$ then it has a plus sign, $+\vec{\rho}(S)$.

The second property ensures uniqueness for the edge terms $\vec{e}$:

(ii) Let edge e be incident to S. If $\vec{e}$ occurs on the right-hand side of an expansion for $\vec{\delta}(S)$ ($\vec{\rho}(S)$) its sign is $+$ if and only if it occurs in the vector $\vec{\delta}(S)$ ($\vec{\rho}(S)$).

For example (ii) means that if $\vec{e}$ occurs on the right-hand side of an expansion for $\vec{\delta}(S)$ its sign is $+$ if it leaves S and $-$ if it enters S. (The sign is arbitrary if it does neither.)

The affect of these two properties is that whenever a term can potentially be introduced a second time in an expansion, it cancels the previous occurrence instead of building up a coefficient of two. $\Box$

4 Tight Example

This section shows Theorem 3.1 cannot be strengthened even for the simplest problems of interest. Specifically, suppose iterated rounding is used to approximate a minimum cost directed k-ECSS. In each iteration (LP) is the relaxation of a minimum cost directed k-EC augmentation problem. This section proves:

Theorem 4.1 *For any integer $k \ge 1$, an infinite family of instances of the minimum cost directed k-EC augmentation problem has $\frac{\vec{1}}{3}$ as the unique optimum solution to (LP). Round-the-highest has approximation ratio 3 on these instances (for both mincost k-EC augmentation and mincost k-ECSS).*

The proof begins by exhibiting a "generic" supermodular cut problem where $\vec{\frac{1}{3}}$ is the unique optimum. Any supermodular function that can model the generic problem will satisfy a corresponding version of Theorem 4.1. We show this is the case for minimum cost k-ECSS.

We begin by specifying the laminar family $\mathcal{L}$ and the extreme point $\vec{x}$ of the generic example. Every set of $\mathcal{L}$ has degree requirement equal to one. Every edge e has $x_e = 1/3$. Our example can be arbitrarily large, and is composed of units called "modules". There are three rules for constructing the example, illustrated in Fig.2: (In Fig.2(b) ignore the label X – it will be used later.)

(i) An *elementary module* consists of 8 sets, 5 internal edges and 6 incident edges, arranged as in Fig.2(a). The module on the left (right) has its outermost set in $\mathcal{O}$ ($\mathcal{I}$), so we call it an elementary $\mathcal{O}$-module ($\mathcal{I}$-module). The two types of modules are identical except that edges C and α are reversed, so the $\mathcal{O}$-module has two $\mathcal{O}$-sets (the outermost set and its left child) where the $\mathcal{I}$-module has two $\mathcal{I}$-sets.

(ii) A *compound module* consists of two modules combined as illustrated in Fig.2(b). As before the module on the left (right) is a compound $\mathcal{O}$-module ($\mathcal{I}$-module). Each of the two constituent modules is a *child module*, and the new sets and edges form the *parent module*. The two module types have the same edges except for the reversal of the three C edges.

(iii) A complete generic example consists of two modules combined as in Fig.2(c). Each of the two constituent modules is a *root module*.

A generic example can be viewed as two binary trees combined at the root. Hence we can use binary tree terminology, e.g., left child, left sibling, etc.

Fig.2 shows that every $\mathcal{I}$-set is entered by exactly three edges and every $\mathcal{O}$-set is left by exactly three edges. Hence taking every x_e to be $1/3$ gives every set of $\mathcal{L}$ the desired in- or out-degree of one. We will show this $\vec{x}$ is the unique vector for which all sets of $\mathcal{I}$ ($\mathcal{O}$) have in-degree (out-degree) equal to one.

Before proceeding we point out what's important in the construction and what isn't. Elementary modules of both types are needed to get a rich enough set of constraints. We could omit compound modules of one type from the generic example, but both types are needed to convert the example to a graph.

Some asymmetry in the two halves of an elementary module must be present. If not the degree requirements become linearly dependent and solutions other than $\vec{x}$ are introduced.

The last point applies if we wish to realize the example as a 1-ECSS problem. This outlaws certain configurations. For instance consider an elementary $\mathcal{O}$-module where $S \in \mathcal{O}$ is the left child of the outermost set, and S_1 (S_2) is its left (right) child. We cannot change S_2 from an $\mathcal{O}$-set to an $\mathcal{I}$-set. To see this observe $S \in \mathcal{O}$, $S_1, S_2 \in \mathcal{I}$ implies no integral edge leaves S or enters S_1 or S_2. That makes $S - S_1 - S_2$ a set of out-degree zero *wrt* the integral edges. This would make $\vec{x}$ infeasible since α is the only fractional edge leaving $S - S_1 - S_2$.

Now we prove the uniqueness of $\vec{x}$. To simplify notation we will use the edge identifiers to denote the corresponding values x_e, e.g., we write A instead of x_A, etc. We begin by writing the degree requirements for an elementary module. The four leaves give these requirements, for both $\mathcal{O}$- and $\mathcal{I}$-modules:

$$\begin{array}{ll} A + a + \gamma = 1 & A' + a' + \beta = 1 \\ B + a + \beta = 1 & B' + a' + \gamma = 1 \end{array} \tag{5}$$

For an $\mathcal{O}$-module the parents and grandparents of the leaves give these requirements,

$$\begin{array}{ll} B + \alpha + \beta = 1 & A' + \alpha + \beta = 1 \\ & B' + C + \gamma = 1 \end{array} \tag{6}$$

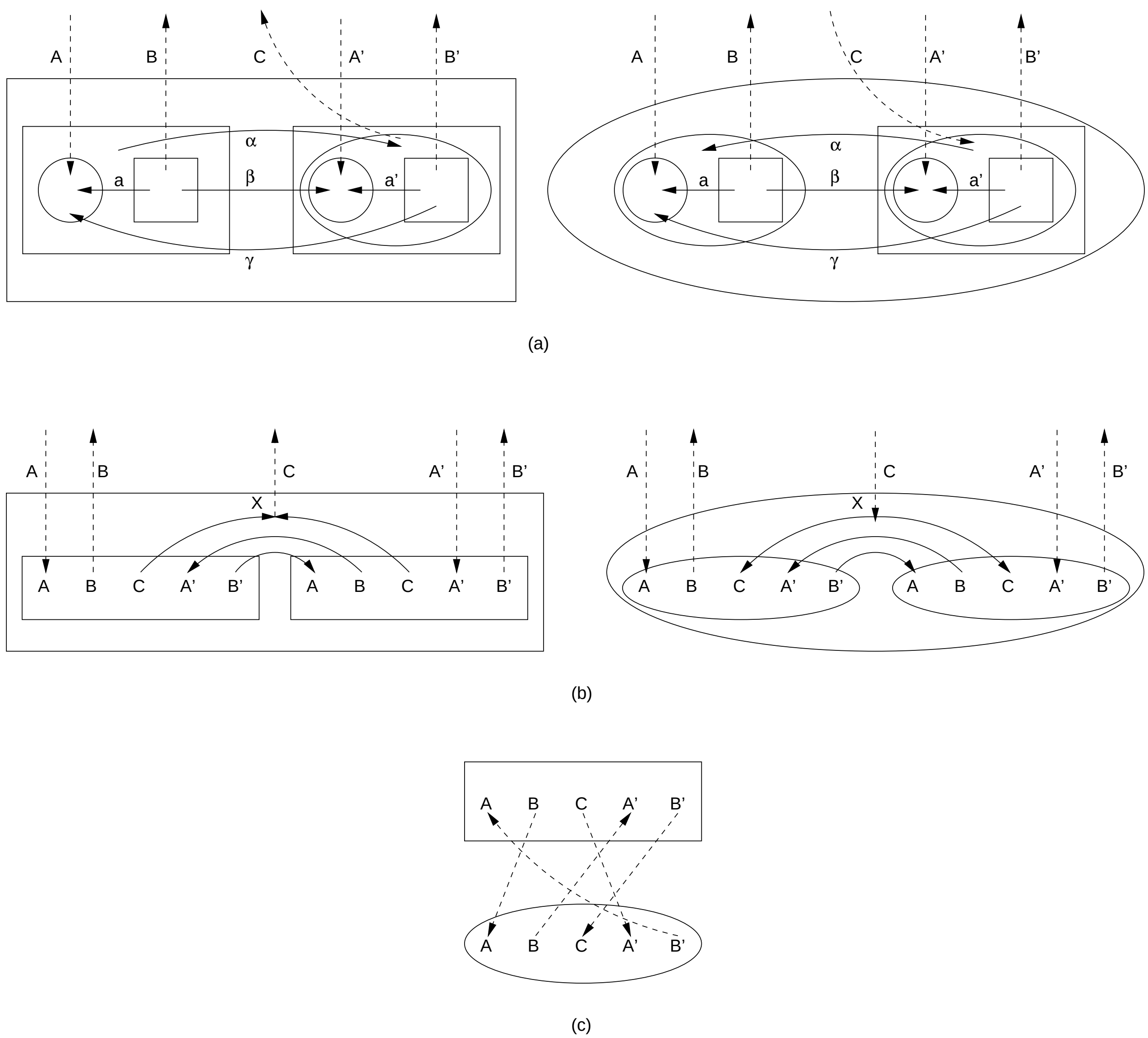

Figure 2: Lower bound example. Every module has incident edges A, B, C, A', B', which are drawn dashed. (a) Elementary modules of type $\mathcal{O}$ (left) and $\mathcal{I}$ (right). (b) Compound modules of type $\mathcal{O}$ (left) and $\mathcal{I}$ (right). (c) Root modules.

$\mathcal{O}$-module		$\mathcal{I}$-module	
compared equation	deduction	compared equation	deduction
	$A' = B$		$A = B'$
(6) bottom	$\gamma = B$	(8) bottom	$\beta = A$
(5) bottom right	$a' = C$	(5) top right	$a' = C$
(5) top right	$\beta = B'$	(5) bottom right	$\gamma = A'$
(5) bottom left	$a = C$	(5) top left	$a = C$
(5) top left	$A = B'$	(5) bottom left	$A' = B$
(6) top left	$\alpha = C$	(8) top left	$\alpha = C$

Table 1: Proof of Lemma 4.2(i).

and the outermost set gives this requirement:

$$B + B' + C = 1 \tag{7}$$

For an $\mathcal{I}$-module the requirements are similar:

$$A + \alpha + \gamma = 1 \qquad \begin{array}{l} B' + \alpha + \gamma = 1, \\ A' + C + \beta = 1 \end{array} \tag{8}$$

$$A + A' + C = 1 \tag{9}$$

Lemma 4.2 *Consider any feasible values for a generic example.*

(i) Any elementary module has $A = B'$ and $B = A'$. Furthermore each of the five internal edges equals one of the external edges A, B or C.

(ii) Any compound module has $A = B'$ and $B = A'$. Furthermore each of the four internal edges equals one of the external edges A, B or C.

(iii) Every edge incident to a root module equals $1/3$. Hence every edge of the entire generic example has value $1/3$.

Proof: (i) Consider an $\mathcal{O}$-module. The top two equations of (6) both have $\alpha + \beta$ on the left-hand side. Hence $A' = B$. Similarly for an $\mathcal{I}$-module (8) gives $A = B'$. The rest of the argument is similar: For $\mathcal{O}$-modules we repeatedly compare an equation with (7) and deduce the value of a variable. For $\mathcal{I}$-modules we repeatedly compare with (9). The argument is presented in Table 1.

(ii) In both modules of Fig.2(b) each edge identifier occurs three times. We differentiate them by using the subscript 1 (2) to refer to the occurrence in the left (right) child module and no subscript at all for the parent. So the leftmost edge is denoted A and also A_1.

We first show that when both children have $A = B'$ and $B = A'$ so does the parent. The argument is the same for both module types. Equating the two variables for each edge (in an order we'll use in a moment) gives

$$A = A_1 \qquad B_1' = A_2 \qquad B_2' = B' \qquad\qquad B = B_1 \qquad A_1' = B_2 \qquad A_2' = A'.$$

The first three quantities are equal since the assumption gives $A_1 = B_1'$, $A_2 = B_2'$. Thus $A = B'$ as desired. Similarly the second three quantities are equal, giving $B = A'$.

For the internal edges we have deduced $A = B_1' = A_2$ and $B = A_1' = B_2$. Consider an $\mathcal{O}$-module. Let $i = 1, 2$. Since we have deduced $B = B_i$ and $B' = B_i'$, the parent requirement

$B + B' + C = 1$ and the child requirement $B_i + B_i' + C_i = 1$ give $C = C_i$. The same argument applies to an $\mathcal{I}$-module.

(iii) We differentiate the edge identifiers in the top and bottom root modules (Fig.2(c)) by using the subscript 1 and 2 respectively. Equating the two variables for each edge, in an appropriate order, gives

$$C_1 = A_2' \qquad B_2 = A_1' \qquad B_1 = A_2 \qquad B_2' = A_1 \qquad B_1' = C_2.$$

(i) and (ii) show consecutive quantities are equal, $A_2' = B_2$, $A_1' = B_1$, $A_2 = B_2'$, $A_1 = B_1'$. Thus all ten quantities are equal. The in-degree equation $B_1 + B_1' + C_1 = 1$ implies the common value is $1/3$.

Now the second parts of (i) and (ii) imply that every edge of the entire example has value $1/3$. □

We have shown that $\vec{x}$ is the unique extreme point for the generic example.

k-ECSS Example

We show for any $k \geq 1$ the mincost k-ECSS problem can give rise to any generic example. We start by showing how to construct the graph G for the k-ECSS problem corresponding to a given generic example. As an overview, G will consist of "base edges" and "generic edges", the latter being those of the generic example. The extreme point $\vec{x}$ will have x_e equal to its upper bound (i.e., the weight of e) for the base edges and $x_e = 1/3$ for the generic edges.

As illustrated in Fig.3, G has six vertices for each elementary module and one vertex X for each compound module (X is as illustrated in Fig2(b)). There is a slight difference in how elementary and compound modules are joined to their parents. For that reason Fig.3(a)–(b) show both the elementary module and the vertex X of its parent. Similarly Fig.3(c)–(d) assume that the two children of the compound module are themselves compound modules. (As a minor point these figures assume a compound module has two children that are either both compound or both elementary. For convenience we restrict ourselves to this case.)

An elementary module and vertex X of its parent gives rise to the subgraph illustrated in Fig.3(a) for $\mathcal{O}$-modules and (b) for $\mathcal{I}$-modules. The base edges are on the left and the generic edges are on the right. The correspondence between those generic edges and Fig.2(a) is clear.

Each compound module (Fig.2(b)) with two compound children gives rise to the subgraph of Fig.3(c)–(d). The two root modules (Fig.2(c)) give rise to the subgraph of Fig.3(e). In all cases X_1 and X_2 are the X vertices of the two child modules. The generic edges incident to the subgraphs of the child modules are identified exactly as indicated in Fig.2(b)–(c).

Note that when $k > 1$ some base edges are parallel with generic edges: X_1X and X_2X in Fig.3(c) and XX_1, XX_2 in (d). We could eliminate parallel edges by, e.g., introducing more vertices. For simplicity we keep the parallel edges.

The vector $\vec{x}$ that we have specified makes the sets of $\mathcal{L}$ in the generic example tight. To verify this first consider an elementary module (Fig.2(a)). The two leaf sets of required in-degree one in $\mathcal{L}$ correspond to $\{v\}$ and $\{v'\}$ in Fig.3(a)–(b). These sets both have in-degree $k-1$ *wrt* the integral edges of $\vec{x}$ (i.e., the base edges). This leaves a required in-degree of one for the generic edges, as desired. Similarly the remaining sets of Fig.2(a) correspond to $\{w\}$, $\{w'\}$, $\{v, w, \mu\}$, $\{v', w', \mu'\}$ and $\{v, w, \mu, v', w', \mu'\}$ of Fig.3(a)–(b). The correspondence of Fig.2(b) with Fig.3(c), and Fig.2(c) with Fig.3(d), is also similar.

To complete the claim that $\vec{x}$ is an extreme point, we must show it is feasible to (LP), i.e., no set separating two vertices has in-degree $< k$ when edge weights are given by $\vec{x}$. We also need to

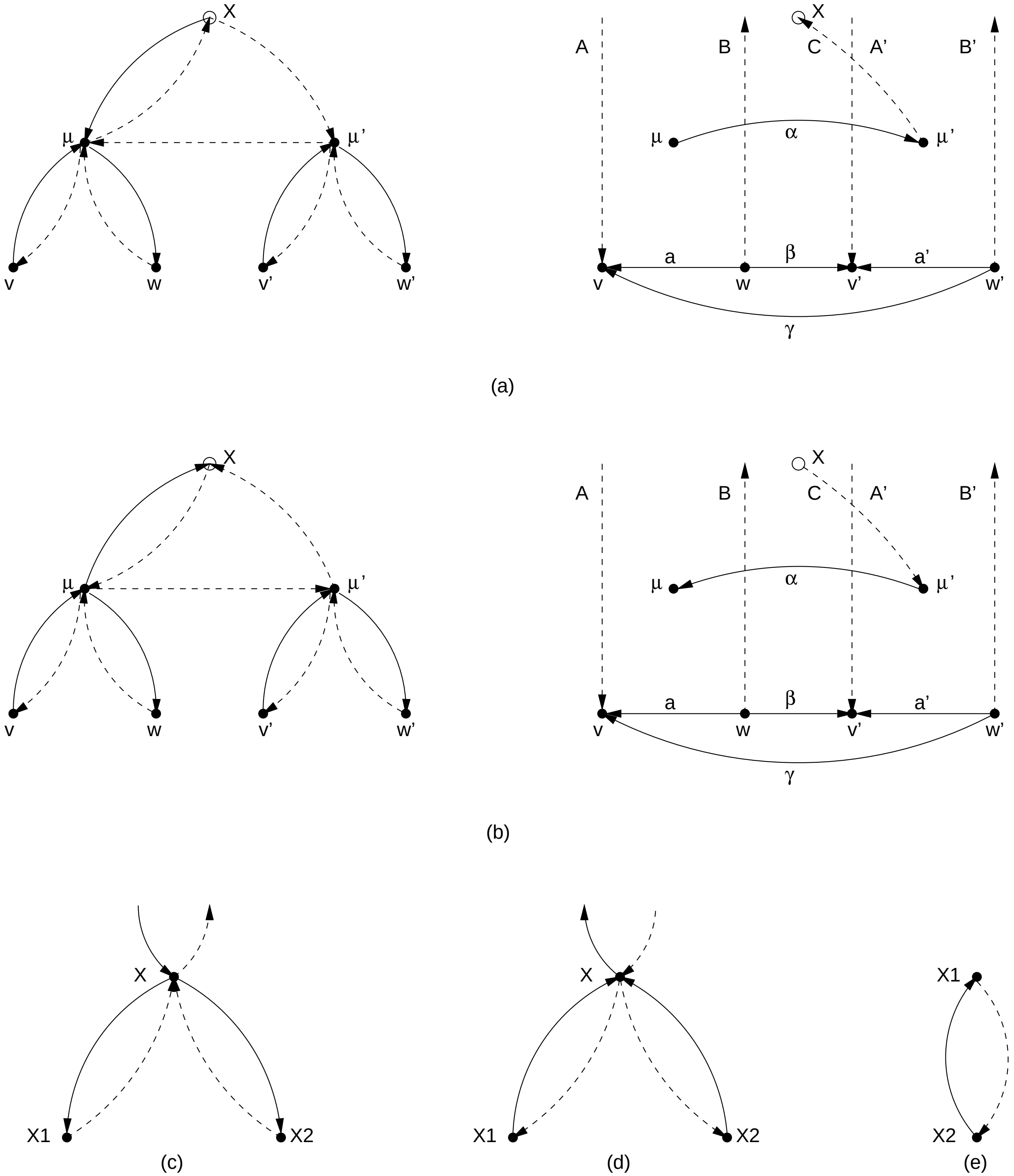

Figure 3: Graph for a generic example. (a) Subgraph for elementary $\mathcal{O}$-module and its parent vertex X. X is drawn hollow as it is in a different module. (b) Subgraph for elementary $\mathcal{I}$-module and its parent. (c) Subgraph for compound $\mathcal{O}$-module with two compound children. (d) Subgraph for compound $\mathcal{I}$-module with two compound children. (e) Subgraph for the root modules. The generic edges for elementary modules are in the right graphs of (a) and (b); generic edges for compound modules and the roots are in Fig.2(b)–(c). All generic edges have weight 1. Base edges are in the left graphs of (a) and (b), and the graphs of (c)–(e). Base edges have weight either k (drawn solid) or $k-1$ (drawn dashed).

exhibit a feasible integral solution $\vec{y}$ to (LP). ($\vec{y}$ will be a minimum cost k-ECSS.) Define y_e to be the upper bound on all base edges and on all A, B, A', B', C' edges. Set y_e to 0 on the remaining edges $(a, a', \alpha, \beta, \gamma)$.

Partition the base edges into two sets: F_0 consists of all base edges chosen up to weight $k-1$ and F_1 consists of the remaining base edges, i.e., F_1 contains one copy of each weight k base edge. F_0 includes a collection of length 2 or 3 cycles that are weakly connected and span the entire graph. Thus F_0 is $(k-1)$-edge connected. Since $\vec{x}$ and $\vec{y}$ both contain all base edges it suffices to show that these vectors, with F_0 deleted, are both strongly connected. For the rest of the feasiblity proof assume that $\vec{x}$ and $\vec{y}$ refer to these vectors with F_0 deleted. So the only base edges that remain are those of F_1, the solid base edges of Fig.3.

The elementary $\mathcal{O}$-modules have a natural left-to-right order, since they are the leaves of a binary tree. When discussing the $\mathcal{O}$-modules we use M_L and M_R to denote the leftmost and rightmost elementary $\mathcal{O}$-modules, respectively. The same applies to $\mathcal{I}$-modules, and we use M_L and M_R similarly when discussing them.

The following lemma will complete the proof that $\vec{y}$ is strongly connected and so feasible to (LP).

Lemma 4.3 *F_1 plus all the edges A, B, A', B', C form a strongly connected spanning subgraph H of G.*

Proof: First notice in Fig.2(a)–(b) that the A, B edges of any elementary module are joined in H by the base edge path v, μ, w. Similarly edges A', B' are joined by v', μ', w'.

The five edges at the root (Fig.2(c)) are part of a cycle R. To see this observe that edges A_1, B_1 at the root come from the same elementary module (M_L) and so are joined by a path v, μ, w. Similarly for the other pairs of edges at the root $(A_2, B_2;\ A'_1, B'_1;\ A'_2, B'_2)$. Hence the five edges in the root module are part of a length 13 path from C_1 to C_2. The base edge from C_2 to C_1 (Fig.3(e)) completes the cycle R.

Now we show that every vertex of G has paths to and from R. This is clear for a vertex X of a combined module: C edges (Fig.2(b)) provide the path in one direction and base edges (Fig.3(c)–(d)) in the other.

It remains to consider the vertices in elementary modules. Let M be an elementary module distinct from M_L and let M' be the elementary module immediately to its left. Fig.2(b) shows that edges A, B of M are identified with A', B' of M' (in the compound module that is the least common ancestor of leaves M and M'). So vertices v, μ, w of M and v', μ', w' of M' form a length six cycle in H. Edges between X vertices and μ, μ' complete paths that join this cycle to and from R. (Specifically for $\mathcal{O}$-modules use base edge $X\mu$ from M and C edge $\mu' X$ from M'. For $\mathcal{I}$-modules use base edge μX from M and C edge $X\mu'$ from M'.)

This argument accounts for all vertices of elementary modules except v, μ, w of M_L and v', μ', w' of M_R. But these vertices belong to the cycle R. We conclude H is strongly connected. □

Now we show $\vec{x}$ is feasible to (LP) too.

Lemma 4.4 *$\vec{x}$ (with weights decreased by F_0) is strongly connected.*

Proof: Say that two vertices s and t are *strongly connected* if there are st- and ts-flows of value one. We will repeatedly choose two vertices s and t, show they are strongly connected, and then contract s and t. Contracting the graph to one vertex will prove that $\vec{x}$ is strongly connected.

We start by defining a pair of cycles for the $\mathcal{O}$-modules and a similar pair for the $\mathcal{I}$-modules. For the $\mathcal{O}$ pair ($\mathcal{I}$ pair) the cycles traverse all vertices of the elementary $\mathcal{O}$-modules ($\mathcal{I}$-modules). Furthermore the only edges common to both cycles of a pair are base edges.

The "rightward cycle", denoted Y_R for both $\mathcal{O}$ and $\mathcal{I}$, traverses the modules in left-to-right order. For both $\mathcal{O}$ and $\mathcal{I}$, Y_R traverses each elementary module along the path

$$(A)\, v, \mu, w\ (\beta)\, v', \mu', w' (B')$$

where the last edge B' is edge A of the next module (Fig.2(b)). The rest of Y_R is a path from B' of M_R to A of M_L. We define it first for $\mathcal{O}$ and then for $\mathcal{I}$. For $\mathcal{O}$ the path is in the $\mathcal{I}$-module. In terms of Fig.2(c) this path must go from $B'_1 = C_2$ to $A_1 = B'_2$. This path consists of C edges in the $\mathcal{I}$-module (Figs.2(b) and 3(b)), ending in edge $\mu' w'$ of M_R. For $\mathcal{I}$ the path is in the $\mathcal{O}$-module. In terms of Fig.2(c) this path must go from $B'_2 = A_1$ to $A_2 = B_1$. This path is simply v, μ, w in the $\mathcal{O}$-module M_L.

The "leftward cycle" Y_L is similar. It traverses each elementary module along the path

$$(A')\, v', \mu', w'\ (\gamma)\, v, \mu, w\ (B)$$

where the last edge B is edge A' of the next module (Fig.2(b)). The rest of Y_L is a path from B of M_L to A' of M_R. For $\mathcal{O}$ the path is in the $\mathcal{I}$-module. In terms of Fig.2(c) this path goes from $B_1 = A_2$ to $A'_1 = B_2$. So the path is simply v, μ, w in the $\mathcal{I}$-module M_L. For $\mathcal{I}$ the path is in the $\mathcal{O}$-module. In terms of Fig.2(c) the path goes from $B_2 = A'_1$ to $A'_2 = C_1$. This path consists of edge $v'\mu'$ of M_R followed by C edges in the $\mathcal{O}$-module (Figs.3(a) and 2(b)).

Observe that as claimed, Y_L and Y_R for $\mathcal{O}$ have distinct generic edges. This includes the edges of the root module. The same holds for $\mathcal{I}$.

We now show in steps (a)–(d) that the vertices of the elementary module M can be contracted into its parent vertex X. The arguments of (a) and (b) apply to both $\mathcal{O}$- and $\mathcal{I}$-modules M.

(a) v and w are strongly connected: The vw-flow is just the base edge path v, μ, w. The wv-flow routes 1/3 unit from w to v along each of the cycles Y_L, Y_R and 1/3 unit along edge a. Similarly v' and w' are strongly connected.

(b) In $G/\{v, w\}$ the contracted vertex $\{v, w\}$ has base edges to and from μ. Hence we can contract $\{v, w, \mu\}$. Similarly for $\{v', w', \mu'\}$.

(c) The contracted vertices $\{v, w, \mu\}$ and $\{v', w', \mu'\}$ are strongly connected: For simplicity refer to these vertices as μ and μ' respectively. Suppose M is an $\mathcal{O}$-module, $\mathcal{I}$-modules are symmetric. Both flows between μ and μ' use the two cycles Y_L and Y_R. The third path is provided by edge α for the $\mu\mu'$-flow and path μ', X, μ for the $\mu'\mu$-flow. Neither α nor edge $\mu' X = C$ belongs to either cycle, and $X\mu$ is a base edge. So the flows are valid.

(d) Now form the contracted vertex $\Lambda = \{v, w, \mu, v', w', \mu'\}$. Λ is strongly connected with its parent X: First assume M is an $\mathcal{O}$-module. An $X\Lambda$-flow is provided by the base edge $X\mu$. The ΛX-flow routes 1/3 unit on edge $\mu' X$ and 1/3 unit on two other paths, constructed from Y_L (Y_R) and base edges. For Y_R, follow that path from Λ to $B'_1 = C_2$. Then use base edges to get to C_1 (Fig.3(e)) and then the goal vertex X (Fig.3(c)). For Y_L, follow that path from Λ to $B_1 = A_2$. Follow base edges of the $\mathcal{I}$-module to get to C_2 (Fig.3(d)); then follow the same path as Y_R.

$\mathcal{I}$-modules are symmetric: In the $X\Lambda$-flow the paths constructed from Y_L and Y_R both start by following base edges from X to the root $\mathcal{O}$-module's vertex X (Fig.2(d), (e)). Then following edge $C_1 = A'_2$ leads to Y_L. Following base edges in the $\mathcal{O}$-module take us from the root X to $B_1 = A_2$, which is the start of Y_R.

Now consider a compound module. Assume all descendants of X_1 have been contracted into X_1. X and X_1 are strongly connected by exactly the same argument as in (d).

Finally when the graph has been contracted to two vertices, Figs.2(c) shows that the vertices are joined by three generic edges in both directions. □

To complete the definition of the k-ECSS problem we specify the cost function. Say that an edge of an elementary module M is *costly* (*wrt* M) if it is labelled $a, a', \alpha, \beta, \gamma, A'$ or B'. Observe that an edge labelled A' or B' at one of its ends is labelled differently at its other end – the second end is either A or B (Fig.2(b)) or A, B or C (Fig.2(c)). This implies that any edge is costly for only one elementary module. (Edge $B_2 = A_1$ of Fig.2(c) is not costly – it is the only edge with an end labelled A or B that is not costly.) Now define the cost of an edge to be 1 if it is costly (for any elementary module). All other edges have cost 0.

Letting e denote the number of elementary modules, any feasible solution to (LP) costs $\geq 2e$. In proof first note that in an elementary $\mathcal{O}$-module edge $w'\mu'$ is the only cost 0 edge leaving w', and $X\mu'$ is the only cost 0 edge entering $\{v', w', \mu'\}$. Both of these cost 0 edges have weight $k-1$. Hence any feasible solution contains costly edges of weight ≥ 1 leaving w', as well as costly edges of weight ≥ 1 entering $\{v', w', \mu'\}$. Thus each elementary $\mathcal{O}$-module increases the cost by 2. The same holds for each elementary $\mathcal{I}$-module, where we look at the edges entering v' and those leaving $\{v', w', \mu'\}$. This gives total cost $\geq 2e$.

Both $\vec{x}$ and $\vec{y}$ have costly edges of weight exactly 1 leaving w' and entering $\{v', w', \mu'\}$ in an $\mathcal{O}$-module, and entering v' and leaving $\{v', w', \mu'\}$ in an $\mathcal{I}$-module. Thus both vectors have total cost $2e$ and both vectors are optimum for (LP). If our algorithm finds the extreme point $\vec{x}$ it can round up all costly edges, giving an integral solution of cost $6e$. This gives the claimed the approximation ratio of 3.

Finally we note that it is easy to modify the example to make $\vec{x}$ the unique extreme point. In detail, choose a value $\epsilon > 0$. Increase the cost function by ϵ times the in- or out-degree of each set of $\mathcal{L}$. For instance for $\{v\}$ of an elementary module add $\epsilon A + \epsilon a + \epsilon\gamma$ to the cost function. Lemma 4.2 shows that for any $\epsilon > 0$, $\vec{x}$ is the unique optimum for the modified LP. So the algorithm finds a k-ECSS of cost $> 6e$. The cost of the integral solution $\vec{y}$ approaches $2e$ as $\epsilon \to 0$. (Specifically, it is easy to see that $|\mathcal{L}| = e + (e-2) = 2e-2$, and any set of $\mathcal{L}$ has degree ≤ 3 *wrt* $\vec{y}$. Thus $\vec{y}$ has cost $\leq 2e + (2e-2)3\epsilon$.)

5 Mixed Graphs

This section examines supermodular cut problems when the graph is mixed, i.e., both undirected and directed edges are allowed. To put the results in perspective note that any algorithm with approximation ratio r for directed supermodular cut implies an approximation ratio of $\leq 2r$ on mixed graphs: The mixed algorithm simply models each undirected edge of cost c by two antiparallel edges of cost c. Although one might hope that iterated rounding on mixed graphs would perform at least as well, our main result suggests this is not true:

Theorem 5.1 *For any integer $k > 1$, an infinite family of instances of the minimum cost k-EC augmentation problem on mixed graphs has $\vec{\frac{2}{n}}$ as the unique optimum solution to (LP). On these instances round-the-highest has approximation ratio $n/2$.*

The section ends with two more positive results for the special cases $k = 1$ and all edges undirected, respectively.

We sketch the proof of Theorem 5.1. As in Section 4 we start with a generic example, illustrated in Fig.4(a)–(c). There are $n = 2h$ vertices, with h vertices $v_1, \ldots, v_h$ on the left and a symmetric group of vertices $w_1, \ldots, w_h$ on the right. Vertex v_1 (w_1) is joined to every other vertex on the left

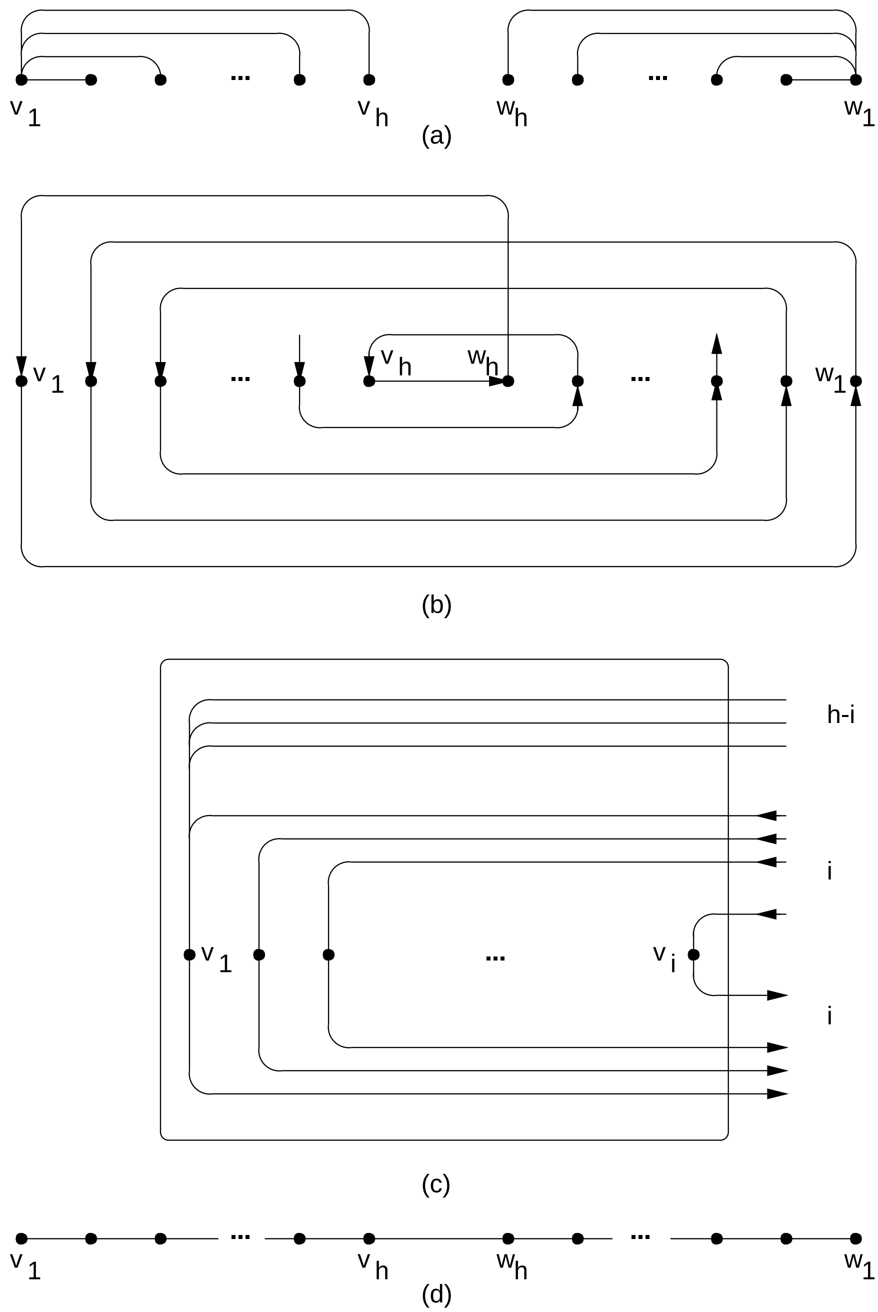

Figure 4: Mixed graph example. (a) Generic undirected edges. (b) Generic directed edges. All generic edges have weight 1. (c) Typical set of $\mathcal{L}$ belongs to $\mathcal{I} \cap \mathcal{O}$, drawn as rounded rectangle. The set consists of the first i vertices. There are $h+i$ incident generic edges. Of these, $h-i$ are undirected, i are directed into the set and i leave the set. (d) Base edges. All base edges have weight $k-1$.

(right) by an undirected edge (Fig.4(a)). There is a cycle of directed edges, each edge joining a left vertex with a right (Fig.4(b)). For each i, $1 \le i \le h$, $\{v_1, \ldots, v_i\}$ ($\{w_1, \ldots, w_i\}$) belongs to $\mathcal{I} \cap \mathcal{O}$ with in-degree and out-degree both required to be 1 (Fig.4(c)).

Observe that $\vec{\frac{1}{h}}$ is the unique solution for $\mathcal{L}$. In proof, this vector clearly satisfies all degree constraints. For uniqueness, the two directed edges incident to v_1 must have the same weight, since $\vec{\rho}(\{v_1\})$ and $\vec{\delta}(\{v_1\})$ are identical except for those two edges. Repeating this argument for increasing i shows the two directed edges incident to each v_i have the same weight (compare the two boundaries of $\{v_1, \ldots, v_i\}$). The same holds for the two directed edges at w_i. Hence the cycle of directed edges forces all directed edges to have the same weight. The in-degree constraint for $\{v_1, \ldots, v_h\}$ show that weight is $1/h$. Finally comparing the in-degrees of $\{v_1, \ldots, v_{i-1}\}$ and $\{v_1, \ldots, v_i\}$ shows each undirected edge has weight $1/h$.

When $k > 1$ we can model the generic example as a mincost k-EC augmentation problem. The base edges form a Hamiltonian path as shown in Fig.4(d), each edge having weight $k - 1$. (This introduces two copies of edges v_1v_2 and w_1w_2. We could easily avoid this by changing each of the two base edges to two antiparallel directed edges.) All directed edges cost 1 and all undirected edges cost 0.

For this problem any solution to (LP) costs ≥ 2 because of the in-degree constraints for $\{v_1, \ldots, v_h\}$ and $\{w_1, \ldots, w_h\}$. Define $\vec{x}$ to be $k - 1$ on all base edges and $1/h$ on all generic edges. Define $\vec{y}$ to be $k - 1$ on all base edges and 1 on undirected edges v_1v_h, w_1w_h and directed edges v_1w_1, w_hv_1. It is easy to see that both vectors cost 2. Also both vectors are feasible (i.e., they make the graph k-edge connected) because of the assumption $k > 1$. $\vec{x}$ is an optimum extreme point, and once we fix its integral components we get the vector $\vec{\frac{1}{h}}$ as an optimum extreme point. As usual a perturbation of the cost function makes it the unique optimum.

Our example leaves open the case $k = 1$. Perhaps surprisingly in this case iterated rounding does better than doubling the approximation ratio:

Theorem 5.2 *For any minimum cost 1-EC augmentation or internal augmentation problem on mixed graphs, any nonzero extreme point to* (LP) *has* L_∞*-norm* $\ge 1/4$. *This bound is tight on an infinite family of instances.*

The theorem is proved in the Appendix.

Finally consider the supermodular directed cut problem when the given graph G is undirected. This can model mixed graph problems such as *mincost k-EC undirected-edge augmentation for digraphs*. Jain's algorithm [8] provides a 2-approximation. To see this recall that a function $f : 2^V \to \Re$ is *weakly supermodular* if

$$(10) \qquad f(A) + f(B) \le \max\{f(A \cap B) + f(A \cup B),\ f(A - B) + f(B - A)\}$$

for any two sets $A, B \subseteq V$. Jain's iterated rounding algorithm gives a 2-approximation for supermodular directed cut when f is weakly supermodular and G is undirected. So the applicability of Jain's algorithm is guaranteed by the next proposition. For this proposition $\overline{A}$ denotes the complementary set $V - A$.

Proposition 5.3 *Let f be a crossing supermodular function on 2^V. Then the function g defined by $g(A) = \max\{f(A), f(\overline{A})\}$ is weakly supermodular.*

Proof: When h is symmetric ($h(A) = h(\overline{A})$ for all A) inequality (10) for h and sets A and B implies (10) for sets A and $\overline{B}$ as well as $\overline{A}$ and $\overline{B}$. Now consider g and sets A, B. If A and B do not cross then, by possibly complementing A and/or B, we can assume A and B are disjoint.

Then trivially $g(A) + g(B) = g(A \cap B) + g(A \cup B)$. If A and B cross, by possibly complementing A and/or B we can assume $g(A) = f(A)$ and $g(B) = f(B)$. Then crossing supermodularity gives $g(A) + g(B) \leq f(A \cap B) + f(A \cup B) \leq g(A \cap B) + g(A \cup B)$. $\square$

References

[1] J. Cheriyan and S. Vempala, *Edge covers of setpairs and the iterative rounding method*, Proc. 8th International Integer Programming and Combinatorial Optimization Conf., (2001), pp.30–44.

[2] J. Cheriyan, S. Vempala and A. Vetta, *Approximation algorithms for minimum-cost k-vertex connected subgraphs*, Proc. 34th Annual ACM Symp. on Theory of Comput., (2002), pp.306–312.

[3] A. Frank, *Kernel systems of directed graphs,* Acta Sci. Math., Szeged, Hungary, 41 (1979), pp.63–76.

[4] A. Frank, *Submodular functions in graph theory,* Discrete Math., 111 (1993), pp.231–243.

[5] G.N. Frederickson and J. Ja'Ja', *Approximation algorithms for several graph augmentation problems,* SIAM J. Comput., 10, 2 (1981), pp.270–283.

[6] L. Fleischer, K. Jain and D.P. Williamson, *Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems,* submitted for publication; union of *An iterative rounding 2-approximation algorithm for the element connectivity problem*, Proc. 42nd Annual IEEE Symp. on Foundations of Comp. Sci., (2001), pp.339–347, and *A 2-approximation for minimum cost $\{0, 1, 2\}$ vertex connectivity*, Proc. 8th International Integer Programming and Combinatorial Optimization Conf., pp.115–129.

[7] H.N. Gabow, M.X. Goemans, E. Tardos and D.P. Williamson, *Approximating the smallest k-edge connected spanning subgraph by LP-rounding*, Proc. 16th Annual ACM-SIAM Symp. on Disc. Algorithms, 2005, to appear.

[8] K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica 21 (2001), pp. 39–60.

[9] S. Khuller and U. Vishkin, *Biconnectivity approximations and graph carvings*, J. ACM 41, 2 (1994), pp.214–235.

[10] G. Kortsarz and Z. Nutov, *Approximation algorithm for k-node connected subgraphs via critical graphs*, Proc. 36th Annual ACM Symp. on Theory of Comput., (2004), pp.138–145.

[11] V. Melkonian and E. Tardos, *Algorithms for a network design problem with crossing supermodular demands*, Networks 43, 4 (2004), pp. 256–265.

[12] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, NY, 2003.

[13] V.V. Vazirani, *Approximation Algorithms*, Springer-Verlag, NY (2001).

Appendix A The 1-EC Augmentation Problem for Mixed Graphs

We begin by proving the first part of Theorem 5.2: For any minimum cost 1-EC augmentation or internal augmentation problem on mixed graphs, any nonzero extreme point to (LP) has L_∞-norm $\geq 1/4$. To restate the problem, we are given a host graph $H = (W, F)$, a subset of vertices $V \subseteq W$, and a set of potential edges E on V with costs. We seek a minimum cost subset $E' \subseteq E$ such that V is 1-edge connected in the graph $(W, E' \cup F)$. The sets E and F are mixed, i.e., they may contain both directed and undirected edges.

Let us also describe the supermodular requirement function f for our problem. The range of f is the set of integers ≤ 1 (where only sets with $f(S) = 1$ give a relevant constraint in (LP)). For any set $S \neq \emptyset, V$,

$$f(S) = 1 \text{ if and only if no path goes from } V - S \text{ to } S.$$

The graph referred to in this characterization is H. When we are solving a 1-EC internal augmentation problem by iterated rounding we can enlarge H by all edges that have been added to the solution so far. The characterization follows easily from the fact that by definition, $f(S) = 1$ exactly when some set $T \subseteq W - V$ has $\rho(S \cup T) = 0$.

As in Section 3 we prove Theorem 5.2 by contradiction. Assume every x_e is $< 1/4$.

We change some basic definitions from Section 3. $\mathcal{L}$ is represented by a forest $\mathcal{F}$, where each set of $\mathcal{L}$ corresponds to one node of $\mathcal{F}$. So unlike Section 3 a set of $\mathcal{I} \cap \mathcal{O}$ has only one node in $\mathcal{F}$. An end of an edge is *available* to the smallest node of $\mathcal{F}$ that contains it. (This is unlike Section 3 but is the same notion as ownership in [8].)

A *t-set* is a set of $\mathcal{L}$ that is the disjoint union of one or more sets of $\mathcal{I} \cap \mathcal{O}$. For example a set of $\mathcal{I} \cap \mathcal{O}$ is the simplest of t-sets. Assume (as in [7]) that the family $\mathcal{L}$ is *light* – it minimizes the total size of all its sets. A set in $\mathcal{I} \cap \mathcal{O}$ contributes twice to the total size. Clearly lightness guarantees that $\mathcal{L}$ is a minimal family, i.e., the constraints of the system (1) are linearly independent.

The following lemma gives the facts that we need about available edges.

Lemma A.1 *Let S be a node with children A_i, $i = 1, \ldots, r$, $r \geq 0$.*

(i) S has ≥ 5 available ends if it is a leaf, and ≥ 6 available ends if it is a leaf in $\mathcal{I} \cap \mathcal{O}$.

S has ≥ 5 available ends if any of the following conditions holds:

(ii) S is an $\mathcal{I}$-set, each A_i is an $\mathcal{O}$-set or the disjoint union of $\mathcal{O}$-sets, and $S \neq \cup A_i$. Symmetrically for S an $\mathcal{O}$-set.

(iii) S is not a t-set but every child A_i is.

(iv) $S \in \mathcal{I} \cap \mathcal{O}$ and every child A_i but possibly one is a t-set.

Proof: *(ii)* No integral edge enters a set of $\mathcal{I}$ or leaves a set of $\mathcal{O}$. Hence $S - \cup A_i$ has no entering integral edge. Since $\vec{x}$ is feasible, the fractional edges entering $S - \cup A_i$ weigh ≥ 1. So if ≤ 4 edges enter then some $x_e \geq 1/4$.

(i) Part *(ii)* shows S has ≥ 5 available ends. Suppose $S \in \mathcal{I} \cap \mathcal{O}$. At least one directed edge e is incident to S, since the vectors $\vec{\rho}(S)$ and $\vec{\delta}(S)$ are distinct. Assume *wlog* that e enters S. Part *(ii)* gives an additional 5 edges that leave S.

(iii) $S \neq \cup A_i$ since equality makes S a t-set. Each A_i is the disjoint union of sets of $\mathcal{I} \cap \mathcal{O}$. Now part *(ii)* applies to S.

(iv) S satisfies the first two hypotheses of *(ii)*. (For the second hypothesis we can assume by symmetry that the child which is not a t-set belongs to $\mathcal{O}$.) So we need only show $S \neq \cup A_i$. We claim a slightly more general statement: A node $S \in \mathcal{I} \cap \mathcal{O}$ cannot be written as the disjoint union of sets B_i, $0 \leq i \leq r$, where $B_i \in \mathcal{I} \cap \mathcal{O}$ for $i \geq 1$ and $B_0 \in \mathcal{L}$.

To prove the claim assume that S is partitioned into the sets B_i. This makes the vectors $\vec{\delta}(S), \vec{\rho}(S), \vec{\delta}(B_i)$ and $\vec{\rho}(B_i)$ linearly dependent, by Proposition 3.6(ii). Hence $B_0 \in \mathcal{I} \oplus \mathcal{O}$, since the constraints of the system (1) are linearly independent.

In the rest of this argument say that a set S is *tight* if its constraint in (LP) is satisfied with equality for our extreme point $\vec{x}$. (Recall that a set $S \in \mathcal{I}$ is necessarily tight, and a set $S \in \mathcal{O}$ has $V - S$ tight.) Assume momentarily that B_0 and $V - B_0$ are both tight. Form the family $\mathcal{L}'$ by starting with $\mathcal{L}$ and removing S from $\mathcal{O}$ and adding B_0 to the family $\mathcal{I}$ or $\mathcal{O}$ that doesn't currently contain it. Let (1) continue to refer to the linear system of Lemma 3.2 using our laminar family $\mathcal{L}$. Form a new linear system ($1'$) that corresponds to $\mathcal{L}'$, where the two constraints for B_0 in ($1'$) are equivalent to tightness of B_0 and $V - B_0$ respectively. Our assumption implies $\vec{x}$ remains a solution to ($1'$). ($1'$) has only one solution (since (1) has a unique solution, and the constraint vector $\vec{\delta}(S)$ in (1) is spanned by the constraint vectors of ($1'$), by Proposition 3.6(ii)). So $\vec{x}$ is the unique solution to ($1'$). This makes $\mathcal{L}'$ a valid choice of laminar family for $\vec{x}$. But this contradicts the lightness of the family $\mathcal{L}$.

It remains only to verify our assumption that B_0 and $V - B_0$ are tight. Each set $S, V - S$, B_i and $V - B_i$ $(i > 0)$ is tight since $S, B_i \in \mathcal{I} \cap \mathcal{O}$, $i > 0$. Suppose $B_0 \in \mathcal{O}$ and we need to show that B_0 is tight. (When $B_0 \in \mathcal{I}$ we need to show that $V - B_0$ is tight – this argument is symmetric). Proposition 3.6(ii) gives $\delta(S) - \rho(S) = \sum_{i=0}^{r} \delta(B_i) - \rho(B_i)$ when all in- and out-degrees are computed using weights given by the edges of host graph F plus those of $\vec{x}$. For these weights all terms are exactly 1 except for $\rho(B_0)$ which is unknown. So this implies $\rho(B_0)$ also equals 1. To show B_0 is tight we need only show $f(B_0) = 1$.

Our opening characterization of f shows that we want to prove that no vertex $s \in V - B_0$ can reach a vertex of B_0 (in graph H). We proceed by contradiction. Any such s must belong to S, since $f(S) = 1$. Hence s belongs to some set $B_i, i \neq 0$. Now s shows $f(V - B_i) \neq 1$. But $B_i \in \mathcal{O}$ implies $f(V - B_i) = 1$. □

As before classify the nodes of $\mathcal{F}$ as leaf, chain or branching. (A node of $\mathcal{I} \cap \mathcal{O}$ can be any of these types.)

We now show how to pay for all the $\mathcal{L}$-sets and derive the desired contradiction. We begin by assigning two edge ends to every node of $\mathcal{F}$. Specifically, each chain node S has 2 available ends as before (by Lemma 3.5, since with the new definition of $\mathcal{F}$ we can consider S to be a 1-node). Assign these ends to S itself. Each leaf node has 5 available ends (Lemma A.1(i)). We use 4 of them, and assign 2 ends to each leaf and 2 ends to each branching node. Just as before there are ≥ 2 ends from leaves still unassigned (Lemma 3.4) which we set aside for the contradiction.

To complete the proof we need only assign 2 more ends to each node $S \in \mathcal{I} \cap \mathcal{O}$. We will use the following rule:

($*$) Assign any end that is currently unassigned to the smallest set of $\mathcal{I} \cap \mathcal{O}$ that contains it.

Note that a node of $\mathcal{I} \cap \mathcal{O}$ will be assigned any ends available to it, and possibly others. ($*$) does not apply to the 2 ends from leaves that were set aside.

To show that ($*$) pays for the remaining sets $S \in \mathcal{I} \cap \mathcal{O}$ we examine three cases:

Case 1. S a leaf. Lemma A.1(i) shows S began with 6 available ends, of which only 4 were assigned (or put aside). ($*$) assigns the extra 2 to S.

Case 2. S a chain node. Lemma A.1(ii) shows S began with 5 available ends (since whichever family its child is in, S belongs to the other). Only 2 ends were assigned. ($*$) assigns the extra 3 to S, although only 2 are needed.

Case 3. S a branching node. Let the children of S be A_i. If all but at most one child is a t-set Lemma A.1(iv) shows S began with 5 available ends. None were assigned and ($*$) assigns them all to S.

We can now assume ≥ 2 children of S are not t-sets. Let A_i be such a child. Let B_i be a maximal descendant of A_i that is not a t-set, but all its children are. (Maximality refers to viewing B_i as a vertex set. In particular note that maximality makes S the first ancestor of B_i that belongs to $\mathcal{I} \cap \mathcal{O}$. B_i exists since a leaf has no children.) Lemma A.1(iii) shows B_i began with 5 available ends. At most 4 of these were assigned (if B_i is a leaf; for a chain (branching) node only 2 (0) were assigned, respectively). ($*$) assigns the extra end to S since as noted, S is the first ancestor of B_i in $\mathcal{I} \cap \mathcal{O}$. Since there are two sets B_i, S gets the necessary 2 ends. This concludes the proof of the first part of Theorem 5.2.

Tight Example

We sketch the tight example for Theorem 5.2. The generic example is illustrated in Fig.5 (for now ignore v and w in Fig.5(a)). It is built up by combining two *simple modules* (Fig.5(a)) into an *elementary module* (Fig.5(b)), and forming *compound modules* (Fig.5(c)) from two smaller elementary or compound modules. We complete the example by forming a *maximal compound module* (Fig.5(d)) from two compound modules, and then forming the entire example (Fig.5(e)) from two maximal compound modules. Every in- or out-degree constraint in $\mathcal{L}$ has value 1.

$\vec{\frac{1}{4}}$ is the unique solution for the edges of the generic example. In proof, first observe that as indicated in Fig.5(a) the 2 undirected edges incident to a simple module must have the same weight a as the internal directed edge. (The right leaf and its parent have the same in-degree, 1. They have the same entering edges except for the internal directed edge and the leftmost undirected edge. Hence these two edges have the same weight.) The elementary module of Fig.5(b) is composed of two simple modules which are forced to have the same value of a (because of the internal undirected edge). The 2 edges labelled α are then forced to have the same weight α (since the leftmost and rightmost leaves have the same degree constraint, 1). Similarly for the 2 edges labelled β. The same logic forces the labels in Fig.5(c). In Fig.5(d) the two constituent modules are forced to have the same value for a, but the weights α', β' of the right module need not be the same as α, β. In Fig.5(e) the 8 edges are arranged in a cyclic fashion with consecutive edges equal (e.g., the two edges of weight α). This forces all 8 edges to have the same weight. Since the two modules in Fig.5(e) belong to $\mathcal{I}$ this common weight must be 1/4. It is now easy to see that $a = 1/4$ (e.g., from the dashed edges of Fig.5(c)). This also shows that $\vec{\frac{1}{4}}$ is feasible, i.e., all degree constraints of $\mathcal{L}$ are satisfied.

To convert the generic example into a mincost 1-ECSS or 1-EC augmentation problem problem we modify each simple module (Fig.5(a)) as follows. Change each of the two leaves into singleton sets. So (as in Fig.5(a)) we have two vertices v, w joined by the internal directed edge vw. Add the reverse edge wv as a base edge. Every generic edge costs 1 and every base edge costs 0.

Let n be the number of vertices in the example graph. Then $n/2$ lowerbounds the optimum of (LP). ($\mathcal{O}$ contains $n/2$ singleton sets, each requiring outgoing edges of cost 1.) Define $\vec{x}$ to be 1 on every base edge and 1/4 on every generic edge. Define $\vec{y}$ to be 1 on every base edge, every undirected edge, and on 2 directed edges at the root module, one in each direction. The cost of these two vectors is $n/2$ and $n/2 + 2$, respectively. Assume for the moment that both vectors are feasible to (LP). Then $\vec{x}$ is an optimum solution and $\vec{y}$ upperbounds the optimum integral solution. Applying round-the-highest to $\vec{x}$ gives a solution of cost $2n$. Hence the approximation ratio is $\geq 2n/(n/2 + 2)$, which approaches 4 as $n \to \infty$.

It is easy to see that $\vec{y}$ is feasible, i.e., it gives a strongly connected graph: A simple induction

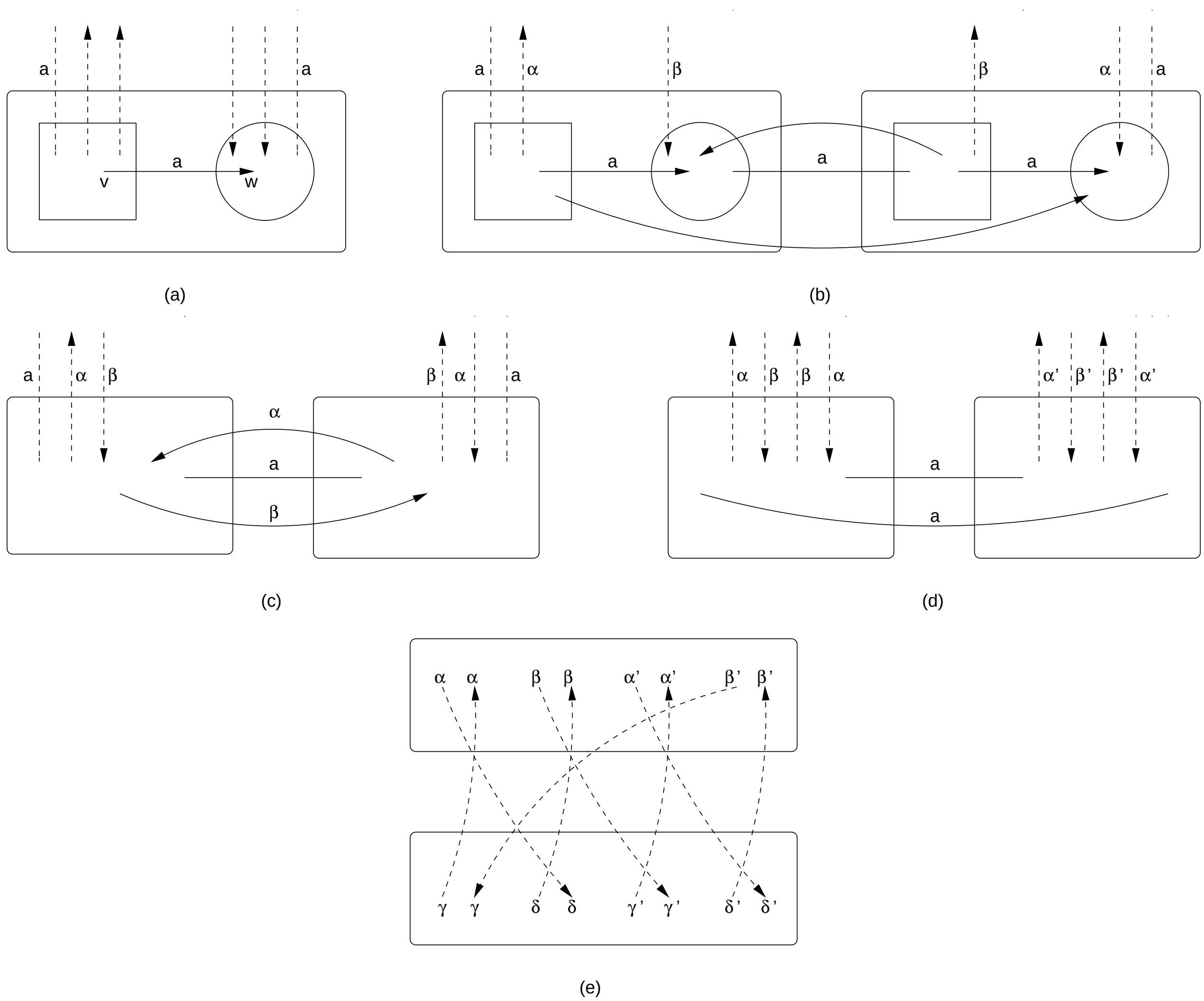

Figure 5: Generic example for 1-EC mixed graph augmentation. Edges incident to a module are drawn dashed. (a) Simple module. (b) Elementary module. (c) Compound module. (d) Maximal compound module. (e) Root module.

shows that every simple, elementary and compound module has a Hamiltonian path from right to left. Hence each maximal compound module has a Hamiltonian cycle. The 2 directed edges at the root tie these cycles together into a strongly connected graph.

The proof that $\vec{x}$ is feasible is longer and again we only sketch it. First some terminology. Let M range over the two maximal compound modules (Fig.5(d–e)). Let M_i range over the two halves of M (Fig.5(d)). In an elementary module Fig.5(b) label the 2 currently unlabelled edges just like Fig.5(c): the upper, right-to-left edge is an α edge and the lower, left-to-right edge is a β edge. Define the following paths (CL has already been used in the previous paragraph).

> CR: a "rightward" Hamiltonian cycle through M, formed by the a-edges;
> CL: a "leftward" Hamiltonian cycle through M, formed by the undirected edges and the base edges;
> $P\alpha$: a path through all vertices of M_i, formed by the α edges and the base edges;
> $P\beta$: a path through all vertices of M_i, formed by the β edges and the base edges.

We start by showing vertices v and w of a simple module are strongly connected. Considering base edge wv, we need only exhibit a vw-flow of value 1. Switch now to an elementary module E, with vertices v_1, w_1 in the left half and v_2, w_2 in the right half. We will exhibit a v_iw_i-flow for $i = 1, 2$. We will form the flow from 4 paths, $P_1, \ldots, P_4$, that are edge-disjoint except for base edges.

P_1 is the generic edge v_iw_i. P_2 consists of 3 edges internal to E (specifically the α and β edges and the base edge w_jv_j, $j \neq i$). P_3 is $CL - w_iv_i$. For P_4 assume E is contained in the maximal compound module M and let M' be the other maximal compound module. Then P_4 for $i = 1$ starts at v_1 and follows $P\alpha$ until it enters M' (along an edge of Fig.5(e)). It then follows CL (of M') to the edge of $P\beta$ that enters M from M'. It then follows $P\beta$ to w_1. P_4 for $i = 2$ is similar, but beginning on $P\beta$ and ending on $P\alpha$.

Next consider a compound module C (Fig.5(c) and assume its two submodules have been contracted to single vertices, say C_1 on the left and C_2 on the right. (A special case is when C is an elementary module and we have contracted v_1, w_1 and also v_2, w_2.) We will show C_1 and C_2 are strongly connected. C_1 and C_2 are joined by 2 one-edge paths P_1, P_2 inside C in each direction (using edges a, α, β). P_3 is provided by the C_1C_2-subpath of CL or the C_2C_1-subpath of CR. P_4 is constructed like the previous case, except that the C_1C_2-path (C_2C_1-path) begins and ends on $P\alpha$ ($P\beta$) respectively.

Finally suppose that the graph has been contracted into 4 vertices, specifically the two halves of the two maximal modules (Fig.5(d)). It is easy to check this graph is strongly connected by examining every possible cut. For instance the "diagonal" cut $(\{M_1, M_2'\}, \{M_1', M_2\}$, where the top module (bottom module) of Fig.5(e) has left half M_1 (M_1') and right half M_2 (M_2')) is crossed by the 4 a-edges.