

Book title goes here

H.Narayanan

June 11, 2009

Foreword

No foreword

Preface

No preface for the chapter.

Contents

1	Hybrid Analysis and Combinatorial Optimization	3
1.1	Introduction	3
1.2	Preliminaries	4
1.3	Topological Hybrid Analysis Procedure	8
1.4	Proofs for Topological Hybrid Analysis	12
1.5	Principal Partition Problem	13
1.6	Building maximally distant forests	15
1.7	Network Analysis through Topological Transformation	20
1.7.1	Solution of the node fusion - fission problem	22
1.8	Notes	24

Chapter 1

Hybrid Analysis and Combinatorial Optimization

1.1 Introduction

In this chapter we discuss the hybrid analysis problem and sketch one of its natural generalizations. Focusing attention on these naturally leads to the study of fundamental combinatorial optimization problems which can be solved using the matroid union operation (see [Narayanan+Patkar09]) and the Dilworth truncation operation (see Section 1.7).

Electrical network analysis is the process of finding pairs of vectors (\mathbf{v}, \mathbf{i}) , such that \mathbf{v} satisfies Kirchhoff's Voltage Law (KVL) for the graph \mathcal{G} of the network, \mathbf{i} satisfies Kirchhoff's Current Law (KCL) for \mathcal{G} and the pair (\mathbf{v}, \mathbf{i}) satisfies the device characteristic of the network. For ease of discussion, we will assume that the network is static (no derivative terms in the constraints) and has a unique solution, i.e., a unique (\mathbf{v}, \mathbf{i}) pair satisfies the constraints of the network. The basic methods of analysis reduce the constraints (KCL, KVL, device characteristic) of the network to a more compact form involving node voltages or loop currents. In the former case, once the node voltages are obtained, by the use of KVL the branch voltage vector can be obtained uniquely and thence using device characteristic, the branch current vector. In the latter case, from the loop currents, the branch current vector can be obtained through use of KCL and thence using device characteristic, the branch voltage vector. A natural generalization of these procedures is to pick as unknowns some voltages and some currents from which by the use of KCL and KVL either the voltage or current of every branch can be obtained, after which the use of device characteristic will enable us to obtain the remaining variable of the branch. This class of methods where the unknowns are a mixture of current and voltage variables is called 'hybrid analysis'.

Hybrid Analysis is originally due to G.Kron ([Kron39],[Kron63]) and was simplified by Branin ([Branin62]). A description very accessible to the general

reader is available in [Brameller+John+Scott69]. The development presented here is however based on a topological version reported in [Narayanan79]. It has the advantage of greater flexibility in the choice of unknowns and also advantages in storage. The manner in which voltage and current variables are chosen in hybrid analysis can be viewed in a general way as a process of transforming the given network through the operations of node fusions and fissions into another simpler network. Such a general transformation can be called ‘topological transformation of electrical networks’ [Narayanan87] and has applications particularly in the parallel processing of network analysis.

Natural questions that arise are on how to choose the unknowns minimally. In the case of hybrid analysis this leads to the question of ‘Principal Partition’ and in the more general case of topological transformations this leads to the ‘Principal Lattice of Partitions’ problem. The former is discussed in Section 1.5 and the latter is sketched in Section 1.7.

1.2 Preliminaries

We need a few preliminary definitions and results before we move on to a discussion of the methods.

We assume familiarity with the notions of graph, subgraph, directed graph, path, circuit, cutset, connectedness, connected components, (spanning) tree, cotree etc. A forest of the graph is obtained by taking a tree for each component and a coforest is its complement. For a graph \mathcal{G} , which for us will invariably be directed, $V(\mathcal{G})$, $E(\mathcal{G})$ denote the vertex set and edge set respectively. The number of edges in a forest of \mathcal{G} is its rank, denoted by $r(\mathcal{G})$, and that in a coforest is its nullity, denoted by $\nu(\mathcal{G})$. A separator of a graph is a subset of edges with the property that there is no circuit of the graph containing an edge inside and an edge outside the subset. Minimal separators are called elementary separators. Any connected graph can be uniquely decomposed into subgraphs (called 2-connected components) on elementary separators, which will be linked to each other at ‘hinges’ or ‘cut vertices’. A hinge is the only vertex where two connected subgraphs, whose edge sets are complements of each other, meet. Electrically speaking, i.e., in terms of KCL and KVL, it is as though these subgraphs are disconnected.

Vectors are treated as functions from a set to a field, invariably that of real numbers. Examples are voltage and current vectors defined on the set of edges of a graph and potential vector defined on the set of nodes of a graph. If \mathbf{f} is a vector on S and $T \subseteq S$, the *restriction* of \mathbf{f} to T denoted \mathbf{f}/\mathbf{T} , is the vector on T , whose values agree with the values of \mathbf{f} on T . Given a collection \mathcal{K} of vectors on S which includes the zero vector, and a subset $T \subseteq S$, the collection $\mathcal{K} \cdot T$ is made up of all restrictions of vectors in \mathcal{K} and the collection $\mathcal{K} \times T$ is the subset of $\mathcal{K} \cdot T$ where each vector is the restriction of a vector that is zero on $S - T$. If \mathbf{f}, \mathbf{g} are both vectors on a set S , the *dot product* $\langle \mathbf{f}, \mathbf{g} \rangle \equiv \sum_{e \in S} f(e).g(e)$. The vectors \mathbf{f}, \mathbf{g} are said to be *orthogonal* if their dot product is zero. The collection of all vectors on S orthogonal to vectors in \mathcal{K} , is denoted \mathcal{K}^\perp . When \mathcal{K} is a

vector space and $T \subseteq S$, it can be shown directly that $(\mathcal{K} \cdot T)^\perp = \mathcal{K}^\perp \times T$. Using the fact that when S is finite and \mathcal{K} is a vector space, we have $\mathcal{K}^{\perp\perp} = \mathcal{K}$, it would then follow that $(\mathcal{K} \times T)^\perp = \mathcal{K}^\perp \cdot T$. (See [Narayanan+Patkar09].)

The incidence matrix (usually denoted by \mathbf{A}), of a directed graph, has one row per node and one column per edge, with the (i, j) entry being $+1(-1)$ if edge j is directed away (towards) node i and zero otherwise. The matrix obtained from the incidence matrix, by omitting one row per component of the graph, is called the reduced incidence matrix and has the same row space as the incidence matrix. For a graph \mathcal{G} , a *current vector* \mathbf{i} is a vector on $E(\mathcal{G})$ that is orthogonal to the rows of the incidence matrix of \mathcal{G} , *equivalently*, that satisfies Kirchoff's current equations (KCE): $\mathbf{A}\mathbf{x} = \mathbf{0}$.

A *voltage vector* \mathbf{v} of \mathcal{G} is a vector on $E(\mathcal{G})$ that is linearly dependent on the rows of the incidence matrix of \mathcal{G} i.e., $\mathbf{v}^\mathbf{T} = \lambda^\mathbf{T}\mathbf{A}$ for some vector λ .

The vector λ assigns a value to each node of \mathcal{G} and is called a *potential vector*. We say \mathbf{v} is *derived* from the node potential vector λ .

Voltage vectors and current vectors form vector spaces denoted by $\mathcal{V}_v(\mathcal{G}), \mathcal{V}_i(\mathcal{G})$, and called voltage space of \mathcal{G} and current space of \mathcal{G} respectively. An immediate consequence of the definition of the voltage and current spaces is the celebrated *Tellegen's Theorem* which states that $(\mathcal{V}_v(\mathcal{G}))^\perp = \mathcal{V}_i(\mathcal{G})$.

It is clear from the definition of voltage vector, that we can assign, for the edges of any tree of a connected graph, arbitrary voltage values and this would uniquely determine cotree voltages. For, if tree voltages are given, we can assign a reference potential to some node and by traversing the tree, assign to all other nodes an appropriate unique potential. Thus the tree voltages uniquely fix difference of potential between any pair of nodes and thence fix all cotree voltages. In particular, we can assign to one branch e of the tree t , value 1 and to all others in the tree, value 0. Let $\mathbf{v}^e \equiv (\mathbf{v}_t^e | \mathbf{v}_{\bar{t}}^e)$ be the corresponding voltage vector. Note that when the branch e is removed from the tree, the latter splits into two connected pieces with vertex sets V_{e+}, V_{e-} , say, with V_{e+} being the vertex set where the tail of e is incident. The set of edges in the original graph between these two vertex sets is called the fundamental cutset of e with respect to cotree \bar{t} and denoted $L^*(e, \bar{t})$. The vector \mathbf{v}^e has nonzero values only on the edges in $L^*(e, \bar{t})$ with the value being $+1(-1)$ if the tail is in $V_{e+}(V_{e-})$. The matrix, which has as rows the voltage vectors constructed in the above manner for each edge in a tree t , is called the fundamental cutset matrix $\mathbf{Q}_{\bar{t}}$, with respect to the cotree \bar{t} . Given any voltage vector $\hat{\mathbf{v}} \equiv (\hat{\mathbf{v}}_t | \hat{\mathbf{v}}_{\bar{t}})$, we observe that the vector $\hat{\mathbf{v}} - \sum_{e \in t} (\hat{v}(e)\mathbf{v}^e)$ is a voltage vector but has zero value on all tree branches. The above traversal through tree branches shows that it must be a zero vector. It follows that the rows of the fundamental cutset matrix form a basis for $\mathcal{V}_v(\mathcal{G})$.

Now if $\mathbf{i} \equiv (\mathbf{i}_t | \mathbf{i}_{\bar{t}})$ is any current vector, it is clear, since \mathbf{v}^e, \mathbf{i} are orthogonal, that $\langle \mathbf{v}_t^e, \mathbf{i}_{\bar{t}} \rangle = -\langle \mathbf{v}_t^e, \mathbf{i}_t \rangle = -i(e)$. Thus cotree current values uniquely determine tree current values and, if all cotree current values are zero, so will all tree currents be. In particular, we can assign to one branch c of the cotree \bar{t} , value 1 and to all others in the cotree, value 0. Let $\mathbf{i}^c \equiv (\mathbf{i}_t^c | \mathbf{i}_{\bar{t}}^c)$ be the corresponding current vector. Note that when the branch c is added to the tree t , exactly

one circuit is formed, called the fundamental circuit of c with respect to the tree t denoted by $L(c, t)$. The vector \mathbf{i}^c has non zero values only on the edges in $L(c, t)$, with the value being $+1(-1)$ if the orientation of the edge in the circuit agrees with (opposes) that of c . The matrix, which has as rows the current vectors constructed in the above manner for each edge in a cotree, is called the fundamental circuit matrix \mathbf{B}_t with respect to the tree t . By using the argument that we used for voltage vectors, it follows that the rows of the fundamental circuit matrix form a basis for $\mathcal{V}_i(\mathcal{G})$.

Let \mathcal{G} be a graph, with $E(\mathcal{G}) \equiv E$ and let $T \subseteq E$. We now define some useful derived graphs natural to circuit theory.

The graph $\mathcal{G}_{open}(E - T)$ has the same vertex set as \mathcal{G} but with the edges of $(E - T)$ removed. The graph $\mathcal{G} \cdot T$ is obtained from $\mathcal{G}_{open}(E - T)$ by removing isolated (with no edges incident) vertices.

The vertex set of $\mathcal{G}_{short}(E - T)$ is the set $\{V_1, V_2, \dots, V_n\}$, where V_i is the vertex set of the i^{th} component of $\mathcal{G}_{open}T$, an edge $e \in T$ being directed from V_i to V_j in $\mathcal{G}_{short}(E - T)$, if it is directed from $a \in V_i$ to $b \in V_j$ in \mathcal{G} . The graph $\mathcal{G} \times T$ is obtained from $\mathcal{G}_{short}(E - T)$ by removing isolated vertices. (The two are the same if \mathcal{G} is connected.)

From the above construction of $\mathcal{G} \cdot (E - T)$, $\mathcal{G} \times T$, we have

Theorem 1.2.1 1. *Maximal intersection of forest (coforest) of \mathcal{G} with T is a forest of $\mathcal{G} \cdot T$ (coforest of $\mathcal{G} \times T$).*

2. *Union of forests of $\mathcal{G} \cdot (E - T)$ and $\mathcal{G} \times T$ is a forest of \mathcal{G} .*

3. $r(\mathcal{G}) = r(\mathcal{G} \cdot (E - T)) + r(\mathcal{G} \times T)$.

4. $\nu(\mathcal{G}) = \nu(\mathcal{G} \cdot (E - T)) + \nu(\mathcal{G} \times T)$.

Graphs obtained from \mathcal{G} , by opening some edges and shorting others, are called *minors of \mathcal{G}* . We note that, when some edges are shorted and others open circuited, the order in which these operations are performed does not affect the resulting graph. So we have,

Theorem 1.2.2 *Let $T_1 \subseteq T_2 \subseteq E$. Then*

1. $\mathcal{G} \cdot T_2 \cdot T_1 = \mathcal{G} \cdot T_1$

2. $\mathcal{G} \times T_2 \times T_1 = \mathcal{G} \times T_1$

3. $\mathcal{G} \times T_2 \cdot T_1 = \mathcal{G} \cdot (E - (T_2 - T_1)) \times T_1$.

We have the following important results on the voltage and current spaces associated with minors of graphs.

Theorem 1.2.3 1. $\mathcal{V}_v(\mathcal{G} \cdot T) = (\mathcal{V}_v(\mathcal{G})) \cdot T$

2. $\mathcal{V}_v(\mathcal{G} \times T) = (\mathcal{V}_v(\mathcal{G})) \times T$

Proof : i. Let $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G} \cdot T)$. Now $\mathcal{V}_v(\mathcal{G} \cdot T) = \mathcal{V}_v(\mathcal{G}open(E - T))$.

Thus, $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G}open(E - T))$. Let \mathbf{v}_T be derived from the potential vector λ of $\mathcal{G}open(E - T)$. Now for any edge $e \in T$, $\mathbf{v}_T(e) = \lambda(a) - \lambda(b)$, where a, b are the positive and negative end points of e . However, λ is also a potential vector of \mathcal{G} . Let the voltage vector \mathbf{v} of \mathcal{G} be derived from λ . For the edge $e \in T$, we have, as before, $\mathbf{v}(e) = \lambda(a) - \lambda(b)$. Thus, $\mathbf{v}_T = \mathbf{v}/T$ and therefore, $\mathbf{v}_T \in (\mathcal{V}_v(\mathcal{G})) \cdot T$. Hence $\mathcal{V}_v(\mathcal{G} \cdot T) \subseteq (\mathcal{V}_v(\mathcal{G})) \cdot T$. The reverse containment is proved similarly.

ii. Let $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G} \times T)$. We have $\mathcal{V}_v(\mathcal{G} \times T) = \mathcal{V}_v(\mathcal{G}short(E - T))$.

The vertex set of $\mathcal{G}short(E - T)$ is, say, the set $\{V_1, V_2, \dots, V_n\}$, where V_i is the vertex set of the i^{th} component of $\mathcal{G}openT$, an edge $e \in T$ being directed from V_i to V_j in $\mathcal{G}short(E - T)$, if it is directed from $a \in V_i$ to $b \in V_j$ in \mathcal{G} .

Now, $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G}short(E - T))$. Let \mathbf{v}_T be derived from the potential vector $\hat{\lambda}$ in $\mathcal{G}short(E - T)$. The vector $\hat{\lambda}$ assigns to each of the V_i , the value $\hat{\lambda}(V_i)$. Define a potential vector λ on the nodes of \mathcal{G} as follows: $\lambda(n) \equiv \hat{\lambda}(V_i), n \in V_i$. Since $\{V_1, \dots, V_k\}$ is a partition of $V(\mathcal{G})$, it is clear that λ is well defined. Let \mathbf{v} be the voltage vector derived from λ in \mathcal{G} . Whenever $e \in E - T$ we must have $\mathbf{v}(e) = 0$ since both end points must belong to the same V_i .

Next, whenever $e \in T$ we have $\mathbf{v}(e) = \lambda(a) - \lambda(b)$ where a is the positive end point of e and b , the negative endpoint. Let $a \in V_a, b \in V_b$, where $V_a, V_b \in V(\mathcal{G}short(E - T))$. Then the positive endpoint of e in $\mathcal{G}short(E - T)$ is V_a and the negative end point, V_b . By definition $\lambda(a) - \lambda(b) = \hat{\lambda}(V_a) - \hat{\lambda}(V_b)$. Thus $\mathbf{v}/T = \mathbf{v}_T$. Hence, $\mathbf{v}_T \in (\mathcal{V}_v(\mathcal{G})) \times T$. Thus, $\mathcal{V}_v(\mathcal{G} \times T) \subseteq (\mathcal{V}_v(\mathcal{G})) \times T$.

The reverse containment is proved similarly, but using the idea, that if a voltage vector is zero on all elements of $E - T$, then a potential vector from which it is derived, must have the same value on all vertices of each V_i , since these are vertex sets of components of $\mathcal{G}openT$. ■

Using duality we can now prove

Theorem 1.2.4 *Let \mathcal{G} be a directed graph on edge set E . Let $T \subseteq E$. Then,*

1. $\mathcal{V}_i(\mathcal{G} \cdot T) = (\mathcal{V}_i(\mathcal{G})) \times T$.

2. $\mathcal{V}_i(\mathcal{G} \times T) = (\mathcal{V}_i(\mathcal{G})) \cdot T$.

Proof :

i. $\mathcal{V}_i(\mathcal{G} \cdot T) = (\mathcal{V}_v(\mathcal{G} \cdot T))^\perp$ by Tellegen's Theorem. By Theorem 1.2.3, $\mathcal{V}_v(\mathcal{G} \cdot T) = (\mathcal{V}_v(\mathcal{G})) \cdot T$. Hence, $\mathcal{V}_i(\mathcal{G} \cdot T) = ((\mathcal{V}_v(\mathcal{G})) \cdot T)^\perp = (\mathcal{V}_v(\mathcal{G}))^\perp \times T = \mathcal{V}_i(\mathcal{G}) \times T$.

ii. The proof is similar. ■

It is useful to note some elementary facts about coloops and selfloops. A coloop, by definition, does not belong to any circuit and therefore must belong to every forest. Dually, a selfloop, by definition, does not belong to any cutset and therefore must belong to every coforest. The fundamental circuit and cutset matrices that result when coloop (selfloop) edges are shorted or open circuited are the same. So we have,

Theorem 1.2.5 *Let $T \subseteq E$ be a set of edges composed entirely of self loops and coloops. Then $\mathcal{V}_v(\mathcal{G} \cdot (E - T)) = (\mathcal{V}_v(\mathcal{G} \times (E - T)))$ and $\mathcal{V}_i(\mathcal{G} \cdot (E - T)) = (\mathcal{V}_i(\mathcal{G} \times (E - T)))$.*

We are now in a position to state and prove a result which will enable us to give a topological version of hybrid analysis.

Theorem 1.2.6 *Let (A, B) be a partition of $E(\mathcal{G})$. Let K be a forest and L_A , a coforest of $\mathcal{G} \cdot A$ and t_B be a forest and L , a coforest of $\mathcal{G} \times B$. Let \mathcal{G}_{AL} be the graph $\mathcal{G} \times (A \cup L)$ and let \mathcal{G}_{BK} be the graph $\mathcal{G} \cdot (B \cup K)$. Then*

1. $i_K|i_{L_A}|i_{t_B}|i_L$ is a current vector of \mathcal{G} , iff there exist current vectors $i_K|i_{L_A}|i_L$ of \mathcal{G}_{AL} and $i'_K|i_{t_B}|i_L$ of \mathcal{G}_{BK} .
2. $v_K|v_{L_A}|v_{t_B}|v_L$ is a voltage vector of \mathcal{G} , iff there exist voltage vectors $v_K|v_{L_A}|v'_L$ of \mathcal{G}_{AL} and $v_K|v_{t_B}|v_L$ of \mathcal{G}_{BK} .

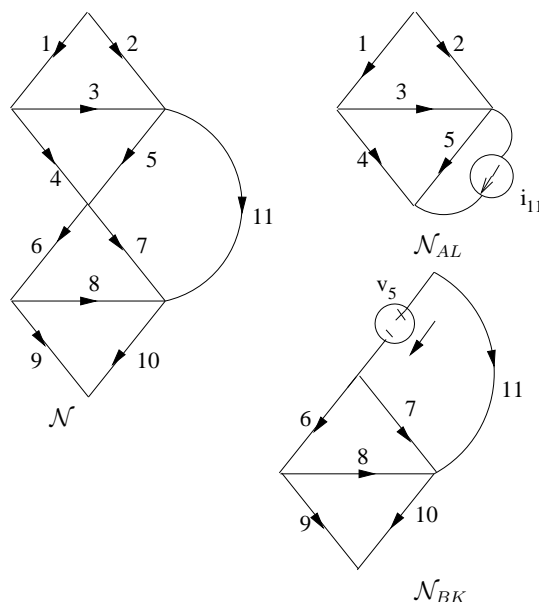
For ease of readability we relegate the proof of this result to Section 1.4.

1.3 Topological Hybrid Analysis Procedure

In this section we present a topological version of the hybrid analysis procedure. Essentially, the network is decomposed into two subnetworks, whose simultaneous analysis, matching certain ‘boundary conditions’, is equivalent to the analysis of the original network. The validity of this procedure rests on Theorem 1.2.6 and requires only that the device characteristic of the subsets of edges of the derived networks appear decoupled in the original network. In the special case of resistive networks we could write nodal equations for one of the networks and loop equations for the other and match boundary conditions. This yields hybrid analysis equations with greater freedom in the choice of unknowns which could translate to good properties such as sparsity for the coefficient matrix.

Let \mathcal{N} be a network on the graph \mathcal{G} . (The reader may use Figure 1.1 to illustrate the procedure.) Let (A, B) be a partition of $E(\mathcal{G})$ such that in the device characteristic of \mathcal{N} the devices in A, B are independent of, i.e., decoupled from, each other. (In Figure 1.1, $A \equiv \{1, 2, 3, 4, 5\}$ and B , the complement.) Let K be a forest and L_A , a coforest of $\mathcal{G} \cdot A$ and t_B be a forest and L , a coforest of $\mathcal{G} \times B$. ($K \equiv \{1, 2, 5\}$, $L_A \equiv \{3, 4\}$, $t_B \equiv \{6, 8, 9\}$, $L \equiv \{7, 10, 11\}$). Let \mathcal{G}_{AL} be the graph $\mathcal{G} \times (A \cup L)$ and let \mathcal{G}_{BK} be the graph $\mathcal{G} \cdot (B \cup K)$. (In the present example, $\mathcal{G} \times (A \cup L)$ would have 7, 10 as self loops. In the figure, the graph shown with caption \mathcal{N}_{AL} is this graph with the self loops omitted. The graph $\mathcal{G} \cdot (B \cup K)$ would have 1, 2 as coloops. In the figure, the graph shown with caption \mathcal{N}_{BK} is this graph with the coloops omitted.)

We now build two networks \mathcal{N}_{AL} and \mathcal{N}_{BK} as follows:- \mathcal{N}_{AL} has graph \mathcal{G}_{AL} with edge set $A \cup L$ built from \mathcal{G} by short circuiting (fusing the end points of) edges in t_B and removing them. The devices in A have the same characteristics as in \mathcal{N} and L has no device characteristic constraints. \mathcal{N}_{BK} has graph \mathcal{G}_{BK} with edge set $B \cup K$ built from \mathcal{G} by open circuiting edges (removing the edges

Figure 1.1: To illustrate the $\mathcal{N}_{AL} - \mathcal{N}_{BK}$ Method

but leaving the end points in place) in L_A . The devices in B have the same characteristics as in \mathcal{N} and K has no device characteristic constraints. (Note that the L, K edges are present in both networks.) Theorem 1.2.6 implies that [Narayanan79] solving \mathcal{N} is equivalent to solving \mathcal{N}_{AL} and \mathcal{N}_{BK} simultaneously keeping i_L, v_K the same in both networks. It may be noted that, if branches in L contain selfloops in the graph \mathcal{G}_{AL} , they may be deleted or contracted (endpoints fused and edge removed) from that graph and the current matching between \mathcal{N}_{AL} and \mathcal{N}_{BK} be confined to the remaining L branches. Similarly, if branches in K contain coloops in the graph \mathcal{G}_{BK} , they may be deleted or contracted from that graph and the voltage matching between \mathcal{N}_{AL} and \mathcal{N}_{BK} be confined to the remaining K branches.

Consider now the case where the device characteristic for the edges has the form

$$(i - \mathcal{J}) = G(v - \mathcal{E}), \quad (1.1)$$

where G is a block diagonal matrix with principal diagonal submatrices G_A, G_B , where we take G_B to be invertible. In this case, hybrid analysis equations can be written as follows:-

1. Write nodal analysis equations for \mathcal{N}_{AL} treating branches in L as current sources of value i_L .
2. Write loop analysis equations for \mathcal{N}_{BK} treating branches in K as voltage sources of value v_K .

3. Force the constraints that i_L is the same in both networks and v_K is the same in both networks.

We now proceed formally. The reader is invited to refer to Figure 1.1.

Let $[A_{rA}A_{rL}] = [A_{rK}A_{r(A-K)}A_{rL}]$ be a reduced incidence matrix of \mathcal{G}_{AL} . Let the device characteristic of the edges in A be expressible as

$$(i_A - \mathcal{J}_A) = G_A(v_A - \mathcal{E}_A). \quad (1.2)$$

We then have, (since $i \in \mathcal{V}_i(\mathcal{G})$),

$$A_{rA}i_A + A_{rL}i_L = 0 \quad (1.3)$$

$$\text{i.e., } A_{rA}(i_A - \mathcal{J}_A) + A_{rL}i_L = -A_{rA}\mathcal{J}_A \quad (1.4)$$

$$\text{i.e., } A_{rA}G_A(v_A - \mathcal{E}_A) + A_{rL}i_L = -A_{rA}\mathcal{J}_A \quad (1.5)$$

$$\text{i.e., } A_{rA}G_A v_A + A_{rL}i_L = -A_{rA}\mathcal{J}_A + A_{rA}G_A \mathcal{E}_A \quad (1.6)$$

Now,

$$\begin{bmatrix} v_A \\ v_L \end{bmatrix} = \begin{bmatrix} A_{rA}^T \\ A_{rL}^T \end{bmatrix} v_{nA}, \quad (1.7)$$

for some v_{nA} (since $\begin{bmatrix} v_A \\ v_L \end{bmatrix} \in \mathcal{V}_v(\mathcal{G}_{AL})$). We thus have,

$$(A_{rA}G_A A_{rA}^T)v_{nA} + A_{rL}i_L = -A_{rA}\mathcal{J}_A + A_{rA}G_A \mathcal{E}_A. \quad (1.8)$$

These are the nodal analysis equations of \mathcal{N}_{AL} . Note that we could have used any matrix which has as its rows a basis of $\mathcal{V}_v(\mathcal{G}_{AL})$, in place of $(A_{rA}|A_{rL})$, resulting in a valid set of equations with ‘voltage type’ of unknowns.

Next for \mathcal{N}_{BK} , we choose the forest t for building the fundamental circuit matrix of \mathcal{G}_{BK} . Let

$$[B_K B_B] \equiv [B_K B_{t \cap B} B_L] \equiv [B_K B_{t \cap B} I_L]$$

be the fundamental circuit matrix of \mathcal{G}_{BK} with respect to forest t (where I_L denotes the identity matrix with its columns corresponding to L).

Let the device characteristic in B be expressible as $v_B - \mathcal{E}_B = R_B(i_B - \mathcal{J}_B)$.

We then have,

$$B_K v_K + B_B v_B = 0 \quad (1.9)$$

$$\text{i.e., } B_K v_K + B_B(v_B - \mathcal{E}_B) = -B_B \mathcal{E}_B \quad (1.10)$$

$$\text{i.e., } B_K v_K + B_B R_B(i_B - \mathcal{J}_B) = -B_B \mathcal{E}_B \quad (1.11)$$

$$\text{i.e., } B_K v_K + B_B R_B i_B = -B_B \mathcal{E}_B + B_B R_B \mathcal{J}_B. \quad (1.12)$$

We have,

$$\begin{bmatrix} i_B \\ i_K \end{bmatrix} = \begin{bmatrix} B_B^T \\ B_K^T \end{bmatrix} y \quad (1.13)$$

for some y , since $\begin{bmatrix} i_B \\ i_K \end{bmatrix} \in \mathcal{V}_i(\mathcal{G}_{BK})$. Hence,

$$B_K v_K + B_B R_B B_B^T y = -B_B \mathcal{E}_B + B_B R_B \mathcal{J}_B. \quad (1.14)$$

Here again we could have used any matrix which has as its rows a basis of $\mathcal{V}_i(\mathcal{G}_{BK})$, in place of $(B_B|B_K)$, resulting in a valid set of equations with ‘current type’ of unknowns.

Now we impose the condition that v_K is the same in both networks and so is i_L . But this means,

$$A_{rK}^T v_{nA} = v_K \quad (1.15)$$

and,

$$I_L^T y = i_L \quad (1.16)$$

So we get the hybrid equations,

$$A_{rA} G_A A_{rA}^T v_{nA} + A_{rL} i_L = -A_{rA} \mathcal{J}_A + A_{rA} G_A \mathcal{E}_A \quad (1.17)$$

$$B_K A_{rK}^T v_{nA} + B_B R_B B_B^T i_L = -B_B \mathcal{E}_B + B_B R_B \mathcal{J}_B \quad (1.18)$$

The matrix $\begin{bmatrix} A_{rA} G_A A_{rA}^T & A_{rL} \\ B_K A_{rK}^T & B_B R_B B_B^T \end{bmatrix}$ is positive definite if G_A , R_B are positive definite. This matrix will usually not be very sparse unless \mathcal{G}_{BK} has a suitable basis for its current space which makes $B_B R_B B_B^T$ sparse. In practice this may often be possible. The real power of these methods, however, is revealed when we try to use iterative methods. We show in Section 1.4 that $B_K A_{rK}^T = -A_{rL}^T$. This fact is computationally useful. Indeed, this means that we can use a variation of the conjugate gradient method to solve Equations 1.17 and 1.18 [Narayanan07][Siva09]. The advantage of such methods is that the matrix need not be stored explicitly - storing \mathcal{G}_{AL} , \mathcal{G}_{BK} and the device characteristic is adequate. The basic subroutine for the conjugate gradient method only requires multiplication of the coefficient matrix by a given vector which arises at each iteration. This process can be carried out entirely by breaking it down into graph theoretic operations and multiplication by the device characteristic matrix (which would often be nearly diagonal).

We note that if in \mathcal{G}_{AL} , there are selfloops in L and in \mathcal{G}_{BK} there are coloops in K , then in Equations 1.17 and 1.18, the matrix entries corresponding respectively to the current and voltage variables associated with such branches would be zero. We would therefore be justified in open circuiting or short circuiting such edges before we write equations.

These methods were originally derived for parallelization of network analysis by G.Kron and were called ‘Diakoptics’ [Kron63]. They exploit the fact that when \mathcal{G}_{AL} , \mathcal{G}_{BK} have several 2-connected components, the matrix

$$\begin{bmatrix} A_{rA} G_A A_{rA}^T & A_{rL} \\ B_K A_{rK}^T & B_B R_B B_B^T \end{bmatrix}$$

will have block diagonal structure within $A_{rA} G_A A_{rA}^T$ and $B_B R_B B_B^T$.

1.4 Proofs for Topological Hybrid Analysis

We need a preliminary lemma for the proof of Theorem 1.2.6.

Lemma 1.4.1 *Let A, B, L, K be defined as in Theorem 1.2.6. Then,*

1. $\mathcal{V}_i(\mathcal{G}_{AL} \cdot A) = \mathcal{V}_i(\mathcal{G} \cdot A)$ and dually, $\mathcal{V}_v(\mathcal{G}_{AL} \cdot A) = \mathcal{V}_v(\mathcal{G} \cdot A)$.
 $\mathcal{V}_i(\mathcal{G}_{BK} \times B) = \mathcal{V}_i(\mathcal{G} \times B)$ and dually $\mathcal{V}_v(\mathcal{G}_{BK} \times B) = \mathcal{V}_v(\mathcal{G} \times B)$.
2. $r(\mathcal{G}_{AL}) = r(\mathcal{G} \cdot A)$; $\nu(\mathcal{G}_{BK}) = \nu(\mathcal{G} \times B)$.

Proof of the Lemma: It can be seen that $K \cup t_B$ is a forest and $L_A \cup L$, a coforest of \mathcal{G} . It follows that $K \cup t_B$ is a forest of $\mathcal{G} \cdot (A \cup t_B)$ and $L_A \cup L$ is a coforest of $\mathcal{G} \times (B \cup L_A)$. Since $K \cup t_B$ is a forest of $\mathcal{G} \cdot (A \cup t_B)$ and K is a forest of $\mathcal{G} \cdot (A \cup t_B) \cdot A (= \mathcal{G} \cdot A)$, it follows that (using Theorem 1.2.1) $r(\mathcal{G} \cdot (A \cup t_B) \times t_B) = r(\mathcal{G} \cdot (A \cup t_B)) - r(\mathcal{G} \cdot A) = |t_B|$. So $\nu(\mathcal{G} \cdot (A \cup t_B) \times t_B) = 0$. Thus the edges of t_B are not part of any circuit in $\mathcal{G} \cdot (A \cup t_B)$. Similarly, since $L_A \cup L$ is a coforest of $\mathcal{G} \times (B \cup L_A)$ and L is a coforest of $\mathcal{G} \times (B \cup L_A) \times B (= \mathcal{G} \times B)$, it follows that (using Theorem 1.2.1) $\nu(\mathcal{G} \times (B \cup L_A) \cdot L_A) = \nu(\mathcal{G} \times (B \cup L_A)) - \nu(\mathcal{G} \times B) = |L_A|$. So $r(\mathcal{G} \times (B \cup L_A) \cdot L_A) = 0$. Thus the edges of L_A are not part of any cutset in $\mathcal{G} \times (B \cup L_A)$.

We observe that $\mathcal{G}_{AL} \cdot A \equiv \mathcal{G} \times (A \cup L) \cdot A = \mathcal{G} \cdot (A \cup t_B) \times A$. But in the graph $\mathcal{G} \cdot (A \cup t_B)$, t_B is a set of coloops. Hence shorting or opening these edges will not affect the KCL or KVL constraints of the resulting graph. Thus, $\mathcal{V}_i(\mathcal{G}_{AL} \cdot A) = \mathcal{V}_i(\mathcal{G} \times (A \cup L) \cdot A) = \mathcal{V}_i(\mathcal{G} \cdot (A \cup t_B) \times A) = \mathcal{V}_i(\mathcal{G} \cdot (A \cup t_B) \cdot A) = \mathcal{V}_i(\mathcal{G} \cdot A)$ and dually, $\mathcal{V}_v(\mathcal{G}_{AL} \cdot A) = \mathcal{V}_v(\mathcal{G} \cdot A)$.

Further, we note that \mathcal{G}_{AL} is obtained by shorting the branches t_B in the forest $K \cup t_B$ of \mathcal{G} . Hence K , which is a forest of $\mathcal{G} \cdot A$ is also a forest of \mathcal{G}_{AL} . This proves that $r(\mathcal{G}_{AL}) = r(\mathcal{G} \cdot A)$.

Next, $\mathcal{G}_{BK} \times B \equiv \mathcal{G} \cdot (B \cup K) \times B = \mathcal{G} \times (B \cup L_A) \cdot B$. But in the graph $\mathcal{G} \times (B \cup L_A)$, L_A is a set of selfloops. Hence shorting or opening these edges will not affect the KCL or KVL constraints of the resulting graph. Thus, $\mathcal{V}_i(\mathcal{G}_{BK} \times B) = \mathcal{V}_i(\mathcal{G} \cdot (B \cup K) \times B) = \mathcal{V}_i(\mathcal{G} \times (B \cup L_A) \cdot B) = \mathcal{V}_i(\mathcal{G} \times (B \cup L_A) \times B) = \mathcal{V}_i(\mathcal{G} \times B)$ and dually, $\mathcal{V}_v(\mathcal{G}_{BK} \times B) = \mathcal{V}_v(\mathcal{G} \times B)$.

Further, we note that \mathcal{G}_{BK} is obtained by opening the branches L_A in the coforest $L \cup L_A$ of \mathcal{G} . Hence L , which is a coforest of $\mathcal{G} \times B$ is also a coforest of \mathcal{G}_{BK} . This proves that $\nu(\mathcal{G}_{BK}) = \nu(\mathcal{G} \times B)$. ■

Proof of Theorem 1.2.6:

1. $i_K | i_{L_A} | i_{t_B} | i_L$ is a current vector of \mathcal{G} , iff there exist current vectors $i_K | i_{L_A} | i_L$ of \mathcal{G}_{AL} and $i'_K | i_{t_B} | i_L$ of \mathcal{G}_{BK} .
2. $v_K | v_{L_A} | v_{t_B} | v_L$ is a voltage vector of \mathcal{G} , iff there exist voltage vectors $v_K | v_{L_A} | v'_L$ of \mathcal{G}_{AL} and $v_K | v_{t_B} | v_L$ of \mathcal{G}_{BK} .

Let $i_K|i_{L_A}|i_{t_B}|i_L$ be a current vector of \mathcal{G} . From Theorem 1.2.3, it follows that $i_K|i_{L_A}|i_L$ is a current vector of \mathcal{G}_{AL} and $i_{t_B}|i_L$ is a current vector of $\mathcal{G} \times B$. But $\mathcal{V}_i(\mathcal{G}_{BK} \times B) = (\mathcal{V}_i(\mathcal{G}_{BK})) \cdot B = \mathcal{V}_i(\mathcal{G} \times B)$. So there exists a current vector $i'_K|i_{t_B}|i_L$ of \mathcal{G}_{BK} . Next, suppose there exist current vectors $i_K|i_{L_A}|i_L$ of \mathcal{G}_{AL} and $i'_K|i_{t_B}|i_L$ of \mathcal{G}_{BK} . Since $i'_K|i_{t_B}|i_L$ is a current vector of \mathcal{G}_{BK} , it follows that $i_{t_B}|i_L$ is a current vector of $\mathcal{G}_{BK} \times B$ and therefore of $\mathcal{G} \times B$. Now $L \cup L_A$ is a coforest of \mathcal{G} . Hence for any arbitrary vector $i_{L_A}|i_L$, there exists a unique current vector $i''_K|i_{L_A}|i_{t_B}|i_L$ of \mathcal{G} . But then $i''_K|i_{L_A}|i_L$ is a current vector of \mathcal{G}_{AL} and $i''_{t_B}|i_L$ is a current vector of $\mathcal{G} \times B$, i.e., of $\mathcal{G}_{BK} \times B$. Now L is a coforest of $\mathcal{G} \times B$ and $L \cup L_A$ is a coforest of \mathcal{G}_{AL} . So for any arbitrary vector $i_{L_A}|i_L$, there is a unique current vector $i^3_K|i_{L_A}|i_L$ of \mathcal{G}_{AL} and $i^3_{t_B}|i_L$ of $\mathcal{G} \times B$. But we already have seen that there exist current vectors $i_K|i_{L_A}|i_L$ of \mathcal{G}_{AL} and $i_{t_B}|i_L$ of $\mathcal{G} \times B$. It follows that $i_K = i''_K$ and $i_{t_B} = i''_{t_B}$ and therefore $i_K|i_{L_A}|i_{t_B}|i_L$ is a current vector of \mathcal{G} . The voltage part of the theorem is dual to the above, i.e., in the proof above we interchange voltage and current, \mathcal{G}_{AL} and \mathcal{G}_{BK} $\prime\prime$ and $\prime\prime\prime$, K and L , t_B and L_A . \blacksquare

Proof of claim $B_K A_{rK}^T = -A_{rL}^T$.

Observe that the rows of $(B_K|B_{t_B}|I_L)$ span the space $\mathcal{V}_i(\mathcal{G}_{BK})$. Hence the rows of $(B_K|I_L)$ span the space $(\mathcal{V}_i(\mathcal{G}_{BK})) \cdot (K \cup L) = \mathcal{V}_i(\mathcal{G}_{BK} \times (K \cup L))$, by Theorem 1.2.3. But $\mathcal{G}_{BK} = \mathcal{G} \cdot (B \cup K)$. Hence the rows of $(B_K|I_L)$ span the space $\mathcal{V}_i(\mathcal{G} \cdot (B \cup K) \times (K \cup L)) = \mathcal{V}_i(\mathcal{G} \times (A \cup L) \cdot (K \cup L))$. Next, we note that the rows of $(A_{rK}|A_{rL})$ span the space $(\mathcal{V}_v(\mathcal{G}_{AL})) \cdot (K \cup L) = \mathcal{V}_v(\mathcal{G}_{AL} \cdot (K \cup L)) = \mathcal{V}_v(\mathcal{G} \times (A \cup L) \cdot (K \cup L))$. The claim follows since rows of $(B_K|I_L)$ and $(A_{rK}|A_{rL})$ are orthogonal.

1.5 Principal Partition Problem

In this section we relate the hybrid analysis problem to the principal partition of graphs. The latter has played a fundamental role in the development of combinatorial optimization in the context of matroids [Narayanan+Patkar09] and submodular functions [Narayanan97]. All the results in the present section are best understood in a unified way as applications of the matroid union theorem and the principal partition for polymatroid rank functions discussed in [Narayanan+Patkar09]. However, for readability we give a self contained, graph based, treatment here while pointing out connections to results of that chapter.

When the network \mathcal{N} is linear, if we write nodal equations for \mathcal{N}_{AL} and loop equations for \mathcal{N}_{BK} (defined as in Subsection 1.3), the total number of equations would be $r(\mathcal{G} \cdot A) + \nu(\mathcal{G} \times B)$. So one could ask for the partition A, B for which the above expression reaches a minimum value. More generally, given a partition (A, B) of $E(\mathcal{G})$, by Theorem 1.2.3 we know that there exists a current vector $\mathbf{i}_A|\mathbf{i}_B$ of \mathcal{G} iff \mathbf{i}_B is a current vector of $\mathcal{G} \times B$ and that there exists a voltage vector $\mathbf{v}_A|\mathbf{v}_B$ of \mathcal{G} iff \mathbf{v}_A is a voltage vector of $\mathcal{G} \cdot A$. Thus $\mathbf{v}_A, \mathbf{i}_B$ of \mathcal{G} can be uniquely determined using only KVL and KCL from the forest voltage vector \mathbf{v}_{t_A} of $\mathcal{G} \cdot A$ and the coforest current vector \mathbf{i}_{L_B} of $\mathcal{G} \times B$. Thus we have the first formulation of the hybrid rank problem:

Given a graph \mathcal{G} , partition $E(\mathcal{G})$ into A and B such that $r(\mathcal{G} \cdot A) + \nu(\mathcal{G} \times B)$ is minimized.

Historically, the following problems, related to the first formulation, were solved at about the same time. After stating them we give their solution in brief. Detailed solution may be found in the references cited therein as well as in [Narayanan97].

i. The topological degree of freedom of an electrical network

[Ohtsuki+Ishizaki+Watanabe70] This problem was posed by G.Kron.

Select a minimum sized set of branch voltages and branch currents from which, by using Kirchhoff's voltage equations and Kirchhoff's current equations, we can find either the voltage or the current associated with each branch. This minimum size is called the *topological degree of freedom of the network*, equivalently, the *hybrid rank* of the graph.

ii. The Shannon switching game [Edmonds65b]

\mathcal{G} is a graph with one of its edges say e_M 'marked'. There are two players - a 'cut' player and a 'short' player. The cut player, during his turn, deletes (opens) an edge leaving the end points in place. The short player, during his turn, contracts an edge, i.e., fuses its end points and removes it. Neither player is allowed to touch e_M . The cut player wins if all the paths between the end points of e_M are destroyed (equivalently, all circuits containing e_M are destroyed). The short player wins if the end points of e_M get fused (equivalently, all cutsets containing e_M are destroyed by shorting of edges). The problem is to analyze this game and characterize situations where the cut or short player, playing second, can always win and to determine the winning strategy.

iii. Maximum distance between two forests [Kishi+Kajitani69]

Define distance between two forests t_1 and t_2 as $|t_1 - t_2|$. Find two forests in a given graph which have the maximum distance between them, i.e., the size of their union is the largest possible.

iv. Forest of minimum size hybrid representation [Kishi+Kajitani69]

Let a forest t be represented by a pair of sets (A_t, B_t) where $A_t \subseteq t, t \cap B_t = \emptyset$ such that $(A_{t_1}, B_{t_1}) = (A_{t_2}, B_{t_2})$ iff $t_1 = t_2$. Note that we can represent the same forest by several pairs, for instance $(t, \emptyset), (\emptyset, E(\mathcal{G}) - t)$ both represent t . We call $|A_t \cup B_t|$ the *size of the representation* (A_t, B_t) . Find a forest which has the representation of minimum size.

v. The maximum rank of a cobase submatrix [Iri69]

For a rectangular $(m \times n)$ matrix with linearly independent rows, let us call an $m \times (n - m)$ submatrix a *cobase* submatrix iff the remaining set of columns are from an identity matrix. The *term rank* of a matrix is the maximum number of nonzero entries in the matrix which belong to distinct rows and distinct columns. Find a cobase matrix of maximum rank, and a cobase matrix of minimum term rank among all matrices row equivalent to the given matrix.

For the above five problems the solution involves essentially the same strategy: Find a set A (or a minimal set A_{min} or a maximal set A_{max}) which minimizes $2r(\mathcal{G} \cdot A) + |E(\mathcal{G}) - A|$. That these sets are unique is proved in Theorems 1.6.1 and 1.6.2. The partition of the graph into $A_{min}, A_{max} - A_{min}, E(\mathcal{G}) - A_{max}$ was called the *Principal Partition of \mathcal{G}* by Kishi and Kajitani

[Kishi+Kajitani69]. This has been discussed in detail for the more general case of polymatroid rank functions in [Narayanan+Patkar09]. The essential ideas will be repeated for graphs in Section 1.6 of this chapter. Below we have given a sketch of the solutions to the five problems. More details may be found in [Narayanan97].

Let t_A be a forest of the subgraph on A . Let $L_{\bar{A}}$ be a coforest of the graph on $\mathcal{G} \times (E(\mathcal{G}) - A)$. Select the branch voltages of t_A and the branch currents of $L_{\bar{A}}$ as the desired set of variables.

If $e_M \in A_{min}$, the short player can always win. If $e_M \in (E(\mathcal{G}) - A_{max})$ the cut player can always win. If $e_M \in A_{max} - A_{min}$, whoever plays first can always win. The winning strategies involve the construction of appropriate maximally distant forests during every turn.

Kishi and Kajitani gave an algorithm for building a pair of maximally distant forests which is essentially the well known algorithm for building a base of the union of two matroids (see [Edmonds65a] for the case where the matroids are identical - essentially the same algorithm works for the general case).

Select a forest t which has maximal intersection with A . The representation $(t \cap A, (E(\mathcal{G}) - t) \cap (E(\mathcal{G}) - A))$ has the least size among all representations of all forests. As is easily seen, the minimum size among all representations of forests of \mathcal{G} is also the same as the topological degree of freedom of \mathcal{G} and the above maximum distance.

The solution is similar for the last problem. Let S be the set of all columns and let $r(\cdot)$ be the rank function on the collection of subsets of S . Then the maximum rank of a cobase matrix = the minimum term rank of a cobase matrix. Select two maximally distant bases (bases \equiv maximally independent columns). In this case the matroid union algorithm described in [Narayanan+Patkar09], essentially a generalization of the algorithm for building maximally distant forests, has to be used. Perform row operations so that an identity matrix appears corresponding to one of these. The submatrix corresponding to the complement of this base is the desired cobase matrix which has both maximum rank as well as minimum term rank.

1.6 Building maximally distant forests

The five problems stated in the previous section were solved originally without reference to the matroid union theorem. Indeed, the problems of finding maximally distant forests, of minimal representation of forests and the topological degree of freedom were solved graph theoretically. The algorithm for building maximally distant forests of a graph is essentially the same as the above mentioned matroid union algorithm except that both the matroids whose union is sought are the polygon matroids of the same graph. We sketch this algorithm informally in this section and also relate it to the principal partition of a graph as described by Kishi and Kajitani [Kishi+Kajitani69]. We hope that this treatment helps in better visualization of the more general matroid ideas.

We begin with a simple but useful observation in the following lemma.

Lemma 1.6.1 *Let \mathcal{G} be a graph. Let t_1, t_2 be two forests and let $\overline{t_1}, \overline{t_2}$ be the corresponding coforests of \mathcal{G} . Then the following are equivalent*

1. t_1, t_2 are maximally distant;
2. $|t_1 \cup t_2|$ is the maximum possible;
3. $|t_1 \cap t_2|$ is the minimum possible;
4. $\overline{t_1}, \overline{t_2}$ are maximally distant.

Proof : Since the sizes of all forests are the same, maximizing $|t_1 - t_2|$ is the same as maximizing $|t_1 \cup t_2|$ and minimizing $|t_1 \cap t_2|$. Sizes of all coforests are the same. So these hold also for coforests. But maximizing $|t_1 \cup t_2|$ is the same as minimizing $|\overline{t_1} \cap \overline{t_2}|$. So t_1, t_2 being maximally distant is the same as $\overline{t_1}, \overline{t_2}$ being maximally distant. ■

Algorithm Maximally Distant Forests

Let t_1, t_2 be two forests of $\mathcal{G} = (V, E)$. Define a directed graph $G(t_1, t_2)$ with E as the set of vertices and with directed edges as described below.

Whenever in $\mathcal{G}, e \notin t_j, j = 1, 2$, draw, in $G(t_1, t_2)$, directed edges from the vertex e to all vertices which are edges of \mathcal{G} in the fundamental circuit $L(e, t_j)$ (formed when e of \mathcal{G} is added to tree t_j) and mark each directed edge as a t_j edge. From a given vertex e_s in $G(t_1, t_2)$, it is easy to determine the set of all vertices that can be reached through directed paths by using breadth first search. In the process, the shortest path (in terms of number of edges in the path), from e_s to every vertex in $G(t_1, t_2)$, can be determined.

The present algorithm starts from some pair of forests t_1, t_2 (which could even be the same forest) and tries to build another pair t'_1, t'_2 for which the distance $|t'_1 - t'_2|$ is greater than the distance $|t_1 - t_2|$. Clearly, if $t_1 \cup t_2$ covers all edges in E , the forests are maximally distant. Suppose $e_{out} \notin t_1 \cup t_2$. In $G(t_1, t_2)$, we find the set of all vertices (which are edges of \mathcal{G}) reachable from vertex e_{out} through directed paths. If no vertex in this set corresponds to an edge common to both t_1 and t_2 , we repeat the process with another such e_{out} from which an edge e_{com} common to both t_1 and t_2 may be reached. If no such vertex e_{out} exists we stop and output the current t_1, t_2 as maximally distant.

Let $e_{out}, (t_1), e_1, (t_2), \dots, e_i, (t_j), e_{i+1}, \dots, e_k, (t_j), e_{k+1}, \dots, e_{n-1}, (t_m), e_{com}$ be a shortest path from e_{out} to e_{com} , where $e_r, (t_p), e_{r+1}$ indicates that from the vertex e_r there is a directed edge to e_{r+1} , which is marked t_p , where t_p could be either t_1 or t_2 (the indices used for trees and edges being unrelated). We will use this path to alter t_1, t_2 . The key idea we use is the following: $L(e_i, t_j)$ does not have e_{k+1} as a member as otherwise we could have shortened the path to $e_{out}, (t_1), e_1, (t_2), \dots, e_i, (t_j), e_{k+1}, \dots, e_{n-1}, (t_m), e_{com}$. Therefore in the forest $\hat{t}_j = t_j \cup e_k - e_{k+1}$, the fundamental circuit $L(e_i, \hat{t}_j)$ will be the same as $L(e_i, t_j)$.

We can therefore alter t_1 and t_2 as

$$\begin{aligned}
t_m &\leftarrow t_m \cup e_{n-1} - e_{com} \\
&\quad \dots \\
t_j &\leftarrow t_j \cup e_i - e_{i+1} \\
&\quad \dots \\
t_1 &\leftarrow t_1 \cup e_{out} - e_1
\end{aligned}$$

The result of the above alteration would be that e_{out} would now move into $t_1 \cup t_2$ while $e_{com} \notin t_1 \cap t_2$. Therefore the size $|t_1 \cup t_2|$ and the distance $|t_1 - t_2|$ would have increased.

We repeat the above step until from none of the e_{out} we can reach any e_{com} and output the current t_1, t_2 as maximally distant.

By Lemma 1.6.1, equivalently, \bar{t}_1, \bar{t}_2 may be output as maximally distant.

More directly, the above algorithm can be converted into one which finds maximally distant coforests by working with coforest \bar{t}_i in place of forest t_i and replacing $L(e_i, t_j)$ wherever it occurs, by the fundamental cutset $L^*(e_i, \bar{t}_j)$.

We know that $e_k \in L^*(e_i, \bar{t}_j)$ iff $e_i \in L(e_k, t_j)$. This yields the relationship between $G(t_1, t_2)$ and $G(\bar{t}_1, \bar{t}_2)$ stated in the following lemma.

Lemma 1.6.2 *Reversing the direction of arrows in $G(t_1, t_2)$ and marking t_j edges by \bar{t}_j yields $G(\bar{t}_1, \bar{t}_2)$.*

We justify the above algorithm through the following theorem. Here $r(A), \nu(A)$ denote respectively $r(\mathcal{G} \cdot A), \nu(\mathcal{G} \times A)$.

Theorem 1.6.1 *Let t_1, t_2 denote forests of \mathcal{G} on edge set E and let $A \subseteq E$. Then,*

1. $|t_1 \cup t_2| \leq 2r(A) + |E - A|$ and the inequality becomes an equality only if t_1, t_2 are maximally distant and $t_i \cap A, i = 1, 2$, are disjoint forests of $\mathcal{G} \cdot A$ and $(t_1 \cup t_2) \cap (E - A) = (E - A)$;
2. the pair of forests t_1, t_2 output by Algorithm Maximally Distant Forests and the set of all edges A_{min} in E corresponding to vertices in $G(t_1, t_2)$ which can be reached from $E - (t_1 \cup t_2)$ satisfy $|t_1 \cup t_2| = 2r(A_{min}) + |E - A_{min}|$ and hence t_1, t_2 are maximally distant;
3. $\max |t_1 \cup t_2| = \min 2r(A) + |E - A|$, where t_1, t_2 are forests of $\mathcal{G}(V, E)$ and $A \subseteq E$;
4. if \bar{t}_1, \bar{t}_2 denote coforests of \mathcal{G} , then $|\bar{t}_1 \cup \bar{t}_2| \leq 2\nu(A) + |E - A|$ and the inequality becomes an equality iff \bar{t}_1, \bar{t}_2 are maximally distant, $\bar{t}_i \cap A, i = 1, 2$, are disjoint coforests of $\mathcal{G} \times A$ and $(\bar{t}_1 \cup \bar{t}_2) \cap (E - A) = (E - A)$.

Proof : *i.* We have $t_i \cap A, i = 1, 2$, as a subforest of $\mathcal{G} \cdot A$ and $(t_1 \cup t_2) \cap (E - A) \subseteq (E - A)$. So $|t_1 \cup t_2| \leq 2r(A) + |E - A|$. It is clear that if the inequality becomes an equality, the corresponding t_1, t_2 must be such that $|t_1 \cup t_2|$ is a maximum and therefore be maximally distant. The inequality becomes an equality iff the

set A on the RHS is such that $|(t_1 \cup t_2) \cap A| = 2r(A)$, i.e., $t_i \cap A, i = 1, 2$, are disjoint forests of $\mathcal{G} \cdot A$ and $(t_1 \cup t_2) \cap (E - A) = (E - A)$.

ii. If the t_1, t_2 output by the algorithm are such that $t_1 \cup t_2 = E$, the inequality will be satisfied as an equality by taking A_{min} to be the null set. If on the other hand, the algorithm outputs t_1, t_2 such that $t_1 \cup t_2 \neq E$, then A_{min} has the following properties. First, it contains $E - (t_1 \cup t_2)$, or equivalently, $(t_1 \cup t_2) \supseteq (E - A_{min})$ and A_{min} does not contain any edge in $t_1 \cap t_2$. Next in $\mathcal{G} \cdot A_{min}$, every edge in $t_i \cap A_{min}, i = 1, 2$, can be reached from an edge outside $t_1 \cup t_2$ by repeatedly taking fundamental circuits with respect to the two forests. Thus $t_i \cap A_{min}, i = 1, 2$, span both each other as well as edges in $E - (t_1 \cup t_2)$. So $t_i \cap A_{min}, i = 1, 2$, are both forests of $\mathcal{G} \cdot A_{min}$ and, further, have no intersection since A_{min} does not contain any edge in $t_1 \cap t_2$. Thus we have, $|t_1 \cup t_2| = 2r(A_{min}) + |E - A_{min}|$.

iii. is now immediate from *i.* and *ii.*

iv. The proof for the coforest case is dual, i.e., by replacing in the above argument, forests by coforests, $G(t_1, t_2)$ by $G(\bar{t}_1, \bar{t}_2)$, A_{min} by A_{min}^* , $\mathcal{G} \cdot A_{min}$ by $\mathcal{G} \times A_{min}^*$ and $r(\mathcal{G} \cdot A_{min})$ by $\nu(\mathcal{G} \times A_{min}^*)$. ■

Theorem 1.6.2 *Let t_1, t_2 (\bar{t}_1, \bar{t}_2) be maximally distant forests (coforests) of graph \mathcal{G} on edge set E . Let A_{min} (B_{min}) denote the set of all edges in E corresponding to vertices in $G(t_1, t_2)$ ($G(\bar{t}_1, \bar{t}_2)$) which can be reached from $E - (t_1 \cup t_2)$ ($E - (\bar{t}_1 \cup \bar{t}_2)$).*

1. $A_{min}(B_{min})$ is the unique minimal set that minimizes $2r(A) + |E - A|$ ($2\nu(A^*) + |E - A^*|$).
2. $\hat{A} \subseteq E$ minimizes $2r(A) + |E - A|$ iff $E - \hat{A}$ minimizes $2\nu(A^*) + |E - A^*|$.
3. $E - B_{min}$ ($E - A_{min}$) is the unique maximal set that minimizes $2r(A) + |E - A|$ ($2\nu(A^*) + |E - A^*|$).
4. An edge e belongs to A_{min} (e belongs to B_{min}) iff there exist maximally distant forests t_1, t_2 (coforests \bar{t}_1, \bar{t}_2) s.t. $e \in (E - (t_1 \cup t_2))$, ($e \in (E - (\bar{t}_1 \cup \bar{t}_2))$),

Proof : i. We know by Theorem 1.6.1, that a subset \hat{A} minimizes $2r(A) + |E - A|, A \subseteq E$ iff for every pair of maximally distant forests t_1, t_2 $|t_1 \cup t_2| = 2r(\hat{A}) + |E - \hat{A}|$ and that this happens iff $t_i \cap \hat{A}, i = 1, 2$, are disjoint forests of $\mathcal{G} \cdot \hat{A}$ and $(t_1 \cup t_2) \cap (E - \hat{A}) = (E - \hat{A})$, i.e., $\hat{A} \supseteq E - (t_1 \cup t_2)$. Thus in $G(t_1, t_2)$, we see that \hat{A} contains all vertices corresponding to $E - (t_1 \cup t_2)$ and further it is not possible to reach outside \hat{A} from within since each $t_i \cap \hat{A}, i = 1, 2$, is a forest of $\mathcal{G} \cdot \hat{A}$. Thus $A_{min} \subseteq \hat{A}$. However A_{min} itself minimizes $2r(A) + |E - A|, A \subseteq E$ and so is the unique minimal minimizing set.

The proof for the dual statement follows by arguing with coforests \bar{t}_1, \bar{t}_2 and $G(\bar{t}_1, \bar{t}_2)$.

ii. We have, $t \cap A$ is a forest of $\mathcal{G} \cdot A$ iff $\bar{t} \cap (E - A)$ is a coforest of $\mathcal{G} \times (E - A)$ (Theorem 1.2.1 of Preliminaries). Next, $t_i \cap A, i = 1, 2$, are disjoint iff $(E - A) \supseteq$

$t_1 \cap t_2$, i.e., iff $(E - A) \supseteq (E - (\bar{t}_1 \cup \bar{t}_2))$ and $\bar{t}_i \cap (E - A), i = 1, 2$, are disjoint iff $A \supseteq \bar{t}_1 \cap \bar{t}_2$, i.e., iff $A \supseteq E - (t_1 \cup t_2)$. From Theorem 1.6.2 we know that the expression $2r(A) + |E - A|$ reaches a minimum iff $t_i \cap A, i = 1, 2$, are disjoint forests of \mathcal{G} . A and $A \supseteq E - (t_1 \cup t_2)$ and the expression $2\nu(A^*) + |E - A^*|$ reaches a minimum iff $\bar{t}_i \cap (E - A), i = 1, 2$, are disjoint coforests of $\mathcal{G} \times (E - A)$ and $(E - A) \supseteq (E - (\bar{t}_1 \cup \bar{t}_2))$. The result follows.

iii. This follows immediately from *i.* and *ii.* above.

iv. Observe that every $e \in (t_1 \cup t_2) \cap A_{min}$ can be reached from some $e_{out} \in A_{min} - (t_1 \cup t_2)$. In $G(t_1, t_2)$ we therefore have a shortest path from e_{out} to e , say $e_{out}, (t_1), e_1, (t_2), \dots, e_i, (t_j), e_{i+1}, \dots, e_{n-1}, (t_m), e$. Modifying t_1, t_2 as in *Algorithm Maximally Distant Forests* would give us a new pair of maximally distant forests t'_1, t'_2 which would not contain e as a member. The result for B_{min} follows by using $G(\bar{t}_1, \bar{t}_2)$. ■

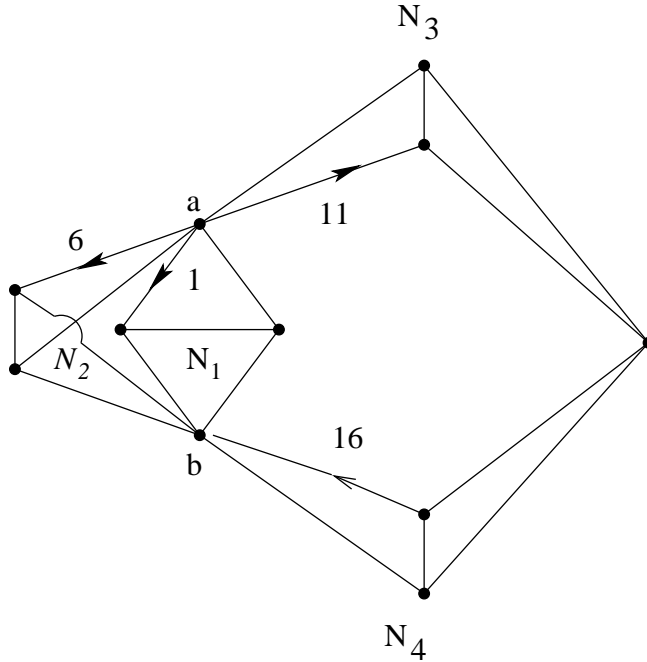
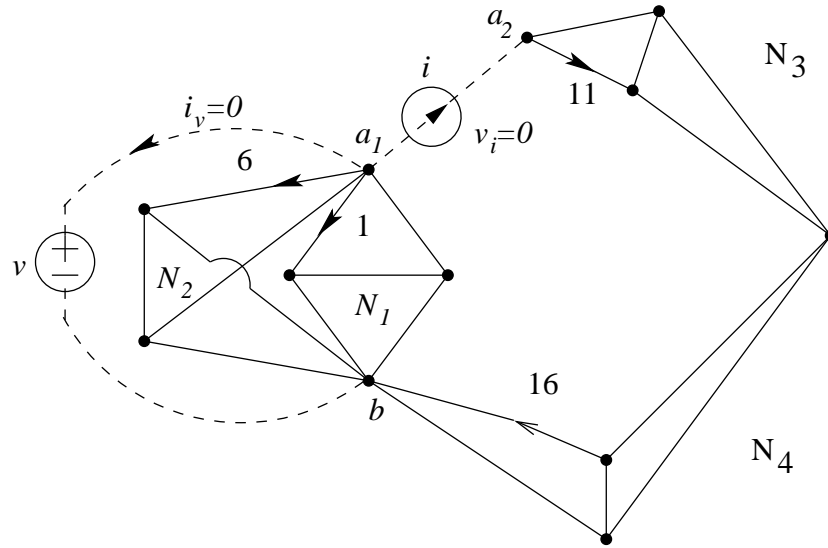


Figure 1.2: Network \mathcal{N} to Illustrate the Fusion-Fission Method

Figure 1.3: A Network equivalent to \mathcal{N} with Virtual Sources

1.7 Network Analysis through Topological Transformation

We say that we use topological transformation while analysing networks, if at intermediate stages of the analysis, we modify the topology of the network. Instances of such transformations are the construction of two derived networks during topological hybrid analysis, multiport decomposition etc. In this section we consider a fairly general class of transformations which we may call the fusion-fission method, and also sketch certain optimization problems which arise naturally during this study and which generalize the principal partition problem. Detailed description of these ideas may be found in [Narayanan90], [Narayanan91] and [Narayanan97].

Fusion-Fission method:

Consider the network in Figure 1.2. Four subnetworks have been connected together to make up the network. Assume that the devices in the subnetworks are decoupled. Clearly the networks in Figure 1.2 and Figure 1.3 are equivalent, provided the current through the additional unknown voltage source and the voltage across the additional unknown current source are set equal to zero. But the network in Figure 1.3 is equivalent to that in Figure 1.4 under the additional conditions

$$i_{v1} + i_{v2} + i = 0$$

$$v_{i3} + v_{i4} - v = 0.$$

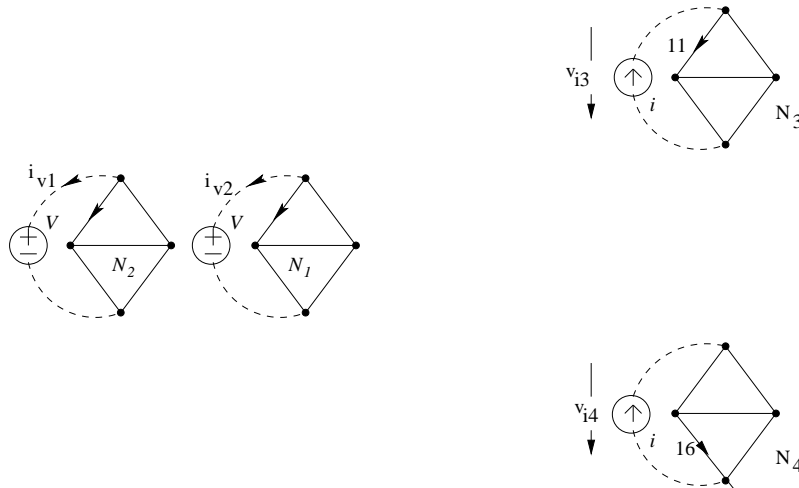


Figure 1.4: Network \mathcal{N} decomposed by the Fusion-Fission Method

(The variables $i_{v1}, i_{v2}, v_{i3}, v_{i4}$ can be expressed in terms of currents and voltages of the original graph so that the additional constraints will involve only old current and voltage variables and the new variables i, v .)

As can be seen, the subnetworks of Figure 1.4 are decoupled except for the common variables v and i and the additional conditions.

A natural optimization problem here is the following:

Given a partition of the edges of a graph into E_1, \dots, E_k , what is the minimum size set of node pair fusions and node fissions by which all circuits (equivalently cutsets) passing through more than one E_i are destroyed?

In the present example the optimal set of operations is to fuse nodes a and b and cut node a into a_1, a_2 as in Figure 1.3. Artificial voltage sources are introduced across the node pairs to be fused and artificial current sources are introduced between two halves of a split node.

The above formulation can be stated in a more convenient form as

Node fusion - fission problem

Let \mathcal{G} be a graph and let Π_s be a specified partition of $E(\mathcal{G})$ s.t. $\mathcal{G} \cdot N_i$ is connected for each $N_i \in \Pi_s$. Find a minimum length sequence of node pair fusions and node fissions which, when performed on \mathcal{G} , result in a graph \mathcal{G}_{new} in which each circuit intersects only one of the blocks of Π_s (equivalently each cutset intersects only one of the blocks of Π_s).

Assuming each $\mathcal{G} \cdot N_i$ to be connected is reasonable electrically speaking since, if they are not, we can usually be working with their connected components rather than with themselves while analysing.

We will show that this problem generalizes the hybrid rank problem (see Section 1.5). We first note that the result of a node fission followed by a node fusion can always be achieved by a node fusion followed by a node fission. Thus whenever we have a sequence of node fissions and fusions needed for converting a graph into another, the result can always be achieved by a sequence of node fusions followed by a sequence of node fissions.

We think of the original network as being made up of a number of single edge networks and the problem is to decouple them in the above manner. At the end of the fusions and fissions no edges belonging to different networks should belong to the same circuit. Therefore every edge should have become a selfloop or a coloop. Let A, B be the subsets of edges which are selfloops and coloops respectively. Every edge in A has its endpoints fused. This is exactly equivalent to fusing the endpoints of the edges of a forest of $\mathcal{G} \cdot A$. Thus with $r(\mathcal{G} \cdot A)$ node fusions all edges in A would have become selfloops. After these node fusions the resulting graph would be $\mathcal{G} \times B$. In any graph, making all edges into coloops can be achieved with minimum number of operations by cutting each coforest edge at one of its end points. Therefore making all edges in B into coloops requires $\nu(\mathcal{G} \times B)$ node fissions. Thus the minimizing of $r(\mathcal{G} \cdot A) + \nu(\mathcal{G} \times B)$ over all partitions A, B is equivalent to finding a minimum length sequence of node fusions and fissions which will make every edge in the graph into a selfloop or coloop.

1.7.1 Solution of the node fusion - fission problem

In this subsection we give a sketch of the ideas involved in the solution of the node fusion-fission problem. Let \mathcal{G} be a graph and Π_s , a partition of $E(\mathcal{G})$. The *fusion rank of \mathcal{G} relative to Π_s* is the minimum length of a sequence of node pair fusions needed to destroy every circuit that intersects more than one block of Π_s . The *fission rank of \mathcal{G} relative to Π_s* is the minimum length of a sequence of node fissions needed to destroy every circuit that intersects more than one block of Π_s . The *hybrid rank of \mathcal{G} relative to Π_s* is the minimum length of a sequence of node pair fusions and node fissions needed to destroy every circuit that intersects more than one block of Π_s .

Now consider the situation where we use both fusions and fissions, with all the fusions occurring first. Any sequence of node pair fusions would ultimately fuse certain groups of nodes into single nodes. Hence, as far as the effect of these node pair fusions on the graph is concerned, we may identify them with a partition of $V(\mathcal{G})$ (with singleton blocks being permitted) each block of which would be reduced to a single node by the fusions. The number of node pair fusions required to convert a set of nodes V to a single node is $(|V| - 1)$. Hence, if Π is a partition of $V(\mathcal{G})$, the number of node pair fusions required to go from \mathcal{G} to the graph obtained from \mathcal{G} by fusing blocks of Π into single vertices, which we shall denote by $\mathcal{G}_{fus.\Pi}$, is $|V(\mathcal{G})| - |\Pi|$. This number we would henceforth call, the *fusion number of Π* . The fission rank of $\mathcal{G}_{fus.\Pi}$ relative to a partition Π_s of $E(\mathcal{G})$ would be called the *fission number of Π relative to Π_s* .

The sum of the fusion number and the fission number of Π relative Π_s would be called the *fusion - fission number of Π relative to Π_s* . Our task is to find a partition of $V(\mathcal{G})$ which minimizes this number.

We now define a bipartite graph which relates Π_s to $V(\mathcal{G})$. Let $B_{\mathcal{G}}$ be the bipartite graph associated with \mathcal{G} , with ‘left vertices’ $V_L \equiv V(\mathcal{G})$ and ‘right vertices’ $V_R \equiv E(\mathcal{G})$, with $e \in V_R$ adjacent to $v \in V$ iff edge e is incident on v in \mathcal{G} . Let $B(\Pi_s)$ be the bipartite graph obtained from $B_{\mathcal{G}}$ by fusing the right vertices in the blocks of Π_s and replacing parallel edges by single edges. Let $X \subseteq V(\mathcal{G})$. Let $|\Gamma_L| (X)$ denote the size of the set of right vertices adjacent to vertices in X , $(|\Gamma_L| - \lambda)(X)$ denote $|\Gamma_L| (X) - \lambda$. Let Π be a partition of $V(\mathcal{G})$. We define $(|\Gamma_L| - \lambda)(\Pi)$ to be the sum of the values of $(|\Gamma_L| - \lambda)$ on the blocks of Π . We then have the following result whose proof we omit in the interest of brevity.

Theorem 1.7.1 *Let \mathcal{G} be a connected graph. Let Π_s be a partition of $E(\mathcal{G})$ s.t. $\mathcal{G} \cdot N_i$ is connected for each $N_i \in \Pi_s$. Let Π be a partition of $V(\mathcal{G})$. Then*

1. *the fusion - fission number of Π relative to Π_s equals*

$$\overline{(|\Gamma_L| - 2)(\Pi)} + |V(\mathcal{G})| - |\Pi_s| + 1$$

2. *the hybrid rank of \mathcal{G} relative to Π_s equals*

$$\min(\overline{(|\Gamma_L| - 2)(\Pi)} + |V(\mathcal{G})| - |\Pi_s| + 1),$$

Π a partition of $V(\mathcal{G})$.

The problem of minimizing $\overline{(|\Gamma_L| - 2)(\Pi)}$ over partitions of $V(\mathcal{G})$ falls under computing ‘Dilworth truncation’ of a submodular function. The Principal Lattice of partitions problem is that of computing all partitions which minimize $(|\Gamma_L| - \lambda)(\cdot)$ for some λ . Strongly polynomial algorithms are available for this purpose. Details are available in [Narayanan90], [Narayanan91], [Narayanan97].

There are strong analogies between the principal partition (minimize $r(X) - \lambda|X|$, $r(\cdot)$ submodular, over subsets of a given set) and the principal lattice of partitions (minimize $\overline{(r - \lambda)(\cdot)}$, $r(\cdot)$ submodular, over partitions of a given set) problems. In the case of principal partition, if $\lambda_1 > \lambda_2$, a minimizing set corresponding to the former is always a subset of any minimizing set corresponding to the latter. In the case of principal lattice of partitions, if $\lambda_1 > \lambda_2$, a minimizing partition corresponding to the former is always finer than any minimizing partition corresponding to the latter [Narayanan97]. In addition, in the case of graphs, the principal partition problem of Kishi-Kajitani (on the edge set of a graph) can be posed, as described earlier, as that of finding an optimal node fusion- fission problem and therefore can actually be solved as a principal lattice of partitions problem on the vertex set of the graph [Narayanan97]. Indeed the current fastest principal partition algorithm for graphs is actually of this type [Patkar+Narayanan91].

1.8 Notes

The hybrid analysis notion is peculiar to network theory giving rise naturally to the hybrid rank problem. This problem and its generalizations can be regarded as unifiers for large parts of combinatorial optimization including the theory of submodular functions. This theme has been enlarged in [Narayanan97]. The principal partition and principal lattice of partitions have many practical applications particularly in building partitioners for large scale systems (see [Patkar+Narayanan09]) and in structural solvability of systems ([Ozawa76],[Ozawa+Kajitani79], see [Narayanan+Patkar09] for more references).

Bibliography

- [Brameller+John+Scott69] A. Brameller, M. John and M. Scott: *Practical Diakoptics for Electrical Networks* (Chapman and Hall, London, England, 1969).
- [Branin62] F.H.Branin Jr.: The relationship between Kron's method and the classical methods of network analysis. *Matrix and Tensor Quarterly* **12** (1962) 69-105.
- [Edmonds65a] J. Edmonds: Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards* **69B** (1965) 67-72.
- [Edmonds65b] J. Edmonds: Lehman's switching game and a theorem of Tutte and Nash-Williams. *ibid.* **69B** (1965) 73-77.
- [Iri69] M. Iri: The maximum rank minimum term rank theorem for the pivotal transformations of a matrix. *Linear Algebra and Its Applications* **2** (1969) 427-446.
- [Kishi+Kajitani69] G. Kishi and Y. Kajitani: Maximally Distant Trees and Principal Partition of a Linear Graph. *IEEE Transactions on Circuit Theory* **CT-16** (1969) 323-329.
- [Kron39] G. Kron: *Tensor Analysis of Networks* (J.Wiley, New York, 1939).
- [Kron63] G. Kron: *Diakoptics - Piecewise Solution of Large Scale Systems* (McDonald, London, 1963).
- [Narayanan79] H. Narayanan: A theorem on graphs and its application to network analysis. *Proceedings of IEEE International Symposium on Circuits and Systems* (1979) 1008-1011.
- [Narayanan87] H. Narayanan: Topological transformations of electrical networks. *International Journal of Circuit Theory and its Applications* **15** (1987) 211-233.

- [Narayanan90] H. Narayanan: On the minimum hybrid rank of a graph relative to a partition of its edges and its application to electrical network analysis. *International Journal of Circuit Theory and its Applications* **18** (1990) 269-288.
- [Narayanan91] H. Narayanan: The principal lattice of partitions of a submodular function. *Linear Algebra and its Applications* **144** (1991) 179-216.
- [Narayanan97] H. Narayanan: Submodular Functions and Electrical Networks. *Annals of Discrete Mathematics* **54** North Holland (London, New York, Amsterdam) (1997). (Revised version at <http://www.ee.iitb.ac.in/hn/book/>).
- [Narayanan07] H. Narayanan: Mathematical Programming and Electrical Network Analysis II: Computational Linear Algebra through Network analysis, International Symposium on Mathematical Programming for Decision Making: Theory and Applications (ISMPDM07), ISI Delhi, January 10-11, 2007.
- [Narayanan+Patkar09] H.Narayanan and S.B.Patkar: Matroids, chapter in the present book.
- [Ohtsuki+Ishizaki+Watanabe70] T.Ohtsuki, Y.Ishizaki and H.Watanabe: Topological Degrees of Freedom and Mixed Analysis of Electrical Networks. *IEEE Transactions on Circuit Theory* **CT-17** (1970) 491-499.
- [Ozawa76] T. Ozawa: Topological conditions for the solvability of active linear networks. *International Journal of Circuit Theory and its Applications* **4** (1976) 125-136.
- [Ozawa+Kajitani79] T. Ozawa and Y.Kajitani: Diagnosability of Linear Active Networks. *IEEE Transactions on Circuits and Systems* **CAS-26** (1979) 485-489.
- [Patkar+Narayanan91] S. Patkar and H. Narayanan: Fast algorithm for the principal partition of a graph. *Proceedings of Eleventh Annual Symposium on Foundations of Software Technology and Theoretical Computer Science (FSTTCS-11)* **LNCS-560** (1991) 288-306.
- [Patkar+Narayanan09] S.B.Patkar and H.Narayanan: Partitioning, chapter in the present book.
- [Siva09] V. Siva Sankar, H. Narayanan, and Sachin B. Patkar: Exploiting Hybrid Analysis in Solving Electrical Networks, 22nd International Conference on VLSI Design, New Delhi, January 5-9, 2009, 206-210.

Index

- cobase submatrix, 14
- coforest, 4
- coloop, 7
- current vector, 5

- dot product, 4

- forest, 4
- fundamental circuit, 6
- fundamental circuit matrix, 6
- fundamental cutset, 5
- fundamental cutset matrix, 5
- fusion-fission method, 20

- graph, 4
 - hybrid rank, 22

- hinge, 4
- hybrid rank problem, 13
- hybrid representation, 14

- incidence matrix, 5

- matrix
 - term rank, 14
- maximally distant forests
 - algorithm for, 16
- maximally distant trees, 14

- nullity, 4

- orthogonal vectors, 4

- partition
 - fission number, 22
 - fusion - fission number, 23
 - fusion number, 22
- potential vector, 5

- principal lattice of partitions, 4, 23
- principal partition, 4, 13

- rank, 4

- self loop, 7
- separator, 4
 - elementary, 4
- Shannon switching game, 14

- Tellegen's Theorem, 5
- topological degree of freedom, 14
- topological hybrid analysis, 8
- topological transformation, 20

- voltage vector, 5