Final version appears in Discrete Mathematics 114 (1993).

Reliable Assignments of Processors to Tasks and Factoring on Matroids

Charles J. Colbourn
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
Ehab S. Elmallah
Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2H1
Canada

Abstract

In the simple assignment problem, there are n processors, m tasks, and a relation between the processors and tasks; this relation indicates the ability of the processor to perform the task. When the processors fail independently with known probabilities, two performance issues arise. First, with what probability can the operating processors all be kept busy? Second, with what probability can the operating processors perform the same number of tasks that all processors could? We formulate these questions on the underlying transversal matroid. We first prove that counting minimum cardinality circuits in this matroid is #P-complete, and hence that both questions are also #P-complete. Secondly, we devise a factoring algorithm with series and parallel reductions to compute exact solutions of the above problems. We then outline some efficient strategies for bounding the probabilities.

1 Introduction

Consider a multiprocessor system having a set C= $c_1, c_2, \ldots, c_n$ of processors available for executing a set T= $t_1, t_2, \ldots, t_m$ of tasks at a certain time. During a time interval of interest, a processor c_i operates with a certain probability p_i independent of other processors. Each processor is capable of executing a certain subset of tasks in T, depending on its hardware configuration. However, during the time interval of interest a processor can only be assigned to execute at most one task. The situation can be modelled by forming a bipartite graph B on vertices C representing processors and T representing tasks. Vertices c_i and t_j are adjacent if and only if processor c_i is capable of performing task t_j. A valid assignment of processors to tasks corresponds to a *matching* in B. (For background on matchings, see [LP86]). we denote the cardinality of a maximum matching in B by $r(B)$.

There are two natural goals: to maximize processor utilization, and to maximize throughput (the number of tasks performed). To achieve both, we choose an assignment that is a *maximum* matching; we denote the cardinality of such a matching by $r(B)$. Two simplified reliability problems now arise. First, what is the probability that the operating processors can still perform $r(B)$ tasks? And what is the probability the operating processors can all be assigned tasks simultaneously? We refer to the two problems as the *task-reliability* ($TRel$) and the *processor-reliability* ($PRel$) problems, respectively.

Before formulating these questions more precisely, it is perhaps important to remark that the simple model introduced here omits more information that is essential in practical multiprocessor scheduling, most importantly time-dependent behaviour. Nevertheless, it captures the basics of the scheduling problem at one instant of time.

Define a *state* to be a subset S of the processors; we interpret that all processors in S are operating, while all others have failed. A state S is *processor-operational* if all processors in S can be assigned tasks simultaneously; in other words, the processors of S form the endvertices of edges in some matching of B. In this case, S is a *matchable set*. A state S is *task-operational* if using processors in this state, $r(B)$ tasks can be assigned; in other words, S contains at least a matchable set of cardinality $r(B)$.

The set of matchable sets in a bipartite graph form the independent sets of a matroid, the *transversal matroid* $TM(B)$. A basis of this matroid is precisely a matchable set of cardinality $r(B)$. Now if a state S is task-operational, S contains a base of $TM(B)$. The task-operational states are precisely the sets that *span* $TM(B)$ (i.e., those that contain at least a basis of $TM(B)$). Equivalently, $\bar{S}$ is a subset of a dual basis of $TM(B)$, and hence the task-operational states are precisely complements of the independent sets in the dual of $TM(B)$. Duals of transversal matroids form a special class of linking systems [Sc79] called *strict gammoids* (see for example [Mc72], [IP73] and [Br87]), and hence we denote the dual matroid by $SG(B)$.

Now we can formulate our reliability questions precisely. Two classes of reliability problems are of interest. In the general problem the operational probabilities of the elements in E are specified by a vector $\tilde{p}$. The special problem where all elements operate with the same probability p is called a *functional* reliability problem. Given a matroid $\mathcal{M} = (E, \mathcal{I})$ having a rank function $r(\mathcal{M})$ with $|E| = n$, the *independence polynomial* $Ind(\mathcal{M}; p)$ is the polynomial $\sum_{i=0}^{i=r} F_i p^i (1-p)^{n-i}$, where F_i is the number of independent sets of cardinality i in $\mathcal{M}$. The *span polynomial* $Span(\mathcal{M}; p)$ is the polynomial $\sum_{i=r}^{i=n} S_i p^i (1-p)^{n-i}$, where S_i is the number of sets of cardinality i that contain a basis of $\mathcal{M}$. By duality, $Ind(\mathcal{M}; p) = Span(dual(\mathcal{M}); 1-p)$.

For the functional reliability problems, we define the *processor reliability* $PRel(B; p)$ to be $Ind(TM(B); p)$; this is precisely the probability that the operating processors form a matchable set. Similarly, we define the *task reliability* $TRel(B; p)$ to be $Span(TM(B); p)$; again, this is the probability that $r(B)$ tasks can still be performed. Computing processor and task reliabilities are precisely the problems we introduced earlier. In the general case, the *Span* reliability of $\mathcal{M}$, denoted $SRel(\mathcal{M}, \tilde{p})$, is the probability of obtaining a spanset of $\mathcal{M}$. Thus, $TRel(B, \tilde{p})$ is just $SRel(TM(B), \tilde{p})$, and by duality, $PRel(B, \tilde{p})$ (the *Independence* reliability $IRel(TM(B), \tilde{p})$) is $SRel(dual(TM(B)), \mathbf{1} - \tilde{p})$, where $\mathbf{1}$ is a vector of 1's of length $|E|$.

Naturally, this translation to the matroid domain does not make the problems any easier; however, it does suggest employing techniques that have been useful in other matroid reliability problems. The primary example of this is the network reliability problem. Given a graph $G = (V, E)$ in which edges operate independently with probability p, the *functional all-terminal reliability* is the polynomial $Span(Gr(G); p)$ where $Gr(G)$ is the graphic matroid of G; more usually, this is formulated as $Ind(Cog(G); 1-p)$ where $Cog(G)$ is the cographic matroid of G. A very large literature exists on network reliability; see [Co87] for an introduction. In particular, a number of results on all-terminal reliability rely on matroid structure, and one can therefore hope to extend them to our problems here.

Before returning to processors and tasks, it is worth remarking on a further matroid reliability analysis problem, suggested in [CP89]. What is $Ind(Gr(G); p)$? It is the probability that the operating edges of G form an acyclic subgraph, and hence that the failed edges form an edge feedback set (set of edges whose removal destroys all cycles) for G. Thus $Ind(Gr(G); 1-p)$ is the probability that the "operating" edges form an edge feedback set. While we know of no concrete application for such a reliability computation, we expect that it may prove useful.

In the following, we consider the four matroid reliability analysis problems: task reliability, processor reliability, all-terminal reliability and acyclicity. We assume throughout that the matroid is presented as the corresponding graph.

2 Counting Independent and Spanning Sets

Determining any independence polynomial exactly amounts to determining each of its coefficients exactly (see for example [PB83]). If any coefficient is difficult to compute, so therefore is the polynomial. Let us first determine what information can be extracted from an independence polynomial. One can determine the rank r and the number of bases F_r of the matroid. One can determine the total number of independent sets. Now suppose that c is the cardinality of a smallest circuit of the matroid. For $i < c$, we must have $F_i = \binom{n}{i}$. If there are C_c circuits of cardinality c, $F_c = \binom{n}{c} - C_c$. Hence from the independence polynomial, we can determine c and C_c.

For cographic matroids, determining the number of bases is just counting spanning trees. The circuit size c is just the edge-connectivity of the graph. Finally, Bixby [Bi75] and Lomonosov and Polesskii [LP72] show that C_c is polynomially bounded (in fact, [DKL76] gives a structural description of the minimum cuts); computing it in polynomial time is then straightforward [RC87]. Nevertheless, all-terminal reliability is a #P-complete problem: computing the total number of independent sets is #P-complete [PB83].

For graphic matroids, counting bases is counting spanning trees. Determining c is computing the *girth* (the size of the shortest cycle) of the graph. Determining C_c, the number of shortest cycles, seems not to have been addressed; we outline an efficient algorithm here. First suppose c is even. Every shortest cycle contains $c/2$ pairs of vertices at distance $c/2$ in the graph. To count shortest cycles, for each pair of vertices at distance $c/2$, find all paths of length $c/2$ between these two vertices. These paths are necessarily internally vertex-disjoint since the girth is c. Hence there are $O(n)$ such paths, and any two form a c-cycle giving $O(n^2)$ c-cycles for this pair. Hence the total number of c-cycles is $O(n^4)$. (This is best possible; $K_{n,n}$ has $n^2(n-1)^2/4$ 4-cycles). When c is odd, pick a pair of vertices at distance $(c-1)/2$. This short path is necessarily unique. A cycle is completed by any path of length $(c+1)/2$; since there are $O(n)$ candidates for such a path, there are $O(n^3)$ shortest cycles in total. (This is also best possible, upon consideration of the complete graph). The factoring technique of [RC87] can be applied to compute the exact number C_c in polynomial time given these bounds on the magnitude. Perhaps surprisingly, for graphic matroids it does not appear to be known that computing the independence polynomial is computationally difficult.

Finally, we return to our main interest: transversal matroids. The main result we obtain in this case follows:

Theorem 2.1: Given a bipartite graph B, determining the minimum cardinality of a circuit of $TM(B)$ is NP-hard, and counting circuits of specified cardinality is #P-complete.

Proof:

We reduce the k-clique problem, $k \geq 3$, to the circuit problem in polynomial time. An instance of the clique problem is a graph $G = (V, E)$ and an integer k. Now let $\ell = \binom{k}{2} - k - 1 = \binom{k-1}{2}$. Define a bipartite graph B as follows. One class of the bipartition is E; the other is $V \cup \{z_1, \ldots, z_\ell\}$. Edges are placed between all $\{z_i\}$ and all vertices corresponding to E, and between vertex v and edge e if and only if edge e is incident to vertex v. Now choose a set $K \subseteq E$. Suppose that $|K| < \binom{k}{2}$. We claim that K

is matchable. To see this, we must only ensure that $|K| - \ell$ edges in K are incident with at least $|K| - \ell$ different vertices, as the remaining elements of K can be matched to the $\{z_i\}$. If K has fewer than ℓ edges, the result is immediate.

Otherwise, $\binom{k-1}{2} \leq |K| < \binom{k}{2}$ and K forms a subgraph H of G on at least $k-1$ vertices. Denote by H_t and H_c the union of the (possibly empty) acyclic and cyclic connected components of H, respectively. Thus, $H = H_t \cup H_c$. All the edges $E(H_t)$ can be matched to distinct vertices of H_t and all the vertices $V(H_c)$ can be matched to distinct edges in H_c. Thus, it suffices to show that $|E(H_t)| + |V(H_c)| \geq |K| - \ell$ in each of the following cases:

1. If $\binom{k-1}{2} < |E(H_c)|$ then $|V(H_c)| \geq k \geq |K| - \ell$.

2. If $\binom{k-2}{2} < |E(H_c)| \leq \binom{k-1}{2}$ then $|V(H_c)| \geq k-1 \geq |K| - \ell - 1$. Thus, if $|K| - \ell \leq k-1$ we are done, otherwise, $|K| - \ell = k$ and one more edge can be matched in H_t since $|E(H_t)| = |K| - |E(H_c)|$ $\geq |K| - \binom{k-1}{2} = |K| - \ell = k \geq 3$.

3. Finally, if $|E(H_c)| \leq \binom{k-2}{2}$ then $k \geq 4$ and $|E(H_t)| = |K| - |E(H_c)| > |K| - \binom{k-1}{2} = |K| - \ell$.

At this point, we know that there is no circuit of size less than $\binom{k}{2}$. Now consider a set K of size $\binom{k}{2}$. If K induces a k-clique, it is not matchable, since the neighbourhood of K in B contains at most $k + \ell < \binom{k}{2}$ vertices. If on the other hand K does not induce a k-clique, it must induce a subgraph with at least $k+1$ vertices, and is therefore matchable. This establishes that the k-cliques of G are in one-to-one correspondence with the k-circuits of B.

Since the clique problem is NP-hard, and counting cliques is #P-complete, we have the required results.

■

Corollary 2.2: Computing the processor reliability is #P-complete.
Proof:

Computing $Ind(TM(B);p)$ determines, among other things, the size and number of minimum cardinality circuits. ■

It is worth remarking that Theorem 2.1 establishes a complexity result of independent interest. We define a *Hall set* to be a set of vertices that is matchable to a unique set of vertices in the other class of the bipartition. Hall sets arise in algorithms for finding maximum matchings [LP86]. By incrementing ℓ by one in the proof of Theorem 2.1, we obtain

Corollary 2.3: Counting Hall sets of minimum cardinality is #P-complete.

Next we turn to task reliability. We expect that counting circuits is difficult here as well; this amounts to counting cocircuits ("cutsets") given the bipartite graph. However, no result of this type is known. Nevertheless, we can prove the following simple lemma:

Lemma 2.4: Provided that equal operation probabilities are not stipulated, computing processor reliability is polynomially reducible to computing task reliability.
Proof:

Given a bipartite graph $B = (X \cup Y, E)$ in which elements of X represent processors that operate with known probabilities, we construct a graph B' as follows. One class of the bipartition is X. The other contains Y and $|X|$ vertices $X' = \{x' : x \in X\}$. Vertices in Y are assigned operation probability 1. Each vertex x' is assigned $1 - p_x$, where p_x is the operation probability of x in B. Edges in B' between X and Y are as in B; edges between X and X' are all edges of the form $\{x, x'\}$ for $x \in X$. In B' the elements of $Y \cup X'$ represent processors and $r(B') = |X|$.

To complete the proof we show that processor reliability in B is task reliability in B'. This follows from the following remarks:

1. A state S is processor-operational in B if and only if the state $Y \cup (X' - S')$ is task operational in B', where $S' = \{x' : x \in S\}$.

2. The probability of having a state S in B equals the probability of having the state $Y \cup (X' - S')$ in B'. ∎

Performing contractions on the nodes of Y in B' leads to the strict gammoid $SG(B)$; however, this proof enables us to convert among transversal matroids, providing not all vertices need have the same operation probability.

At this point, we conclude that both task and processor reliability are computationally difficult problems. Hence we are left with serious problems: first, can we extend the well known class of graph factoring algorithms (see [Co87] for a background) to solve our present problems on matroids? Second, can we approximate, or bound, the reliabilities efficiently? We devote the remainder of the paper to the above questions.

3 The Factoring Algorithm

Using common reliability terminology, we call the system $(\mathcal{M}, \tilde{p})$ a *probabilistic matroidal system*. The bases and the spansets of $\mathcal{M}$ are called *minpaths* and *pathsets*, respectively, of the system. To simplify notation, p denotes a vector of probabilities throughout this section. We also need the following definitions on matroids (e.g. see [We76]). Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and X be a subset of E. The matroid obtained by deleting X is denoted $\mathcal{M} - X$, $\mathcal{M} - X = (E - X, \mathcal{I}^d)$ where $\mathcal{I}^d$ contains all subsets of $E - X$ that belong to $\mathcal{I}$. The matroid obtained by $contracting$ X is denoted $\mathcal{M} \bullet X$ (the notation $\mathcal{M}/X$ is also a common notation for the contracted matroid). If $X = \{e\}$, where e is an element in some basis of $\mathcal{M}$, then a set I is a basis of $\mathcal{M} \bullet X$ if and only if $I \cup \{e\}$ is a basis of $\mathcal{M}$. In general, the rank of a set S in $\mathcal{M} \bullet X$ is given by $r_{\mathcal{M} \bullet X}(S) = r_{\mathcal{M}}(S \cup X) - r_{\mathcal{M}}(X)$. The order of deleting a subset X of elements and contracting another disjoint subset Y to form a *minor* is immaterial.

A *pivotal decomposition* of the pathsets of $\mathcal{M}$ with respect to e is a partitioning into two disjoint subsets depending on whether or not e appears in a pathset. Hence, we have

$$SRel(\mathcal{M}, p) = p_e \; SRel(\mathcal{M} \bullet e, p) + (1 - p_e) \; SRel(\mathcal{M} - e, p) \tag{1}$$

where $SRel(\mathcal{M} \bullet e, p)$ and $SRel(\mathcal{M} - e, p)$ are the probabilities of obtaining a pathset in $E - e$ of $\mathcal{M} \bullet e$ and $\mathcal{M}$, respectively. The analogous expression for computing the all-terminal reliability of a graph G, $ARel(G, p)$,

$$ARel(G, p) = p_e \; ARel(G \bullet e) + (1 - p_e) \; ARel(G - e) \tag{2}$$

is well known. Here, e is any edge in the graph and $G - e$ and $G \bullet e$ denote the graphs obtained by deleting and contracting e respectively.

The factoring algorithm applies equation 1 recursively using a sequence of pivots. For this purpose, we rewrite equation 1 in a form suitable to describe the span reliability of any intermediate minor $\mathcal{M}_0$ in the algorithm. $\mathcal{M}_0$ is obtained from $\mathcal{M}$ be contracting a subset E_0^c and deleting some other subset. Any spanset of $\mathcal{M}_0$ can be extended to a spanset of $\mathcal{M}$ by adding the elements of E_0^c to it. We use the expanded notation

$SRel_{\mathcal{M}\bullet E_0^c}(\mathcal{M}_0, p)$ to denote the probability of obtaining a spanset of the matroid $\mathcal{M} \bullet E_0^c$ whose elements are in $\mathcal{M}_0$. Using the expanded notation, we may now write

$$SRel_{\mathcal{M}\bullet E_0^c}(\mathcal{M}_0, p) = \\ p_e \; SRel_{\mathcal{M}\bullet\{E_0^c \cup e\}}(\mathcal{M}_0 \bullet e, p) + (1 - p_e) \; SRel_{\mathcal{M}\bullet E_0^c}(\mathcal{M}_0 - e, p) \tag{3}$$

and $SRel(\mathcal{M}, p)$ is just $SRel_{\mathcal{M}\bullet\phi}(\mathcal{M}, p)$.

3.1 Series and Parallel Reliability Transformations

The SP-factoring algorithm reduces the work done by performing series and parallel reliability reductions whenever possible. To set a background, recall that a *bridge* in $\mathcal{M}$ is an element that is contained in every basis and a *loop* is a circuit of one element. Two elements e_1 and e_2 are in *series* if e_2 is a bridge in $\mathcal{M} - e_1$. Similarly, two non-loop elements e_1 and e_2 are in *parallel* if e_2 is a loop in $\mathcal{M} \bullet e_1$. Series and parallel relations are symmetric and transitive. Degenerate cases where any two non-loop elements of $\mathcal{M}$ are in series or in parallel occur when $r(\mathcal{M}) = |E|$ and $r(\mathcal{M}) = 1$, respectively. The above definition of series elements extends the classical definition of series edges in a graph, when applied to its forest matroid. For example, any two edges in a graph that form a cut-set are now considered in series. The following lemma specifies the exact transformations used.

Lemma 3.1: Let $(\mathcal{M}, p)$ be a probabilistic matroidal system and x and y be two of its elements.

1. *The Series Reduction*: if x and y are in series then contract x and assign the new probability $p'_y = p_x p_y / \alpha$, where $\alpha = p_x + p_y - p_x p_y$, to y; call the resulting probability vector p_1. Now, $SRel(\mathcal{M}, p) = \alpha \; SRel(\mathcal{M} \bullet x, p_1)$.

2. *The Parallel Reduction*: if x and y are in parallel then delete x and assign the new probability value $p_x + p_y - p_x p_y$ to y; call the resulting probability vector p_1. Then $SRel(\mathcal{M}, p) = SRel(\mathcal{M}, p_1)$.

Proof: Straightforward by using x and y as pivots in equation 1. ■

We now outline the important steps in an SP-factoring algorithm that performs reductions before and after each pivoting step. As shown below, four input parameters are used to specify a minor $\mathcal{M}_0$ of a given matroid $\mathcal{M}$ and its associated probability vector p_0 to the function $SRel$. $\mathcal{M}_0$ is assumed to be obtained from $\mathcal{M}$ be contracting E_0^c and deleting E_0^d from E.

Function $SRel(\mathcal{M}, E_0^c, E_0^d, p_0)$.

1. Starting with the system $(\mathcal{M}_0, p_0)$, repeat the following steps until no more reduction is possible: remove all possible loops, contract all possible bridges and perform all possible series and parallel reductions. Call the new system $(\mathcal{M}_1, p_1)$ where $\mathcal{M}_1 = (\mathcal{M} - E_1^d) \bullet E_1^c$. Denote by α_1 the product of the operation probabilities of all contracted bridges and the α parameters from any possible series reductions. If no bridge contraction or series reduction has been applied then set $\alpha_1 = 1$.

2. If $\mathcal{M}_1$ is empty then Return (α_1), otherwise, choose any element e, $e \in E_1$, as a pivot.

3. Repeat step (1) on the system $(\mathcal{M}_1 \bullet e, p_1)$. Call the new system $(\mathcal{M}_2, p_2)$ and the multiplicative parameter α_2. If $\mathcal{M}_2$ is empty then set $P_c = \alpha_2$. Else, $P_c = \alpha_2 \; SRel(\mathcal{M}, E_2^c \cup \{e\}, E_2^d, p_2)$.

4. Repeat step (1) on $(\mathcal{M}_1 - e, p_1)$. Call the new system $(\mathcal{M}_3, p_3)$ and the multiplicative parameter α_3. If $\mathcal{M}_3$ is empty then set $P_d = \alpha_3$. Else, $P_d = \alpha_3\ SRel(\mathcal{M}, E_3^c, E_3^d \cup \{e\}, p_3)$.

5. Return ($\alpha_1\ (p_e P_c + (1 - p_e) P_d)$).

6. End.

A measure of the efficiency of this algorithm is the number of calls to $SRel$; these calls can be represented in a binary "computation tree", denoted $T_{SP}(\mathcal{M})$. Hence, an equivalent measure is the number of leaves in this computation tree.

3.2 Recognizing Series and Parallel Reductions

An efficient strategy for identifying series and parallel elements throughout the algorithm follows from the three points mentioned below. The method requires an efficient method for computing the rank function of the given matroid $\mathcal{M}$ to decide whether any two given elements x and y are in series or in parallel in a minor $\mathcal{M}_0$, $\mathcal{M}_0 = (\mathcal{M} - E_0^d) \bullet E_0^c$.

1. Recall that $r(\mathcal{M}_0)$ is by definition $r_{\mathcal{M}_0}(E - \{E_0^d, E_0^c\})$. For any set S

$$r_{\mathcal{M}_0}(S) = r_{\mathcal{M}}(S \cup E_0^c) - r_{\mathcal{M}}(E_0^c).$$

2. x and y are in series if and only if

   (a) x and y are not bridges in $\mathcal{M}_0$, i.e. $r(\mathcal{M}_0) = r(\mathcal{M}_0 - x) = r(\mathcal{M}_0 - y)$ and

   (b) x is a bridge in $\mathcal{M}_0 - y$, i.e. $r(\mathcal{M}_0 - \{x, y\}) < r(\mathcal{M}_0)$.

3. x and y are in parallel if and only if

   (a) neither x nor y is a loop in $\mathcal{M}_0$, i.e. $r_{\mathcal{M}_0}(x) = r_{\mathcal{M}_0}(y) = 1$ and

   (b) y is a loop in $\mathcal{M}_0 \bullet x$, i.e. $r_{\mathcal{M}_0 \bullet x}(y) = 0$.

Correctness of the above remarks follows immediately from the definitions. Finding a possible pair of series or parallel elements can then be accomplished by testing all possible pairs of elements. Since testing any such pair requires a fixed number of evaluations of the rank function, one can easily obtain the following timing result.

Lemma 3.2: Let $\mathcal{M}$ be a matroid on n elements whose rank function can be computed in $O(f(n))$ time, for some function $f(n)$, and let $\mathcal{M}_0$ be one of its minors. One can decide whether a pair of the n_0 elements of $\mathcal{M}_0$ is in series or in parallel in time $O(n_0^2 f(n))$.

In the case of the underlying transversal matroid $\mathcal{M}$ of a bipartite graph $B = (T, C)$, the rank of a subset S of C is the size of a maximum matching in the subgraph induced by (T, S). Maximum matchings in bipartite graphs can be computed in $O(n^{2.5})$ time [HK73]. The same algorithm can be used to compute the rank of any subset S in the dual matroid $\mathcal{M}^*$ of $\mathcal{M}$ using $r_{\mathcal{M}^*}(S) = |S| - r(\mathcal{M}) + r_{\mathcal{M}}(E - S)$.

3.3 Performance of the Algorithm

In general, the number of nodes in a computation tree generated by the factoring algorithm grows exponentially with the number of elements in the system. However, by a careful choice of the pivoting elements one may obtain substantial improvements for systems that can be greatly reduced using series and parallel reductions. To gain more insight into the situation, we start by recalling that an upper bound on the minimum possible number of leaf nodes of $T_{SP}(\mathcal{M})$ can be obtained by evaluating any invariant function $f(\mathcal{M})$ that obeys

(A-i) the deletion-contraction rule: for any nonloop element e, $f(\mathcal{M}) = f(\mathcal{M} - e) + f(\mathcal{M} \bullet e)$ and

(A-ii) $f(B) = 1$ for any basis B of $\mathcal{M}$ and $f(B) = 0$ if B is a proper subset of a basis.

The function $\#B(\mathcal{M})$ whose value is the number of bases of a given matroid $\mathcal{M}$ is an obvious example that satisfies conditions (A-i) and (A-ii). However, one may not expect this function to give a tight upper bound on the number of leaves of $T_{SP}(\mathcal{M})$ since its value is not generally preserved under series and parallel reductions. That is, if $\mathcal{M}^*$ is obtained from $\mathcal{M}$ by a series or a parallel reduction then $\#B(\mathcal{M}^*)$ is usually less than $\#B(\mathcal{M})$ by a nonconstant factor, depending on $\mathcal{M}$.

In a search for a more useful function, Satyanarayana and Chang [SC83] have studied the *domination* function on graphs with remarkable results. The definition of such a function, as will be shown shortly, depends on the set of minpaths defining the reliability problem under consideration. For the K-terminal reliability problem, [SC83] have shown that the domination function satisfies conditions (A-i) and (A-ii) above. Moreover, it is invariant under parallel reductions and a special type of series reductions. Other extensions of such results appear in [AS84] and [Wo85]. We also refer the reader to [AB84] for a related survey. Our main results in this section is to show that similar results hold for the domination function defined with respect to the span reliability of matroids.

For convenience, we start by reproducing some definitions from [SC83] and [AB84] when applied to a matroid $\mathcal{M}$. An element e is said to be *irrelevant* if it is not contained in any minpath (base), otherwise, it is *relevant*. A *formation* of a subset E' of elements having no irrelevant element is a set of minpaths whose union is E'. A subset may have no formation if at least one of its elements is irrelevant. In general, there can be more than one possible formation for E'. A formation is odd or even depending on whether the number of its elements (minpaths) is odd or even. The *signed domination* of E' with respect to the set $Bases(\mathcal{M})$, denoted $sdom(E', Bases(\mathcal{M}))$, is the number of odd formations minus the number of even formations. If E' does not have any formation then its signed domination equals zero. The *domination* of E', denoted $DOM(E', Bases(\mathcal{M}))$, equals the absolute value of the signed domination. For convenience, let $sdom(\mathcal{M})$ ($DOM(\mathcal{M})$) denote $sdom(E, Bases(\mathcal{M}))$ ($DOM(E, Bases(\mathcal{M}))$, respectively).

From the above definitions, it is immediate that the DOM function satisfies condition (A-ii) above for matroids. To show that it obeys the deletion-contraction rule we use an elegant result of Barlow [Ba82] (mentioned also in [AB84]) on the signed dominations of *coherent systems*, where

Definition. *A coherent system is a pair (E, ρ) where E is a finite set of elements and $\rho = \{P_1, \ldots, P_k\}$ is a family of subsets of E such that*

(B-i) no P_i is contained in another member of the family and

(B-ii) $E = \cup_{i=1}^{k} P_i$.

Naturally, we will use for ρ the minpaths of a matroid. Following [AB84], a pivotal decomposition of the set ρ using an element e yields the two subsets $\rho(\bar{e}) = \{P_i | e \notin P_i \ and \ P_i \in \rho\}$ and $\rho(e) = \{P_i | e \in P_i \ and \ P_i \in \rho\}$ corresponding to the cases where e is failed and e is operating, respectively. The system $(E - e, \rho(\bar{e}))$ associated with the set $\rho(\bar{e})$ might not be coherent since $\cup_{P_i \in \rho(\bar{e})} P_i$ might be a proper subset of $E - e$. The system $(E - e, \eta(\rho - e))$ corresponding to e operating is defined as follows. First, let $\rho - e = \{P_1 - e, \ldots, P_k - e\}$. Second, let $\eta(\rho - e)$ be the set simplification of $\rho - e$, that is $\eta(\rho - e)$ is obtained from $(\rho - e)$ by omitting a set $P_i - e$ if it contains another set $P_j - e$. For example, if $\rho - e = \{(1,2),(1)\}$ then $\eta(\rho - e) = \{(1)\}$. Barlow's signed domination theorem can now be stated

$$sdom(E, \rho) = sdom(E - e, \eta(\rho - e)) - sdom(E - e, \rho(\bar{e})) \tag{4}$$

If the system (E, ρ) fails to satisfy condition (B-ii) then E has no formation, the left hand side of equation 4 is, by definition, zero, and the two terms on the right hand side are equal.

Our objective now is to show that a specialized form of equation 4 exists for matroids, using the contraction and deletion operations. Clearly, for a matroid $\mathcal{M}$ and one of its nonbridge elements e, if $\rho(\bar{e}) = \{P_i | e \notin P_i \ and \ P_i \in Bases(\mathcal{M})\}$ the two systems $(E - e, \rho(\bar{e}))$ and $(E - e, Bases(\mathcal{M} - e))$ are equal. Otherwise, (if e is a bridge) then $(E - e, \rho(\bar{e})) = (E - e, \phi)$. The remaining ingredient is

Lemma 3.3: Let $\mathcal{M}$ be a matroid and e be one of its elements then

$$(E - e, Bases(\mathcal{M} \bullet e)) = (E - e, \eta(Bases(\mathcal{M}) - e)).$$

Proof:

We show that $Bases(\mathcal{M} \bullet e) = \eta(Bases(\mathcal{M}) - e)$. The statement follows easily if e is a loop. So, assume that e is a not a loop. If $P \in Bases(\mathcal{M} \bullet e)$ then $P + e \in Bases(\mathcal{M})$ and hence $P \in Bases(\mathcal{M}) - e$. Moreover, there is no element in $Bases(\mathcal{M}) - e$ that is contained in P, so $P \in \eta(Bases(\mathcal{M}) - e)$.

Now, assume that $P \in \eta(Bases(\mathcal{M}) - e)$. Two possibilities arise: (1) $P + e$ is a basis of $\mathcal{M}$ and (2) P is a basis of $\mathcal{M}$ and $e \notin P$. In the first case, $P \in Bases(\mathcal{M} \bullet e)$ follows by definition. The second case leads to a contradiction. To see this, note that for some element x in P the set $P + e - x$ is a basis of $\mathcal{M}$, this implies that $P - x \in Bases(\mathcal{M}) - e$. But $P - x \subset P$, contradicting the assumption that $P \in \eta(Bases(\mathcal{M}) - e)$. ■

Barlow's result then implies:

Lemma 3.4: Let $\mathcal{M}$ be a matroid and e be one of its elements then

$$sdom(\mathcal{M}) = sdom(\mathcal{M} \bullet e) - sdom_{\mathcal{M}}(\mathcal{M} - e) \tag{5}$$

where $sdom_{\mathcal{M}}(\mathcal{M} - e) = sdom(E - e, Basis(\mathcal{M}))$ $(= sdom(\mathcal{M} - e)$ if e is not a bridge).

Equation 5 opens the way to prove that the domination function obeys the deletion-contraction rule (the proof follows [SC83] very closely):

Lemma 3.5: Let $\mathcal{M}$ be a matroid on n elements. Then for any element e

1. $sdom(\mathcal{M}) = (-1)^{n - r(\mathcal{M})} DOM(\mathcal{M})$ and
2. if e is a not a loop then $DOM(\mathcal{M}) = DOM(\mathcal{M} \bullet e) + DOM_{\mathcal{M}}(\mathcal{M} - e)$.

Proof:

1. By induction on $|E| = n$. The statement follows easily for any matroid on a single element. Assume (1) holds for all matroids with fewer than n elements and let $\mathcal{M}$ be a matroid with n elements. If $r(\mathcal{M}) = 0$ then $sdom(\mathcal{M}) = 0$ and the statement follows easily. Otherwise, let e be a non-loop element in $\mathcal{M}$. By equation 5 and the induction hypothesis one may write

$$\begin{aligned} &sdom(\mathcal{M}) = \\ &\quad (-1)^{n-1-r(\mathcal{M}\bullet e)} DOM(\mathcal{M} \bullet e) - (-1)^{n-1-r(\mathcal{M}-e)} DOM_{\mathcal{M}}(\mathcal{M} - e) \end{aligned} \tag{6}$$

   If e is not a bridge then $r(\mathcal{M} \bullet e) = r(\mathcal{M}) - 1$ and $r(\mathcal{M} - e) = r(\mathcal{M})$. the right hand side of equation 6 simplifies to

$$\begin{aligned} &= (-1)^{n-r(\mathcal{M})} DOM(\mathcal{M} \bullet e) - (-1)^{n-1-r(\mathcal{M})} DOM_{\mathcal{M}}(\mathcal{M} - e). \\ &= (-1)^{n-r(\mathcal{M})} (DOM(\mathcal{M} \bullet e) + DOM_{\mathcal{M}}(\mathcal{M} - e)) \end{aligned} \tag{7}$$

   If e is a bridge then $r(\mathcal{M} \bullet e) = r(\mathcal{M}) - 1$, $sdom(\mathcal{M} \bullet e) = sdom(\mathcal{M})$ and $sdom_{\mathcal{M}}(\mathcal{M} - e) = 0$. Again equation 7 holds. Taking the absolute value of both sides yields

$$DOM(\mathcal{M}) = DOM(\mathcal{M} \bullet e) + DOM_{\mathcal{M}}(\mathcal{M} - e)$$

   and statement (1) follows.

2. Statement (2) follows easily from the proof of part (1). ■

A second immediate consequence of equation 5 is that the value of the domination function is preserved under parallel reductions since for any two parallel elements x and y, $sdom(\mathcal{M} \bullet x) = 0$. Thus, the number of leaves of a tree $T_p(\mathcal{M})$ resulting from a P-factoring algorithm is exactly $DOM(\mathcal{M})$. In this latter case, a $T_p(\mathcal{M})$ tree obtained by choosing at each step a pivot that results in two substructures having nonzero dominations has the minimum possible total number of nodes. Any element whose contraction does not create a loop satisfies this requirement.

On the other hand, domination is not in general invariant under series reductions. Satyanarayana and Chang, however, identified a special case of series reductions in which a similar statement on $T_s(\mathcal{M})$ holds. The special case arises in computing the K-terminal reliability of a graph G having a vertex v of degree 2, $v \notin K$, incident to two edges e and e'. Here, $DOM(G - e) = 0$ since e' becomes irrelevant. Hence, domination is preserved under this type of reductions. Naturally, the same argument does not hold for any vertex v, $v \in K$, of degree 2. An analogous situation does not seem to apply for the $TRel$ or the $PRel$ problems.

4 Packing Bases and Circuits

One main strategy for obtaining bounds is to attempt to bound each coefficient of the independence polynomial. The most powerful current method known that applies to matroids is due to Ball and Provan [BP82]. However, one of the primary required pieces of information in the upper bound is the size of a minimum cardinality circuit. In view of Theorem 2.1, then, we do not expect to find useful upper bounds here; this contrasts with network reliability where the Ball-Provan bounds are among the best efficiently computable bounds currently available. For lower bounds, application of the Ball-Provan method requires knowledge of the number of bases. At present, no efficient algorithm is known for counting bases in transversal matroids. Hence we resort to other techniques.

Let $\mathcal{M} = (X, \mathcal{I})$ be a matroid. A *packing* of $\mathcal{M}$ by bases (circuits) is a collection of disjoint bases (resp., circuits) of $\mathcal{M}$. Suppose that $X_1, \ldots, X_s$ is a packing of $\mathcal{M}$ with bases. If any basis operates, the overall state must be operational; since the bases chosen are disjoint, they operate independently. Hence we have that

$$SRel(\mathcal{M}, \tilde{p}) \geq 1 - \prod_{i=1}^{s} \left[1 - \prod_{x \in X_i} p_x \right].$$

The above inequality has been derived and used for graphic matroids in [Po70]; see [Co87] for some generalizations to other reliability problems on graphs. *Any* packing by bases leads to a lower bound on the independence probability; close inspection shows that a better bound is obtained by taking more bases. Using Edmonds's matroid partition algorithm [Ed65], we can pack with the maximum number of bases efficiently. Thus for any matroid reliability problem, we can obtain an efficient lower bound. Remark that lower bounds on span probabilities lead to lower bounds on independence probabilities in the dual, and vice versa. One very important remark here is that the method can be applied even when operation probabilities are allowed to be different.

A similar strategy can be applied to packings by circuits. Let $C_1, \ldots, C_t$ be a packing by circuits. If all elements in any one of the circuits are chosen, the set cannot be independent. We have

$$IRel(\mathcal{M}, \tilde{p}) \leq \prod_{i=1}^{t} \left[1 - \prod_{x \in C_i} p_x \right].$$

Other types of upper bounds in this direction appear in [Lo74] (see also [Co87]). Once again, any packing by circuits gives an upper bound that is computable efficiently. However, finding packings with the most circuits is apparently much more difficult. For graphic matroids, this is the problem of edge-partition into cycles, and is known to be NP-hard [Ho81]. For cographic matroids, the problem is edge-packing by network cutsets, and is also known to be NP-hard [Co88]. No previous research on this problem for transversal matroids has been done; however, the problem is hard here as well:

Theorem 4.1: Deciding whether a transversal matroid (presented as a bipartite graph) can be packed with at least m circuits is NP-hard.

Proof:

Holyer [Ho81] proves that, for any fixed $k \geq 3$, determining whether a graph can be edge-partitioned into k-cliques is NP-hard. Colbourn [Co84] used Holyer's method to show that determining whether a tripartite graph has an edge partition into triangles is NP-hard. Suppose that such a graph has tripartition $X \cup Y \cup Z$, and edge set $E = E_{XY} \cup E_{XZ} \cup E_{YZ}$ (with the obvious interpretation). Form a bipartite graph B with one class being $X \cup Y \cup Z$, and the other containing E together with E'_{XY}, a second copy of the edges in E_{XY}. A vertex and an "edge", primed or not, are adjacent in B if they are incident in the original graph. Let $TM(B)$ be defined on the set $E \cup E'_{XY}$.

The minimum size of a circuit in $TM(B)$ is four and $TM(B)$ can be packed with at least $|E \cup E'_{XY}|/4$ disjoint circuits if and only if every circuit is of the form $\{\{x, y\}, \{x, z\}, \{y, z\}, \{x, y\}'\}$. Now, such a packing of $TM(B)$ with circuits is precisely an edge-partition of the original graph into triangles. ■

In all of these cases, one can still obtain useful bounds by adopting a greedy strategy to construct a packing by circuits; however, the complexity results limit the accuracy one can hope to achieve by such heuristics.

5 Concluding Remarks

In this first study of reliability in assignment problems, we have found a number of striking similarities with the network reliability problem. Largely, these are a consequence of the matroid structures of the problems, and hence we have highlighted that structure here. As a direct consequence, we have been able to devise a factoring algorithm with series and parallel reductions. We recall that the class of matroids in which each member can be constructed using series and parallel extensions have been studied in [Br71] as a generalization of the well known class of series-parallel graphs; [Ox86] mentions some interesting relations along this line. We have also encountered some important differences. The biggest difference is the difficulty of determining the size and number of minimum circuits.

Some other reliability problems that can be analyzed in a similar way are now in order. Consider the problem of scheduling a set of tasks on a single processor system. Each task is assumed to require one time unit and has a *release* time r_i and a *deadline* time d_i, $d_i > r_i$. Each task is available to the processor with a given probability. The situation can be formulated on a bipartite graph having an edge between a vertex τ_i representing the ith time unit of the processor and a task t_i if $r_i \leq \tau_i \leq d_i$.

Here, the $TRel$ problem corresponds to computing the probability of finishing the maximum possible number of tasks before the deadlines. Equivalently, it is the probability of using the maximum number of time slots, and hence keeping the processor as busy as possible. Similarly, $PRel$ corresponds to the probability that each available task can be completed on time.

Second, we observe that our assignment problem can be extended to model a situation where processors require different classes of computational resources $(R_1, R_2, \ldots, R_k)$ to operate. Such resources may correspond for example to the availability of adequate space in the hierarchy of the memory system. A computing element can only produce a useful work if it is operational and is matched to a job and a computational resource in each of the k classes. It then follows that a set of processors is active if each processor can be matched in each of the bipartite graphs (C, J), (C, R_1), …, (C, R_k). The resulting structure is the intersection of $k + 1$ transversal matroids; this case warrants more research.

Despite the inherent complexity of assigning tasks to unreliable processors, we have been able to develop efficiently computable bounding methods for task and processor reliability. We expect that the development of improvements on the methods here would be fruitful, both in theoretical questions on transversal matroids, and practical concerns with task-processor assignment.

Acknowledgements

We thank Bill Pulleyblank for many stimulating discussions on this research, Janelle Harms for pointing out the construction in Lemma 2.4, and the referees for many helpful comments. Research of the authors is supported by NSERC Canada under grant number A0579 (CJC) and OGP36899 (ESE).

References

[AB84] A. Agrawal and R.E. Barlow, "A Survey of Network Reliability and Domination Theory", *Operations Research* 32, 3 (1984) 478-492.

[AS84] A. Agrawal and A. Satyanarayana, "An $O(|E|)$ Time Algorithm for Computing the Reliability of a Class of Directed Networks", *Operations Research* 32, 3 (1984) 493-515.

[BP82] M.O. Ball and J.S. Provan, “Bounds on the reliability polynomial for shellable independence systems”, *SIAM J. Alg. Disc. Meth.* 3 (1982) 166-181.

[Ba82] R.E. Barlow, “Set Theoretic Signed Domination for Coherent Structures”, *Report ORC 82-1, Univ. California Berkeley*, 1982.

[Bi75] R.E. Bixby, “The minimum number of edges and vertices in a graph with edge-connectivity n and m n-bonds”, *Networks* 5 (1975) 253-298.

[Br87] R.A. Brualdi, “Transversal matroids”, in: *Combinatorial Geometries* (N. White, editor) Cambridge University Press, Cambridge (1987) 72-97.

[Br71] T. Brylawski, “A Combinatorial Model for Series-Parallel Networks”, *Trans. of the American Math. Soc.* 154 (1971) 1-22.

[Co84] C.J. Colbourn, “The complexity of completing partial latin squares”, *Discrete Applied Math.* 8 (1984) 25-30.

[Co87] C.J. Colbourn, *The Combinatorics of Network Reliability,* Oxford University Press, 1987.

[Co88] C.J. Colbourn, “Edge-packings of graphs and network reliability”, *Discrete Math.* 72 (1988) 49-61.

[CP89] C.J. Colbourn and W.R. Pulleyblank, “Matroid Steiner problems, the Tutte polynomial and network reliability”, *J. Combinatorial Theory* (B) 41 (1989) 20-31.

[DKL76] E.A. Dinic, A.V. Karzanov, M.V. Lomonosov, “The Minimum Cuts’ Structure of a Graph”, in: *Issledovaniya po diskretnoi optimizaccii, Moscow, Nauka* (1976) 290-306 (Russian).

[Ed65] J. Edmonds, “Minimum partition of a matroid into independent subsets”, *J. Res. NBS* B68 (1965) 67-72.

[Ho81] I. Holyer, “The NP-completeness of some edge-partition problems”, *SIAM J. Computing* 10 (1981) 713-717.

[HK73] J. Hopcroft and R.M. Karp, “An $n^{2.5}$ Algorithm for Maximum Matching in Bipartite Graphs”, *SIAM Journal on Computing* 2 (1973) 223-231.

[IP73] A.W. Ingleton and M.J. Piff, “Gammoids and Transversal Matroids”, *J. of Combinatorial Theory* (B) 15 (1973) 51-68.

[Lo74] M.V. Lomonosov, “Bernoulli Scheme with Closure”, *Problemy Peredachi Informacii,* 10, 1 (1974) 91-101 (Russian).

[LP72] M.V. Lomonosov and V.P. Polesskii, “Lower bounds of network reliability”, *Problems of Information Transmission* 8 (1972) 118-123.

[LP86] L. Lovász and M.D. Plummer, *Matching Theory,* North-Holland, Amsterdam, 1986.

[Mc72] C. McDiarmid, “Strict Gammoids and Rank Functions”, *Bull. London Math. Soc.* 4 (1972) 196-198.

[Ox86] J. Oxley, *"Graphs and Series-Parallel Networks"*, in Theory of Matroids, ed. by White N., Cambridge University Press, 1986.

[Po70] V.P. Polesskii, "On a Method for Constructing Reliable Communication Networks", in Discrete Automata and Communication Networks, Moscow, *Nauka,* (1970) 13-19. (Russian).

[PB83] J.S. Provan and M.O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected", *SIAM J. Computing* 12 (1983) 777-788.

[RC87] A. Ramanathan and C.J. Colbourn, "Counting almost minimum cutsets with reliability applications", *Math. Programming* 39 (1987) 253-261.

[SC83] A. Satyanarayana and M.K. Chang, "Network Reliability and the Factoring Theorem", *Networks* 13 (1983) 107-20.

[Sc79] A. Schrijver "Matroids and Linking Systems", *J. of Combinatorial Theory* (B) 26 (1979) 349-369.

[We76] D.J.A. Welsh, *Matroid Theory,* Academic Press, London, 1976.

[Wo85] R.K. Wood, "A Factoring Algorithm Using Polygon-to-Chain Reductions for Computing K-Terminal Network Reliability", *Networks* 15 (1985) 173-190.