Randomness-Optimal Unique Element Isolation, With Applications to Perfect Matching and Related Problems

Suresh Chari*
Pankaj Rohatgi**
Aravind Srinivasan***

TR 93-1359
June 1993

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Supported in part by NSF grant CCR-9123730.
**Supported in part by NSF grant CCR-9123730 and an IBM Graduate Fellowship. Also supported in part by the United States Army Research Office through the Army Center of Excellence for Symbolic Methods in Algorithmic Mathematics (ACSyAM), Mathematical Sciences Institute of Cornell University Contract DAAL03-91-C-0027.
***Supported in part by NSF PYI award CCR-89-96272 with matching support from UPS and Sun Microsystems, and by an IBM Graduate Fellowship.

Randomness-Optimal Unique Element Isolation, with Applications to Perfect Matching and Related Problems

Suresh Chari* Pankaj Rohatgi† Aravind Srinivasan‡
Department of Computer Science
Cornell University
Ithaca, NY 14853

Abstract

In this paper, we precisely characterize the randomness complexity *of the unique element isolation problem, a crucial step in the RNC algorithm for perfect matching due to Mulmuley, Vazirani & Vazirani[21] and in several other applications. Given a set S and an unknown family* $\mathcal{F} \subseteq 2^S$ *with* $|\mathcal{F}| \leq Z$*, we present a scheme to assign polynomially bounded weights to the elements of S, using only* $O(\log Z + \log |S|)$ *random bits, such that the minimum weight set in* $\mathcal{F}$ *is unique with high probability. This generalizes and improves the results of Mulmuley, Vazirani & Vazirani who give a scheme which uses* $O(S \log S)$ *random bits independent of Z. We also prove a matching lower bound for the randomness complexity of this problem. This new weight assignment scheme yields a randomness–efficient* RNC^2 *algorithm for perfect matching which uses* $O(\log Z + \log n)$ *random bits where Z is any* given *upper bound on the number of perfect matchings in the input graph. This generalizes the result of Grigoriev & Karpinski[11] who present an* NC^3 *algorithm when Z is polynomially bounded and also gives an improvement on the running time in this case. The worst–case randomness complexity of our algorithm is* $O(n \log(m/n))$ *random bits, as opposed to the previous bound of* $O(m \log n)$ *bits. Our technique also gives randomness–efficient solutions for several problems in which the unique element isolation tool is used, such as RNC algorithms for variants of matching and basic problems on linear matroids such as matroid intersection and matroid matching. We also obtain a randomness–efficient alternative to the random reduction from SAT to USAT, the language of uniquely satisfiable formulas, due to Valiant and Vazirani[32]. This reduction can be derandomized in the case of languages in FewP to yield new proofs of the results* $FewP \subseteq \oplus P$ *and* $FewP \subseteq C_{=}P$*.*

*Supported in part by NSF grant CCR-9123730.

†Supported in part by by NSF grant CCR-9123730 and an IBM Graduate Fellowship. Also supported in part by the United States Army Research Office through the Army Center of Excellence for Symbolic Methods in Algorithmic Mathematics (ACSyAM), Mathematical Sciences Institute of Cornell University Contract DAAL03-91-C-0027.

‡Supported in part by NSF PYI award CCR-89-96272 with matching support from UPS and Sun Microsystems, and by an IBM Graduate Fellowship.

1 Introduction

Given a graph G, a *matching* in G is a subset of the edges such that no two edges in the subset are incident on the same vertex. A *maximum matching* in G is a matching in G of maximum cardinality, and a *perfect matching* is a special type of maximum matching in which each vertex is covered by an edge, *i.e.*, each vertex is *matched.* Finding a perfect matching, if any, in a graph is one of the fundamental problems in combinatorial optimization. While its sequential complexity has been well-studied (Lovász & Plummer [19]), a big open question in the theory of parallel algorithms is whether a perfect matching can be found or even detected in NC. Due to its importance, considerable effort has been devoted towards developing parallel algorithms for the perfect matching problem. For instance, sublinear time parallel algorithms for general graphs (Goldberg, Plotkin & Vaidya [10], Vaidya [31], Grover [12]) and for bipartite graphs (Goldberg, Plotkin, Shmoys & Tardos [9]) are known. NC algorithms have also been developed for special classes of graphs such as co–comparability graphs (Kozen, Vazirani & Vazirani [17]), strongly chordal graphs (Dahlhaus & Karpinski [7]), graphs with polynomially many perfect matchings (Grigoriev & Karpinski [11]), and planar bipartite graphs (Miller & Naor [20]). A very successful approach for the general problem has been to use randomness: both Monte Carlo RNC algorithms (Karp, Upfal & Wigderson [14], Mulmuley, Vazirani & Vazirani [21]) and Las Vegas RNC algorithms (Karloff [13]) have been developed.

In this paper, we investigate the *parallel randomness complexity* of perfect matching and related problems, *i.e.*, the amount of randomness required to solve them in parallel. For perfect matching we present an RNC^2 algorithm which uses $O(\log Z + \log n)$ random bits where Z is any *given* upper bound on $|\mathrm{Mat}(G)|$, the number of perfect matchings in the graph G. This improves and generalizes the result of Grigoriev & Karpinski [11] who give an NC^3 algorithm when Z is polynomially bounded. Since a graph on n vertices with m edges can have at most $(2m/n)^n$ perfect matchings, the worst case randomness complexity of our algorithm is $O(n\log(m/n))$ which improves on the previous bound of $O(m\log n)$ due to Mulmuley, Vazirani & Vazirani [21]. Thus by linking the randomness complexity of this problem to the number of perfect matchings, we unify and generalize previous results, while also improving on them. In special cases, *e.g.*, $K_{3,3}$–free graphs, the number of perfect matchings in the graph can be computed in NC^2 (see for example Vazirani [33] and also the work of Kastelyn[15] and Little [18]); in general, if no good upper bound on $|\mathrm{Mat}(G)|$ is known, our results yield an RNC^3 algorithm to find a perfect matching which uses at most $O(\log|\mathrm{Mat}(G)|)$ random bits with high probability, if the input graph has at least one matching. This is a significant reduction if $0 < |\mathrm{Mat}(G)| \ll (2m/n)^n$. All these results are obtained from a *randomness–optimal* generalization of the Isolating Lemma, a tool used to isolate a perfect matching in the RNC algorithm of Mulmuley, Vazirani and Vazirani[21]. Since this abstract and powerful tool has several applications such as to basic problems on linearly representable matroids, variants of matching and random reductions in structural complexity theory, our generalization results in randomness–efficiency in these settings as well.

Given a set $S = \{x_1, x_2, \ldots, x_N\}$ and an *unknown* family $\mathcal{F} \subseteq 2^S$, the Isolating Lemma of [21] shows that if the x_i's are *independently* assigned weights uniformly at random from the range $\{1, 2, \ldots, 2N\}$, then with probability at least $1/2$, the minimum weight set in $\mathcal{F}$ is *unique.* This weight assignment scheme clearly requires $\Theta(N \log N)$ random bits and an question left open in the original paper was whether the assumption that the weights be assigned independently is necessary. Our scheme generalizes this as follows: given an upper bound Z on $|\mathcal{F}|$, we assign polynomially bounded weights to the x_i's using only $O(\log Z + \log N)$ random bits and achieve the same result. In the worst case ($Z = 2^N$), our scheme needs $O(N)$ random bits as compared to $\Theta(N \log N)$; more importantly, for smaller Z, we get much better randomness complexity. Also, since even in the worst case we assign polynomial weights to N different random variables using $O(N)$ random bits, the weight assignments are *not* independent, thus settling the open question of [21]. Our scheme provides an explicit construction of a collection of $(NZ)^{O(1)}$ weight assignments such that, for every family $\mathcal{F}$ of size at most Z, at least one (in fact, at least 50%) of these assignments makes the minimum weight subset in $\mathcal{F}$ unique. Such a collection of size $O(NZ)$ is guaranteed to exist by applying Adleman's technique [1] to the original Isolating Lemma, and thus this is one of the few instances where Adleman's result can be made constructive with only a polynomial blowup in size. Furthermore, we also show that our weight assignment scheme is optimal by proving a matching lower bound for this generalization, *i.e.*, any scheme which assigns polynomial weights to the x_i's must use $\Omega(\log Z + \log N)$ random bits even to achieve isolation with only *nonzero* probability.

This randomness–efficient generalization of the isolation process can be directly plugged into the settings where the original Isolating Lemma has been used to obtain randomness efficiency. Besides the new algorithm for perfect matching we get randomness–efficient algorithms for variants of perfect matching such as exact matching and maximum matching. We get RNC^2 algorithms for exact and maximum matching which use $O(\log Z + \log n)$ random bits, where Z is a *given* upper bound on the number of exact matchings and maximum matchings respectively. Matroid intersection and matroid matching are generalizations of bipartite and general graph matching, and the Isolating Lemma has been used to obtain RNC^2 algorithms for these basic problems on linearly representable matroids (Narayanan, Saran & Vazirani [22]). Here, by obtaining good bounds on the size of the family on which the Isolating Lemma operates, we obtain significant savings in the number of random bits required. The randomness complexities of our algorithms for matroids, when specialized to the case of matching, yield precisely our results on matching. In particular, we give NC^2 algorithms for these problems on matroids under certain restrictions which in the case of graph matching correspond exactly to a polynomial bound on the number of perfect matchings. This generalizes the results of [11] to matroids and also generalizes a result due to Tiwari[28].

Another important application of the Isolating Lemma is to obtain an alternative to the Valiant & Vazirani random reduction [32] from SAT to $USAT$. Our generalization yields a new random reduction whose randomness complexity is logarithmic in the number of satisfying assignments. The worst case complexity matches the best–known bound of $O(n)$ (Tarui [26]) and also directly gives new proofs of the results that $FewP$, where the number of satisfying assignments is polynomially bounded, is contained in $\oplus P$ (Cai & Hemachandra [3]) and in $C_{=}P$ (Köbler, Schöning, Toda & Toran [16]).

This paper is organized as follows: In Section 2 we describe the weight assignment scheme and also show that it has optimal randomness complexity. Section 3 lists the applications of the new tool to perfect matching and its variants. Applications to problems on linearly representable matroids and random reductions in structural complexity are listed in Section 4.

2 The Generalized Isolating Lemma

In this section we present our randomness–efficient generalization of the Isolating Lemma and also show its optimality by proving a matching randomness complexity lower bound for this problem. We start with a formal statement of the Isolating Lemma:

Definition A *set system* $(S, \mathcal{F})$ consists of a finite set $S = \{x_1, \ldots, x_N\}$ and a family $\mathcal{F}$ of subsets of S. A weight assignment $\vec{w} = \langle w_1, \ldots, w_N \rangle$ to the elements $x_1, \ldots, x_N$, extends naturally to sets in $\mathcal{F}$ with $\vec{w}(S_j) = \sum_{x_i \in S_j} w_i$.

The crux of the RNC^2 algorithm of Mulmuley, Vazirani & Vazirani is the following probabilistic tool:

Lemma 1 [**Isolating Lemma**] *[21]* Let $(S, \mathcal{F})$ be any set system. If the elements x_i of S are assigned random weights *uniformly and independently* from $\{1, 2, \ldots, 2N\}$ then, with probability at least $1/2$, there will be a *unique* minimum weight set in $\mathcal{F}$.

Two features make the isolating lemma widely applicable. First, the isolation process works for arbitrary and unknown families $\mathcal{F}$ and second, isolation is achieved by assigning only polynomial weights. For example, in the bipartite perfect matching problem where S is the set of edges and $\mathcal{F}$ is the collection of perfect matchings, the first feature allows isolation to be done in RNC without any knowledge of the perfect matchings and the second ensures that a matching can be found by inverting a matrix with *polynomial* sized entries [21].

2.1 New Isolation Scheme

We prove a randomness–efficient generalization of the Isolating Lemma which also assigns polynomial weights and which depends only on any given upper bound Z on the size of the *unknown* family $\mathcal{F}$. The motivation for this generalization is that in several settings where the Isolating Lemma has been used to obtain randomized algorithms in the general case, we have deterministic solutions when the number of solutions is small[11, 28, 3]. These deterministic solutions have been obtained by techniques that are specialized to each case and our aim is to obtain them by carefully parametrizing the randomness complexity of the algorithms for the general case. To do this we prove the following generalization:

Lemma 2 **[Generalized Isolating Lemma]** *Let* $(S,\ \mathcal{F})$ *be any set system and let* Z *be a* given *upper bound on the size of the unknown family* $\mathcal{F}$*. There is a simple scheme which uses* $O(\log Z + \log N)$ *random bits to assign integer weights to the* x_i*'s in the range* $[0, N^7)$ *such that with probability at least* $1/4$*, there is a* unique *minimum weight set in* $\mathcal{F}$*.*

PROOF. We outline a four-step process for assigning weights to the x_i's and prove that this scheme has the desired properties. At step j, we assign an intermediate weight $\vec{w}^{(j)}$. Only steps 2 and 4 are randomized and these steps together use $O(\log Z + \log n)$ random bits.

Since $\mathcal{F}$ is unknown we deterministically assign very large weights in Step 1 so that every set in $\mathcal{F}$ gets a *distinct* weight.

Step 1: For each i, set $w_i^{(1)} = 2^i$.

Under $\vec{w}^{(1)}$, sets in $\mathcal{F}$ have *distinct* weights in $\{1, 2, \ldots, 2^{N+1}\}$. Clearly, if $Z \ll 2^{N+1}$, the same property should be obtainable with much lower weights. Since $\mathcal{F}$ is unknown, a deterministic strategy for reducing weights may fail; instead, we use a randomized strategy.

Step 2: Choose m uniformly at random from $\{1, 2, \ldots, (2NZ^2)^2\}$. For each i, define $w_i^{(2)} = w_i^{(1)} \bmod m$.

Step 2 requires $O(\log Z + \log N)$ random bits. Under $\vec{w}^{(2)}$, the Z or fewer sets in $\mathcal{F}$ have weights in the interval $[0, \min(N(2NZ^2)^2, 2^{N+1})]$, which is a big improvement for small values of Z. We now claim that with good probability, these weights are also *distinct.*

Claim 1 *With probability at least* $1/2$*,* all *sets in* $\mathcal{F}$ *have* distinct *weights under* $\vec{w}^{(2)}$*.*

PROOF. Suppose $\mathcal{F} = \{S_1, \ldots, S_k\}$ where $k \le Z$. Consider the (unknown) integer

$$I = \prod_{1 \le i < j \le k} (\ \vec{w}^{(1)}(S_i) - \vec{w}^{(1)}(S_j)\ ).$$

Clearly the properties of $\vec{w}^{(1)}$ ensure that $I \neq 0$ and that $|I| < 2^{2NZ^2}$. The following number-theoretic proposition, together with the Chinese Remainder Theorem establishes that when m is chosen uniformly from $\{1, 2, \ldots, (2NZ^2)^2\}$, the probability that $m \not| I$ is at least 1/2. Several versions of this proposition appear in the literature; this version is from [30] and the constant 89 has been improved to 3 by Thrash [27].

Proposition 1 *Let $L \geq 89$ and let S be any subset of $\{1, \ldots, L^2\}$ such that $|S| \geq \frac{1}{2}L^2$. Then, the least common multiple of the elements of S exceeds 2^L.*

We claim that if $m \not| I$, then all sets in $\mathcal{F}$ have *distinct* weights under $\vec{w}^{(2)}$. If not, then $\vec{w}^{(2)}(S_i) = \vec{w}^{(2)}(S_j)$ for some $i < j$. Then we have $m \mid (\vec{w}^{(1)}(S_i) - \vec{w}^{(1)}(S_j))$, which contradicts the fact that $m \not| I$. □

Remark: We say that Step 2 *succeeds* if sets in $\mathcal{F}$ get distinct weights under $\vec{w}^{(2)}$. The success probability of this step can be boosted to any constant lesser than 1, since it follows from the results of de Bruijn [8] that for any fixed $p < 1$, there exist constants c_1 and c_2 such that if m is picked uniformly at random from $\{1, 2, \ldots, N^{c_1}Z^{c_2}\}$ in Step 2, then Step 2 will succeed with probability at least p.

Note that if $Z \leq N^c$, then $w_i^{(2)} \leq N^{4c+3}$ and with probability at least 1/2, *all* sets in $\mathcal{F}$ have *distinct* weights. Later, we use this strong condition to design an NC^2 algorithm to find *all* perfect matchings in graphs with at most $N^{O(1)}$ perfect matchings. For larger Z, the weights are still too big and in Steps 3 and 4 we reduce them further. Step 2 assigns q-bit weights to the x_i's where $q = min(N, \log m) \leq min(N, 4\log Z + 2\log(2N))$. Let $t = \lceil q/\log N \rceil$.

Step 3: For each i, write $w_i^{(2)}$ as q-bit number. Split these bits into t blocks of size $\log N$ bits each as shown. Let $b_{i,j}$ be the number in $[0, N-1]$ formed by the bits in block j.

$$w_i^{(2)} : \quad \overbrace{\boxed{b_{i,t-1}}\boxed{\;\cdots\;}\boxed{b_{i,1}}\underbrace{\boxed{b_{i,0}}}_{\log N}}^{q}$$

Let $w_i^{(3)}$ be the *linear form* $\sum_{j=0}^{t-1} b_{i,j} \cdot y_j$ over the variables $y_0, \ldots, y_{t-1}$.

Claim 2 *If Step 2 succeeds then the linear forms $\vec{w}^{(3)}(S_j)$, where $S_j \in \mathcal{F}$, are all* distinct.

Proof: Assume that Step 2 succeeds, *i.e.*, that all the weights $\vec{w}^{(2)}(S_j)$ are *distinct*, where $S_j \in \mathcal{F}$. Note that each $w_i^{(3)}$ *evaluated* at $y_k = 2^{k \log N}$, $0 \leq k \leq t-1$, is *exactly* $w_i^{(2)}$. This implies that each $\vec{w}^{(3)}(S_j)$ evaluates to the *distinct* value $\vec{w}^{(2)}(S_j)$ at $y_k = 2^{k \log N}$, $0 \leq k \leq t-1$, which implies that the forms $\vec{w}^{(3)}(S_j)$'s must be distinct. ∎

Note that each $\vec{w}^{(3)}(S_j)$ is a linear forms with coefficients in the range $[0, N(N-1)]$. We will use this property in a crucial way in the analysis of the next and final step.

Step 4: Choose $r_0, \ldots, r_{t-1}$ uniformly and independently at random from $\{1, 2, \ldots, N^5\}$. For each i, set $w_i^{(4)}$ to be the evaluation of $w_i^{(3)}$ at $y_k = r_k$, $0 \le k \le t-1$.

We claim that $\vec{w}^{(4)}$ achieves the requirements of the Generalized Isolating Lemma. Clearly, Step 4 requires $5 \log N \times t = O(\log Z + \log N)$ random bits and since Step 2 has a similar randomness complexity, the overall procedure requires $O(\log Z + \log N)$ random bits. It is easy to check that each $w_i^{(4)}$ is in $[0, N^7)$. Since Step 2 succeeds with probability at least 1/2, by using $\mathcal{C} = \{\vec{w}^{(3)}(S_j) \mid S_j \in \mathcal{F}\}$ in the following proposition, we obtain that the weight assignment $\vec{w}^{(4)}$ achieves isolation with probability at least 1/4.

Proposition 2 *Let $\mathcal{C}$ be any collection of* distinct *linear forms over at most t variables $\vec{y} = y_0, y_1, \ldots, y_{t-1}$ with coefficients in $\{0, 1, \ldots, N^2-1\}$. Choose a random $\vec{r} = r_0, \ldots, r_{t-1}$ by assigning each r_i uniformly and independently from $\{1, 2, \ldots, N^5\}$. Then in the assignment $\vec{y} = \vec{r}$ there will be a* unique *linear form with minimum value, with probability at least* 1/2.

Proof of Proposition: Our proof parallels that of [21]; we define a variable y_i to be *singular* under an assignment $\vec{r}$ to the variables $\vec{y}$ if there exist two minimum valued linear forms in $\mathcal{C}$ under this assignment, having different coefficients of y_i. Since all the linear forms in $\mathcal{C}$ are initially distinct, if two linear forms attain the minimum weight then there *must* exist some variable which is singular under this assignment.

For each y_i, we will upper bound the probability that it is *singular*. Fix an i and assume that the variables $\vec{r}(\bar{i}) = r_0, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{l-1}$ have been assigned the values $\vec{a}(\bar{i}) = a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{N-1}$. Under this partial assignment the weight of each linear form is of the form $a + by_i$ where a is the *partial weight* of the linear form and the coefficient b is at most N^2-1. The family $\mathcal{C}$ can be partitioned into at most N^2 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{N^2-1}$ based on the coefficient of y_i, *i.e.*, forms in $\mathcal{C}_j$ have j as the coefficient of y_i. By definition, y_i is *singular* under an assignment $\vec{r}$ if and only if there are two linear forms from two different classes in the partition which attain the minimum weight. Consider the probability that the choice of r_i makes y_i singular, conditioned on the partial assignment to the other variables. Let p_j be the minimum partial weight among all the forms in $\mathcal{C}_j$. Note that since all the forms in a particular class have the same coefficient of y_j in their weights, only the forms with minimum partial weight in each class can possibly attain the minimum overall weight. Thus the probability that r_i makes y_i *singular* is exactly the probability that the minimum valued element in the list $\mathcal{D} = [p_0, p_1 + r_i, p_2 + 2r_i, \ldots, p_{N^2-1} + (N^2-1)r_i]$ is not unique. This is clearly bounded by the probability that under a random r_i the values of elements in $\mathcal{D}$ are not all distinct. For each pair of elements in $\mathcal{D}$, *at most one* choice of r_i makes their values equal. Thus

$$\text{Prob.}[\ y_i \text{ is singular} \mid \vec{r}(\bar{i}) = \vec{a}(\bar{i})\ ] \ \le\ \text{Prob.}[\ \exists l, m:\ p_l + l \times r_i = p_m + m \times r_i\ ]$$

$$\leq \binom{N^2}{2} \times \text{Prob.}[\, p_l + l \times r_i = p_m + m \times r_i \,]$$
$$\leq \binom{N^2}{2} \times \frac{1}{N^5} = \frac{1}{2N}$$

Since r_i is chosen independently of the other variables $\vec{r}(\bar{i})$ the probability that the assignment makes y_i singular is at most $\frac{1}{2N}$. Since this holds for each variable, and there are at most $t < N$ variables we have

$$\text{Prob.}[\text{ minimum valued form in } C \text{ is unique }] \geq 1 - \text{Prob.}[\,\exists j \;\; y_j \text{ is singular}]$$
$$\geq 1 - t \times \frac{1}{2N} > \frac{1}{2}$$

□

Since each of the four steps is elementary the entire weight assignment scheme can be implemented in NC^1. Finally, we note that the Generalized Isolating Lemma provides an *explicit construction* of $(NZ)^{O(1)}$ many weight assignment functions for the ground set $S = \{x_1, x_2, \ldots, x_N\}$, such that for any set system $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$, at least half of these functions will produce a unique minimum weight set in $\mathcal{F}$. For a given ground set S, the number of set systems $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$ is $\sum_{i=1}^{Z} \binom{2^N}{i}$; hence, the technique of Adleman [1] combined with the original isolating lemma shows *nonconstructively* that there exists a set of $O(\log(\sum_{i=1}^{Z} \binom{2^N}{i})) = O(NZ)$ many weight assignment functions such that for any set system $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$, at least *one* of these functions will produce a unique minimum weight set in $\mathcal{F}$.

2.2 A Lower bound for the Isolation problem

We establish a matching lower bound of $\Omega(\log Z + \log N)$ random bits on the randomness complexity of the generalized isolation process, and in fact for the following weaker problem, thus showing that the randomness complexity of our weight assignment scheme is optimal to within a constant factor.

Generalized Isolation Let $S = \{x_1, \ldots, x_N\}$ and let k be a constant. Assign weights to the elements of S in the range $[1, N^k]$ such that for any family $\mathcal{F} \subseteq 2^S$ of size at most Z, the minimum weight set in $\mathcal{F}$ is unique with *nonzero* probability.

First we prove a lower bound of $\Omega(\log Z)$ random bits for any randomized scheme for the above problem. Each path of such a randomized algorithm defines a weight assignment f to the x_i's. The following theorem shows that at least $t+1$ different weight assignments are needed in order to achieve isolation in all possible families of size at most Z, where $t = \min(\lfloor \frac{Z}{2} \rfloor, \lfloor \frac{2N-3}{N^{k+1}} \rfloor)$. Hence any randomized scheme for generalized isolation must use $\Omega(\log t) = \Omega(\log Z)$ random bits.

Theorem 1 *Let $f_1, f_2, \ldots, f_t$ be any collection of weight assignments which assign weights in the range $[1, B]$ to the elements of S. If $t \leq \min(\frac{Z}{2}, \frac{2N-3}{NB})$, then there exists a family $\mathcal{F}$*

of subsets of S with $|\mathcal{F}| \leq 2t \leq Z$, such that the minimum weight subset of $\mathcal{F}$ is not unique under any of these assignments.

PROOF. Given any t weight assignments we explicitly construct the family $\mathcal{F}$ as follows. First, for each f_i, we compute the "histogram" of the weights of the subsets of S, *i.e.*, for each nonempty subset X we place a mark above the weight $f_i(X)$ in the histogram for f_i; one distinct mark is made for each such subset X.

The weight of any subset is in the range $[1, NB]$. Initialize, for each i, a pointer p_i to 1 in the histogram for f_i. The pointers p_i advance according to the following rules:

- If there are no marks on weight p_i in the histogram of f_i, then advance p_i to $p_i + 1$.
- If there is exactly one mark on weight p_i which corresponds to some subset X, remove the marks corresponding to X from *all* the histograms and advance p_i to $p_i + 1$.

If more than one pointer can advance, we choose one among them arbitrarily. We continue this process until the pointers cannot move any further. First we argue that not all marks can be removed since every time a mark is removed, one of the pointers advances and hence the potential function $\Psi = \sum_i (p_i - 1)$ always increases. Since each p_i is bounded by $NB + 1$, Ψ is bounded by tNB. By assumption $tNB \leq 2^N - 3$, and hence when the pointers come to a halt there are at least two marks corresponding to nonempty subsets X_{i1} and X_{i2} on weight p_i in the histogram of f_i, for each i. The family $\mathcal{F} = \{X_{i1}, X_{i2} | \ 1 \leq i \leq t\}$ has size at most $2t \leq Z$ and for each i, there are two nonempty minimum weight subsets, X_{i1} and X_{i2}, of minimum weight p_i, under the assignment f_i. □

Also when $B = n^{O(1)}$ and $t = o(N/\log N)$, we can construct two sets which have the same weight under all the assignments $f_1, f_2, \ldots, f_t$, as follows. Since f_1 maps the nonempty subsets of S to the range $[1, NB]$, there exists a family A_1 of nonempty subsets of S, all of which get the same weight under f_1 and such that $|A_1| \geq \frac{2^N-1}{NB}$. Similarly, there exists a family $A_2 \subseteq A_1$ with $|A_2| \geq \frac{2^N-1}{(NB)^2}$, such that $f_i(X_1) = f_i(X_2)\ \forall X_1, X_2 \in A_2$, for $i = 1, 2$. Repeating this, we end up with a set A_t of nonempty subsets of S with $|A_t| \geq \frac{2^N-1}{(NB)^t}$, such that the minimum weight element of A_t is not unique under none of $f_1, f_2, \ldots, f_t$; since $\frac{2^N-1}{(NB)^t} \geq 2$ for $t = o(N/\log N)$, our claim holds. Thus, any scheme for Generalized Isolation requires $\Omega(\log Z + \log N)$ random bits. Theorem 1 also gives similar lower bounds when the elements are assigned superpolynomial weights. In fact, we can prove the same lower bound when the assignments map subsets of S to an arbitrary linearly ordered set of size B since the above proofs use no property of the integers other than their total ordering.

3 Applications to Matching Problems

The Generalized Isolating Lemma immediately yields randomness-efficient algorithms for several problems related to matching. In this section, we only consider matchings in bipartite graphs; with more work, identical results can be obtained for matchings in general graphs [19, 21]. The failure probabilities of all our algorithms can be made inverse polynomial with only a constant blowup in the time and randomness complexities and a polynomial blowup in the number of processors using two point sampling techniques of Chor & Goldreich [5]. The new isolating lemma is a general tool to bound the randomness complexity of a problem in terms of the number of its solutions. However, due to the large polynomial weights used by our lemma, algorithms based directly on the lemma usually suffer a processor penalty.

Notation We denote a bipartite graph G by $G = (U, V, E)$, where $U = \{u_1, u_2, \ldots, u_{n/2}\}$, $V = \{v_1, v_2, \ldots, v_{n/2}\}$, and $|E| = m$. The determinant of a matrix A is denoted by $det(A)$. The number of perfect matchings in G is denoted by $|\mathrm{Mat}(G)|$.

3.1 Algorithms for matching

To apply unique element isolation to perfect matching, we let the ground set be the set of edges and let the family of subsets be the set of perfect matchings of G, as in [21]. By the Generalized Isolating Lemma, given an upper bound Z on $|\mathrm{Mat}(G)|$, we can assign polynomially bounded weights to the edges of G using $O(\log Z + \log n)$ random bits such that there is a unique perfect matching of minimum weight, with good probability. We construct a matrix $M[1..n/2, 1..n/2]$ with

$$M[i,j] = \begin{cases} 2^{w_{ij}} & \text{if } (u_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

where w_{ij} is the weight assigned to edge (u_i, v_j). A perfect matching in G can now be found, if one exists, by inverting M via the NC^2 algorithm of Csanky [6], as shown in [21]. Note further that the additive $O(\log n)$ factor in the randomness complexity can be absorbed by a processor penalty. For any graph $G = (V, E)$, $|\mathrm{Mat}(G)| \leq \prod_{v \in V} degree(v)$, which is at most $(2m/n)^n$ since the sum of the degrees of the vertices is $2m$. In the worst case, by letting $Z = (2m/n)^n$ we obtain a randomness complexity of $O(n \log(m/n))$ which improves the previous bound of $O(m \log n)$ [21]. Thus, we get

Theorem 2 *There is an RNC^2 algorithm to find a perfect matching in a graph G (if one exists) which uses $O(\log Z + \log n)$ random bits, given an upper bound Z on $|\mathrm{Mat}(G)|$. In the worst case, this algorithm uses $O(n \log(m/n))$ random bits.*

A careful analysis of the proof of our isolating scheme when applied to the worst case where $Z = (2m/n)^n$ and $m = \Theta(n^2)$, shows that the following algorithm also achieves the same randomness complexity: the edges of each vertex $u \in U$ are assigned weights *pairwise independently* in the range $[1, m^7]$.

Perfect matching is a special case of the *Exact Matching Problem*: given a graph G with edges colored red and blue arbitrarily and an integer k, an exact matching is a perfect matching of G with *exactly* k red edges (Papadimitriou & Yannakakis [25]). Even though the problem of finding an exact matching is not known to be in P, the Isolating Lemma has been used to solve it in RNC^2 [21]! By observing that isolation is needed only among the exact matchings, we can link the randomness complexity of this problem to a bound X on the number of exact matchings and *not* on the number of perfect matchings. Suppose we assign a nonnegative integral weight w_{ij} to every edge $(u_i, v_j) \in E$. Let y be an indeterminate; construct a symbolic matrix $M_y[1..n/2, 1..n/2]$ such that

$$M_y[i,j] = \begin{cases} y \cdot w_{i,j} & \text{if } (u_i, v_j) \in E \text{ and is colored red,} \\ w_{ij} & \text{if } (u_i, v_j) \in E \text{ and is colored blue,} \\ 0 & \text{otherwise} \end{cases}$$

Then, it is easy to see that the contribution to the coefficient of y^k in $det(M_y)$ comes precisely from the exact matchings in G and that if there is a unique minimum weight exact matching M^*, then its weight equals the highest power of 2, say p^*, which divides this coefficient of y^k. Furthermore, any edge can be tested for membership in M^* by increasing its weight by 1 and testing if the new value of p^* is higher than its old value or not. Hence, this problem can be solved by using our Generalized Isolating Lemma over the ground set E and the (unknown) family of exact matchings of G. Since the determinant of a matrix in one variable can be computed in NC^2 (Borodin, Cook & Pippenger [2]), we get

Theorem 3 *Given an upper bound X on the number of exact matchings in a graph G, an exact matching (if any) in G can be found in RNC^2 using $O(\log X)$ random bits.*

The Isolating Lemma has also been used to find the matching of largest cardinality in bipartite graphs by a reduction to the perfect matching problem. However the NC reductions that have been used earlier [21, 19] do not suffice to get randomness–efficient solutions since they cause a blow up in the number of solutions. Instead, we use the following reduction from maximum matching to exact matching: Given a graph $G = (V, E)$ we first construct 2 copies of G and color the edges in each of the copies red. Each vertex v in the first copy of G is joined to vertex v in the second copy by a blue edge, to yield the resulting graph, G_1. If m is the size of the largest matching in G then it can easily be seen that there is an exact matching with $2m$ red edges in G_1, but no exact matching with more than $2m$ red edges. Also, if Z is an upper bound on the number of maximum matchings in the graph G the new graph has at most Z^2 exact matchings with $2m$ red edges. Also note that if G is bipartite then so is G_1. Thus, for maximum matching, we first construct the new graph G' and the new bound Z^2, and run the above algorithm for exact matching, with each possible value of m simultaneously in parallel using the same random bits. We output the matching of largest cardinality that results in any of the parallel runs.

Theorem 4 *There exists an RNC^2 algorithm for maximum matching which uses $O(\log Z + \log n)$ random bits, where Z is a known bound on the number of maximum matchings in G.*

3.2 Graphs with a polynomial number of perfect matchings

Grigoriev & Karpinski [11] consider the problem of finding all the perfect matchings in graphs with a polynomial bound on the number of perfect matchings. Using techniques from algebra, they give an NC^2 algorithm for this problem if the number of matchings is bounded by $\log^{0.5-\epsilon} n$, and an NC^3 algorithm when the bound is n^c for constant c. Since their algorithm to extract a matching is limited by their procedure to detect if a matching exists they suggest that $\log^{0.5-\epsilon} n$ may be a limiting upper bound on the number of matchings to find all (or even one) of the perfect matchings in NC^2 via parallel algebra. We present a NC^2 algorithm for the detection problem which provides enough information to find all matchings in NC^2 even when the number of matchings is polynomially bounded. Besides the improvement in running time, our algorithm is considerably simpler than the algorithm of Grigoriev & Karpinski.

As before, we let S and $\mathcal{F}$ be E and Mat(G) respectively. Running steps 1 and 2 of the generalized Isolating scheme, we use $O(\log n)$ random bits to assign each edge a polynomially bounded weight such that *all* the perfect matchings of G have *distinct* weights with probability at least 1/2. Let $w_{i,j}$ be the weight assigned to edge (u_i, v_j). Construct a matrix A with

$$A[i,j] = \begin{cases} 2^{2\times w_{ij}} & \text{if } (u_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Since every matching has a distinct weight, $det(A)$ gives information about *every* perfect matching as shown below.

Assume that step 2 succeeds *i.e.* all the perfect matchings indeed get distinct weights. If we denote the matrix obtained from M by removing its ith row and jth column by M_{ij}, then by definition

$$(M^{-1})[j,i] = \frac{(-1)^{i+j} det(M_{ij})}{det(M)}.$$

Now, if there is a perfect matching of weight k containing the edge (u_i, v_j), then $det(M_{ij})$ is of the form

$$\pm 2^{2\times(k-w_{ij})} + \sum_{\ell=0}^{k-1} a_\ell 2^{2\times(\ell-w_{ij})} + \sum_{\ell\geq k+1} a_\ell 2^{2\times(\ell-w_{ij})},$$

where each a_ℓ is in $\{-1, 0, 1\}$. Since

$$|\sum_{\ell=0}^{k-1} a_\ell 2^{2\times(\ell-w_{ij})}| < 2^{2\times(k-w_{ij})-1},$$

this is of the form

$$\pm 2^{2\times(k-w_{ij})} \pm n_0 \times 2^{2(k-w_{ij}+1)} \pm x \times 2^{2(k-w_{ij})-1},$$

where n_0 is a nonnegative integer and $0 \leq x < 1$. Similarly, if there is no perfect matching of weight k containing the edge (u_i, v_j), then $det(M_{ij})$ is of the form

$$\pm n_0 \times 2^{2(k-w_{ij}+1)} \pm x \times 2^{2(k-w_{ij})-1}.$$

Hence, there is a perfect matching of weight k containing the edge (u_i, v_j) if and only if

$$|(M^{-1})[j,i] \cdot det(M) \cdot 2^{2w_{ij}}| \pmod{2^{2(k-1)}}$$

is of the form $4n_1 \pm y$, where n_1 is an integer and $y = 1$ or 2. If there is no matching of weight k then $y = 0$ or 3. Since k is polynomially bounded we can extract all matchings. Also, since we use $O(\log Z + \log n) = O(\log n)$ random bits, we can derandomize this by searching over all points in the sample space in parallel to get

Theorem 5 *There is an NC^2 algorithm to detect if a perfect matching exists and to find all perfect matchings in a graph G, when given that $|\mathrm{Mat}(G)| \leq n^c$ for some constant c.*

We can obtain the same result for exact matching and maximum matching problems given a polynomial bound on the number of exact matchings and maximum matchings respectively.

Grigoriev & Karpinski also give a Las Vegas RNC^3 algorithm for the problem of testing if a graph has at most n^c perfect matchings, which uses expected $O(n^{2c+8.5}\log n)$ random bits. We can, using the algorithm of Theorem 5 and by extending ideas of [17, 13], improve on the running time and substantially on the number of random bits used, as follows: Karloff[13] presents an NC^2 algorithm which calls a perfect matching algorithm (oracle) on $n+1$ graphs (each with $n-1$ or n vertices and at most m edges) in parallel, and which, assuming that these calls worked correctly, will announce *correctly* if G has a perfect matching. or not; in any case, if it ever says that $|\mathrm{Mat}(G)| = 0$, then the number of perfect matchings is indeed zero. Actually, Karloff's result is more general – it works for maximum matchings. Note that in conjunction with any Monte Carlo RNC^2 algorithm to find a perfect matching (in particular, ours) this yields a Las Vegas RNC^2 algorithm to decide if G has a perfect matching or not, and if so, to produce one. We first make the error probability of our perfect matching algorithm at most $\frac{1}{2(n+1)}$ (as mentioned in Section 2, this can be done in RNC^2) and using only $O(n\log(m/n))$ random bits then invoke Karloff's algorithm, which will make calls in parallel to our algorithm, using the *same sequence of random bits* for each call; the probability that at least one of these calls produced a wrong result is at most $\frac{n+1}{2(n+1)} = 1/2$. Hence, we can decide if $|\mathrm{Mat}(G)| > 0$ or not *correctly*, using an expected $O(n\log(m/n))$ many random bits. The above probability can be made any inverse polynomial with a processor penalty using the techniques of two–point sampling[5] If we conclude that $|\mathrm{Mat}(G)| > 0$, then we run the algorithm of Section 3.2 to generate all the perfect matchings of G, assuming that $|\mathrm{Mat}(G)| \leq n^c + 1$. Kozen, Vazirani and Vazirani[17] give a simple NC^2 algorithm to test given a perfect matching whether there are other perfect matchings in the graph. A direct extension of the technique yields a NC^2 procedure to check given a polynomial number of perfect matchings whether there are other perfect matchings. The Las Vegas algorithm will then check if there are other perfect matchings in the graph besides the ones generated and accept if there are other matchings, otherwise it reports a failure. If G has at most n^c perfect matchings, then at most n^c perfect matchings will be generated and we will pass the test for other matchings and so the algorithm accepts correctly with high probability.

Theorem 6 *There exists a Las Vegas RNC^2 algorithm to test if a given graph G has $|\mathrm{Mat}(G)| \leq n^c$ whose expected randomness complexity is $O(n \log(m/n))$.*

3.3 Randomness-Efficient algorithms with no information on $|\mathrm{Mat}(G)|$

The algorithms of Theorem 2 can be used to obtain randomness–efficient algorithms for perfect matching even when no upper bound on the number of perfect matchings is given, by using the worst case upper bound of $Z = (2m/n)^n$. However, the following algorithm gives a much better randomness complexity:

> **for** $i := 0$ **to** $\lceil \log_2 \log_2((2m/n)^n) \rceil$ **do**
> Let $Z = 2^{2^i}$ and run the algorithm of Section 3.1 to find a perfect matching;
> **exit** if a perfect matching is found.

If there is at least one perfect matching, then this algorithm will find a perfect matching with good probability when $2^{2^i} \geq |\mathrm{Mat}(G)|$. If M is the number of perfect matchings then the number of random bits used and the time taken will be $O(\log M)$ and $O(\log^2 n \log\log M)$ respectively with good probability, and so we get

Theorem 7 *Let M=$|\mathrm{Mat}(G)|$. There is an RNC algorithm to find a perfect matching in G with no given upper bound on M, which uses $O(\log M)$ random bits and runs in time $O(\log^2 n \log\log M) = O(\log^3 n)$ with high probability, if $M > 0$; in the* worst case, *it uses $O(n \log(m/n))$ random bits and $O(\log^3 n)$ time.*

4 Other applications

The Isolating Lemma is a very abstract and powerful tool to make one solution from a possibly exponential sized collection stand out. Due to its general nature it has several applications. Thus, our generalized Isolating Lemma leads to randomness–efficient solutions to various problems, which we describe in this section.

4.1 Randomized reductions from SAT to $USAT$

An important result in complexity theory is the random reduction, due to Valiant & Vazirani, from languages in NP to $USAT$, the language of uniquely satisfiable Boolean formulae [32]. This reduction is at the core of several fundamental results in complexity theory, most notably in the results of Toda on the unexpected power of counting classes [29]. The Isolating Lemma has been used to derive an alternative random reduction in [21]. We apply the Isolating Lemma to get a slightly different reduction and here, our generalization yields better randomness complexity.

Given a formula $F(x_1, x_2, \ldots, x_n)$ as input, to use the isolating lemma we let the ground set S be the set of variables $\{x_1, x_2, \ldots, x_n\}$ and let the family $\mathcal{F}$ be the satisfying assignments of F, where a satisfying assignment is represented by the subset of variables that are true in it. Assign weights $w_1, \ldots, w_n$ to the ground set elements as prescribed by the generalized isolating lemma and choose a random element y in the range [1,n^8]. Consider the following reduction:

On input $F(x_1, \ldots, x_n), w_1, \ldots, w_n, y$ we construct an NP machine M which works as follows: Guess an assignment for the variables $x_1, \ldots, x_n$. If the assignment satisfies the formula F and the weight of the assignment under the weight assignment $w_1, \ldots, w_n$ is equal to y then we accept. Notice that since the weight assignment scheme succeeds with probability $\frac{1}{2}$, if F is satisfiable then with probability $\frac{1}{2}$ there is one minimum weight satisfying assignment. Since all the assignments have a weight in the range $[1, n^8]$ and y is chosen uniformly at random from $[1, n^8]$ we have

$$F \text{ is satsfiable} \Longrightarrow \text{Prob}[\ M \text{ accepts on exactly one path }] \geq \frac{1}{2 \times n^8}$$

If F is not satisfiable then the machine M clearly accepts on no paths. Given this machine M we can construct using the Cook–Levin reduction a boolean formula F' which has exactly the same number of satisfying assignments as the number of accepting paths of M. So the random reduction first assigns weights to the variables, chooses a random weight y and then constructs the formula F' described above. By the description above it is easy to see that

$$F \text{ is satisfiable } \Longrightarrow \text{Prob}[\ F' \text{ is uniquely satisfiable }] \geq \frac{1}{2 \times n^8}$$

$$F \text{ is not satisfiable } \Longrightarrow \text{Prob}[\ F' \text{ is not satisfiable }] = 1$$

This reduction requires $O(\log Z + \log n)$ random bits, where Z an upper bound on the number of satisfying assignments of F. In the worst case this randomness complexity is $O(n)$, which improves on the $O(n \log n)$ bound of [21], the $O(n^2)$ bound of [32] and matches the result of Tarui [26]. Thus we obtain:

Theorem 8 *There is a randomized reduction from SAT to $USAT$ which uses $O(\log Z + \log n)$ random bits, where n and Z are the number of variables and an upper bound on the number of satisfying assignments respectively.*

If the number of satisfying assignments of F is polynomially bounded, as in case of languages in the class $FewP$, the reduction uses $O(\log n)$ random bits, This can be derandomized to yield new proofs of the results $FewP \subseteq \oplus P$ [3] and $FewP \subseteq C_{=}P$ [16] as described below.

Definition A language L is in the class $FewP$ if there is a nondeterministic polynomial time machine M which accepts L and a polynomial q such that for all strings x, $M(x)$ accepts on at most $q(|x|)$ paths.

Note that for a language L in $FewP$, given a string x, the Cook–Levin reduction produces a formula F with at most a polynomial number of satisfying assignments such that x is in L iff F is satisfiable.

Definition A language L is in the class $\oplus$P if there is a nondeterministic polynomial time machine M such that x is in L iff M on input x accepts on an odd number of paths.

The class was first defined by Papadimitriou and Zachos[24] who also show that the class is closed under many operation: in fact, $P^{\oplus P} = \oplus P$. When we derandomize the above reduction for languages in $FewP$, we can get a polynomial number of formulas such that F is satisfiable iff one of these formulas is uniquely satisfiable. Also, if F is unsatisfiable then none of these formulas is satisfiable. Since the condition in the antecedents can easily be checked in $P^{\oplus P}$ we can get a new proof of the result that $FewP \subseteq \oplus P$.

Definition A language L is in the class $C_{=}P$ iff there is a nondeterministic polynomial time machine M and a polynomial time computable function f such that x is in L iff M on input x accepts on exactly $f(x)$ paths.

We can use $C_{=}P$ to test if one of a polynomial number of formulas has exactly one satisfying assignments as follows: Let $F_1, \ldots, F_t$ be formulas on n variables each, with $s_1, \ldots, s_t$ satisfying assignments respectively. If we could construct a formula F' with $(s_1-1)\cdots(s_t-1)$ satisfying assignments then F' would have exactly 0 satisfying assignments iff one of the F_js has exactly one satisfying assignment. However doing this directly involves constructing formulas with one less satisfying assignment than the F_js which is highly intractable[23]. We do this slightly indirectly as follows: Each term in the product $(s_1 - 1)\cdots(s_t - 1)$ is of the form $\pm s_{i_1}s_{i_2}\cdots s_{i_r}$ where $r \leq t$. We can easily construct a formula which has $s_{i_1}s_{i_2}\cdots s_{i_r}$ satisfying assignments and so if the coefficient of this term is $+1$ then we are done. However if the coefficient is -1 then we construct a formula with $2^{p(n)} - s_{i_1}s_{i_2}\cdots s_{i_r}$ satisfying assignments, where $p(n) = t \times n$. A nondeterministic polynomial time machine M which accomplishes this works as follows: First, M nondeterministically chooses a term in the product $(s_1 - 1)\cdots(s_t - 1)$. If the term is of the form $s_{i_1}s_{i_2}\cdots s_{i_r}$ it guesses r assignments and accepts iff the assignments satisfy $F_{i_1},\ldots,F_{i_r}$ respectively. If the term is of the form $-1 \times s_{i_1}s_{i_2}\cdots s_{i_r}$, it guesses t assignments and accepts iff the first r formulas do not satisfy $F_{i_1},\ldots,F_{i_r}$ or the last $t-r$ assignments are all zeros. It can be easily verified that if the term has coefficient 1 then M accepts on $s_{i_1}s_{i_2}\cdots s_{i_r}$ paths and if the coefficient is -1 then M accepts on $2^{p(n)} - s_{i_1}s_{i_2}\cdots s_{i_r}$ paths. Since the number of terms with coefficient -1 is 2^{t-1}, M accepts on $(s_1-1)\cdots(s_t-1) + 2^{p(n)} \times 2^{t-1}$ paths. Using the Cook–Levin reduction one can construct a formula G with exactly the same number of satisfying assignments. Now G has exactly $2^{p(n)} \times 2^{t-1}$ satisfying assignments iff one of the original formulas has exactly one satisfying assignments. Thus the reduction also gives a new proof of $FewP \subseteq C_{=}P$.

Here again we wish to emphasize that though there are many proofs of the above two results, our proofs follow directly from bounding the randomness complexity of the general randomized reduction.

4.2 Improved parallel randomness complexity for problems on matroids

The Isolating Lemma has been used to obtain RNC algorithms for basic problems on linearly representable matroids such as matroid intersection and matching (Narayanan, Saran & Vazirani [22]). These problems are generalizations of bipartite and general graph matching.

Definition A set system $M = (S, \mathcal{F})$ is a *matroid* if:

- $\phi \in \mathcal{F}$,
- if $A \subseteq B \subseteq S$ and $B \in \mathcal{F}$, then $A \in \mathcal{F}$, and
- if $A_1 \in \mathcal{F}$ and $A_2 \in \mathcal{F}$ with $|A_2| = |A_1| + 1$, then there exists $x \in A_2 - A_1$ such that $A_1 \cup \{x\} \in \mathcal{F}$.

Every element of $\mathcal{F}$ is called an *independent set.* A basic consequence of the matroid axioms is that every maximal independent set is also a maximum independent set, and the cardinality of any maximum independent set of M is also called the *rank* of M. A matroid M of rank r over a ground set S of cardinality n is *linearly representable* over a field $\mathbf{F}$ if there exists a matrix $C \in \mathbf{F}^{r\times n}$ whose columns are indexed by the elements of S, such that a subset of S is independent in M iff the corresponding set of columns of C are linearly independent over $\mathbf{F}$. M is *linearly represented* if it is presented as the matrix B. Henceforth, the field $\mathbf{F}$ is assumed to be the field of rationals, $\mathbf{Q}$. All the results of [22] and of this section hold *only* for linearly represented matroids. For a good introduction to the theory of matroids see Welsh[34]. For other work on randomized algorithms for the matroid problem see Camerini, Galbliati and Maffioli[4].

The matroid intersection problem is to find a maximum cardinality independent set in both of two given matroids M_1 and M_2, each of rank r and over the same ground set S of cardinality n. In this case, we observe that the size of the family on which the Isolating Lemma operates, as presented in [22], is at most $I \times (\binom{r}{r-h})^2 \times (r-h)!$ where h is the size of the largest set in $M_1 \cap M_2$ and I is the number of sets of size h in $M_1 \cap M_2$. Since this can be bounded by $(nr^2)^h \times r! \leq (nr^2)^r \times r^r \leq n^{4r}$, we can use generalized isolation to reduce the randomness complexity from the previous bound of $O((n+r^2)\log n)$ [22] to $O(r \log n)$, improving by a factor of at least $\Omega(\sqrt{n})$ in all cases, and by more in general. If $h \geq r - O(\log_r n)$ and there is a polynomial (in n) bound on I then since $I(\binom{r}{r-h})^2 \times (r-h)! = n^{O(1)}$, we get the first NC algorithms. In the case of bipartite matching which is a special case of matroid intersection, $h = r$, and a polynomial bound on I corresponds exactly to a polynomial bound on the number of matchings in the input graph. Thus, this gives one way to generalize the results of [11] to matroids.

Theorem 9 *Matroid intersection for linearly represented matroids of rank r can be solved in RNC^2 using $O(r \log n)$ random bits. If the cardinality of the maximum intersection is given to be at least $r - O(\log_r n)$, and if the number of maximum cardinality intersections is given to be bounded by a given polynomial of n, then it can be solved in NC^2.*

For linearly represented vectors, the matroid matching problem is the following: given m pairs of vectors over $\mathbf{Q}^{2n}$, pick the largest number of pairs so that the vectors picked are linearly independent. For this problem, the RNC^2 algorithm of ([22], Theorem 4.3) uses the Isolating Lemma on a set family over a ground set of $m + \binom{2n}{2}$ elements and at most

$$\binom{m+\binom{2n}{2}}{n} \leq (m+2n^2)^n$$

subsets and thus by invoking the Generalized Isolating Lemma, we improve the randomness complexity from $O((m+n^2)\log(m+n^2))$ [22] to $O(n\log(m+n^2))$. We can count the size of the size of the family, to which the Isolating Lemma is applied to in, more carefully to bound the randomness complexity of the algorithm to be $O(\log(I \times \binom{2n}{2n-2h} \times (2n-2h)!))$ where h is the maximum number of pairs that can be picked so that the union is linearly independent and I is the number of ways these h pairs can be picked. As in the case of matroid intersection we can obtain deterministic algorithms if I is polynomial and if $h \geq n - \log_n m$. Tiwari, using sparse interpolation techniques, gives NC algorithms for this problem when the number of full dimensional solutions (*i.e.*, when $h = n$) is given to be polynomially bounded in n and m [28]. Thus, our result yields a generalization.

Theorem 10 *Matroid matching can be solved in RNC^2 using $O(n\log(m+n^2))$ random bits. If $h \geq n - \log_n m$, is the size of the largest number of pairs whose union is linearly independent and there is a polynomial bound on the number of ways such a set of h pairs can be picked, matroid matching can be solved in NC^2.*

5 Conclusions and open problems

The main contribution of this work is our Generalized Isolating Lemma. For various problems solvable via randomness in general [21, 22, 32] and deterministically when the number of solutions is small [17, 11, 3], this lemma bounds their randomness complexity in terms of the number of solutions. Our results span the spectrum of the number of solutions, imply, and sometimes improve the previously known results at the extremes. On the other hand, our lower bound for isolation implies that attempts to solve these problems deterministically based on isolation must also exploit their structure, rather than view them merely as abstract unknown collections of sets.

A direct open problem falling out of the Generalized Isolating Lemma is to reduce the magnitude of the weights assigned by it to the elements of the ground set; currently, they can be as high as N^7. In the context of matching, this will lead to more processor–efficient RNC algorithms. For the random reduction a decrease in the magnitude of the weights of the elements will increase the success probability of the reduction.

The lower bound for isolation that we prove holds only for schemes which work oblivious of the family of solutions that it is tring to isolate one solution from. An interesting question that arises here is: can we use some knowledge of the structure of the solution space that we are trying to isolate one solution from to reduce the randomness complexity of isolation or even perhaps obtain deterministic algorithms? In case of the perfect matching problem it would be interesting to see if we can obtain deterministic algorithms for special classes of graphs with this framework.

Acknowledgments

We would like to thank our advisors Juris Hartmanis and David Shmoys for their guidance, suggestions and support. This paper has benefited greatly from discussions with several other people too. We are particularly indebted to Joseph Cheriyan, Steve Mitchell, Moni Naor, Éva Tardos and Vijay Vazirani. A preliminary version of this paper was presented at the 25$^{\text{th}}$ Annual Symposium on the Theory of Computing. We would like to thank Martin Tompa for making valuable comments on the conference proceedings version of this paper.

References

[1] L. Adleman. Two theorems on random polynomial time. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[2] A. Borodin, S. A. Cook, and N. Pippenger. Parallel computation for well–endowed rings and space–bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[3] J-Y. Cai and L. Hemachandra. On the Power of Parity Polynomial Time. *Mathematical Systems Theory*, 23(2):95–106, 1990.

[4] P. M. Camerini, G. Galbiati, and F. Maffioli. Random pseudopolynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.

[5] B. Chor and O. Goldreich. On the power of two–point sampling. *Journal of Complexity*, 5:96–106, 1989.

[6] L. Csanky. Fast parallel matrix inversion algorithms. *SIAM J. Comput.*, 5:618–623, 1976.

[7] E. Dahlhaus and M. Karpinski. The matching problem for strongly chordal graphs is in NC. Technical Report 855–CS, University of Bonn, 1986.

[8] N. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Proc. Kon. Ned. Akad. Wet.(Indag. Math. 13)*, A54:50–60, 1951.

[9] A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, and É. Tardos. Using interior–point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.*, 21(1):140–150, 1992.

[10] A. V. Goldberg, S. A. Plotkin, and P. M. Vaidya. Sublinear–time parallel algorithms for matching and related problems. *Journal of Algorithms*, 14(2):180–213, 1993.

[11] D. Yu. Grigoriev and M. Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 166–172, 1987. See also D. Yu. Grigoriev, M. Karpinski and M. F. Singer, Fast parallel algorithms for multivariate polynomial interpolation over finite fields, *SIAM J. Comput.*, Vol. 19, 1990, 1059–1063.

[12] L. K. Grover. Fast parallel algorithms for bipartite matching. In *Proceedings of the Integer Programming and Combinatorial Optimization Conference*, pages 367–384, 1992.

[13] H. J. Karloff. A Las Vegas RNC algorithm for maximum matching. *Combinatorica*, 6(4):387–391, 1986.

[14] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in Random NC. *Combinatorica*, 6(1):35–48, 1986.

[15] P. W. Kastelyn. Graph theory and crystal physics. In F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, New York, 1967.

[16] J. Köbler, U. Schöning, S. Toda, and J. Toran. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences*, 44:272–286, 1992.

[17] D. Kozen, U. V. Vazirani, and V. V. Vazirani. NC algorithms for comparability graphs, interval graphs, and testing for unique perfect matching. In *Proceedings, FST & TCS Conference*, Lecture Notes in Computer Science # 206, pages 496–503. Springer-Verlag, Berlin, 1985.

[18] C. H. C. Little. An extension of Kastelyn's method of enumerating the 1–factors of planar graphs. In D. Holton, editor, *Combinatorial Mathematics, Proc. Second Australian Conference*, pages 63–72. Lecture Notes in Mathematics 403, Springer–Verlag, Berlin, 1974.

[19] L. Lovász and M. Plummer. *Matching Theory.* North–Holland, Amsterdam, 1986.

[20] G. L. Miller and J. Naor. Flow in planar graphs with multiple sources and sinks. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 112–117, 1989.

[21] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[22] H. Narayanan, H. Saran, and V. V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arborescences and edge–disjoint spanning trees. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, pages 357–366, 1992.

[23] M. Ogiwara and L. Hemachandra. A Complexity theory for feasible closure properties. In *Proceedings of the Sixth Annual Conference on Structure in Complexity Theory*, pages 16–27, 1991.

[24] C. Papadimitriou and S. Zachos. Two Remarks on the Power of Counting. In Proc. 6^{th} GI Conference on Theoretical Computer Science,. *Lecture Notes in Computer Science Springer-Verlag, #145*, pages 276–296, 1983.

[25] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. Assoc. Comput. Mach.*, 29(2):285–309, 1982.

[26] J. Tarui. Randomized polynomials, Threshold circuits and the Polynomial hierarchy. In *Annual Symposium on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science #480, pages 238–250, 1991.

[27] W. Thrash. A note on the least common multiples of dense sets of integers. Technical Report #93-02-04, Department of Computer Science & Engineering, University of Washington, Seattle, February 1993.

[28] P. Tiwari. Parallel algorithms for instances of linear matroid parity with small number of solutions. Technical Report RC 12766, IBM T.J.Watson Research Center, May 1987.

[29] S. Toda. PP is as hard as the Polynomial-time Hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[30] M. Tompa. Lecture notes on probabilistic algorithms and pseudorandom generators. Technical Report #91-07-05, Department of Computer Science & Engineering, University of Washington, Seattle, July 1991.

[31] P. M. Vaidya. Reducing the parallel complexity of certain linear programming problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 583–589, 1990.

[32] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85–93, 1986.

[33] V. V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Information and Computation*, 80:152–164, 1989.

[34] D. J. A. Welsh. *Matroid Theory.* Academic Press, New York, 1976.