Partitioning into Sets of Bounded Cardinality

Mikko Koivisto*

Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki,
P.O.Box 68, FI-00014 University of Helsinki, Finland
mikko.koivisto@cs.helsinki.fi

Abstract. We show that the partitions of an n-element set into k members of a given set family can be counted in time $O((2-\epsilon)^n)$, where $\epsilon > 0$ depends only on the maximum size among the members of the family. Specifically, we give a simple combinatorial algorithm that counts perfect matchings in a given graph on n vertices in time $O(\text{poly}(n)\varphi^n)$, where $\varphi = 1.618\ldots$ is the golden ratio; this improves a previous bound based on fast matrix multiplication.

1 Introduction

The generic set partitioning problem is as follows. Given an n-element universe N, a family $\mathcal{F}$ of subsets of N, and an integer k, decide whether there exists a partition of N into k members of $\mathcal{F}$, that is, pairwise disjoint sets $S_1, S_2, \ldots, S_k$ such that the union $S_1 \cup S_2 \cup \cdots \cup S_k$ equals N; we call the set $\{S_1, S_2, \ldots, S_k\}$ a *k-partition*, or simply a *partition*, and the tuple $(S_1, S_2, \ldots, S_k)$ an *ordered k-partition* or just an *ordered partition*.

Oftentimes, the family $\mathcal{F}$ is given implicitly by a description of size only polynomial in n. For example, in the graph coloring problem, $\mathcal{F}$ consists of the independent sets of a graph with vertex set N, while in the domatic partitioning problem, $\mathcal{F}$ consists of the dominating sets; these problems are NP-hard. In general, however, the size of the input may already be of order 2^n, and the best one can hope for is an algorithm with complexity within a polynomial factor of 2^n. Fairly recently [2], such a bound was indeed achieved via solving a somewhat harder-looking problem, namely that of *counting* all valid partitions. An intriguing question is, whether the base of the exponent can be lowered to $2-\epsilon$ for some $\epsilon > 0$, given that the size of the set family $\mathcal{F}$ is within a polynomial factor of c^n for some $c < 2$.

In this paper, we answer the question affirmatively in the special case where the given set family consists of sets whose cardinality is bounded by a constant. Throughout the paper the O^* notation suppresses a factor polynomial in n.

Theorem 1. *Given an n-element universe N, a number k, and a family $\mathcal{F}$ of subsets of N, each of cardinality at most r, the partitions of N into k members of $\mathcal{F}$ can be counted in time $O^*\left(|\mathcal{F}|\, 2^{n\lambda_r}\right)$, where $\lambda_r = (2r-2)\big/\sqrt{(2r-1)^2 - 2\ln 2}$.*

* This research was supported in part by the Academy of Finland, Grant 125637.

Previously, such an improved bound has been found in the special case where $\mathcal{F}$ contains only 2-sets, that is, pairs $\{u, v\} \subseteq N$. Then a valid partitioning corresponds to a perfect matching in a graph with vertex set N and edge set $\mathcal{F}$. While the existence of a perfect matching can be decided in polynomial time, the counting version is #P-complete [6]. The fastest known exact algorithm is by Björklund and Husfeldt [1], inspired by Williams's construction [7] and running in time $O^*\big(2^{n\omega/3}\big)$ where ω is the exponent of matrix multiplication. The Coppersmith–Winograd algorithm [4] shows $\omega < 2.38$ and, hence, the bound $O(1.732^n)$ [1]. The bound in Theorem 1 turns out to be slightly better, $O(1.653^n)$. In fact, the bound in Theorem 1 is somewhat crude for small r, and a specialized analysis yields yet a better bound.

Theorem 2. *The perfect matchings in a given graph on n vertices can be counted in time $O^*\big(\varphi^n\big)$, where $\varphi = (1+\sqrt{5})/2 = 1.618\ldots$ is the golden ratio.*

Note, however, that if $\omega = 2$, as conjectured by many, then the matrix multiplication algorithm remains faster, running in time $O(1.588^n)$.

We remark that the coefficient λ_r in Theorem 1 is only slightly larger than $(2r-2)/(2r-1) = 1 - 1/(2r-1)$ and amounts to a rather moderate growth of the bound with r. For example, for $r = 3$, 4, 5, and 6, Theorem 1 gives the bounds $O^*(|\mathcal{F}|\, c^n)$ with $c = 1.769$, 1.827, 1.862, and 1.885, respectively.

We will prove Theorems 1 and 2 (in Section 2) by giving a simple variant of the following folklore dynamic programming algorithm. For any $S \subseteq N$ and $j = 1, 2, \ldots, k$, let $f_j(S)$ be the number of ordered partitions of S into j members of $\mathcal{F}$. Then we have the recurrence

$$f_1(S) = [S \in \mathcal{F}]\ , \quad f_j(S) = \sum_{X \subseteq S} f_{j-1}(S \setminus X)\, [X \in \mathcal{F}] \quad \text{for } j > 1\ , \tag{1}$$

where $[P]$ is 1 if P is true and 0 otherwise. We note that by dynamic programming, the number of k-partitions of N, given as $f_k(N)/k!$, can be computed in time $O^*(|\mathcal{F}|\, 2^n)$, or for large $|\mathcal{F}|$ better in time $O^*(3^n)$. The bound can be reduced to $O^*(2^n)$ by implementing the dynamic programming step (1) using fast subset convolution [3].[1]

To lower the base of the exponent below 2, we will apply an innocent-looking modification, stemming from the idea of counting an ordered partition $(S_1, S_2, \ldots, S_k)$ only if its members are lexicographically ordered. It turns out that this simple constraint yields a substantial exponential speedup when the family $\mathcal{F}$ contains only sets whose cardinality is at most some constant r.

Finally, we note that our dynamic programming algorithm and the runtime analysis readily generalize to arbitrary commutative semirings. Thus, the bounds in Theorems 1 and 2 extend, for example, to the following variant in the min–sum semiring. Given a family of subsets of N, each member S associated with a real-valued cost $f(S)$, find the minimum total cost $f(S_1) + f(S_2) + \cdots + f(S_k)$ over the k-partitions $(S_1, S_2, \ldots, S_k)$, each S_i from the given family.

[1] If dynamic programming is replaced altogether by an inclusion–exclusion algorithm, the running times $O^*(|\mathcal{F}|\, 2^n)$ and $O^*(3^n)$ are achieved in polynomial space [2, 3].

2 Proof of Theorems 1 and 2

We modify the dynamic programming algorithm (1) to consider the members of a partition in a specific order. To this end, let N be an n-element set and $\mathcal{F}$ a family of subsets of N, each of size at most r. Fix a linear order $<$ on N and label the elements of N by $a_1 < a_2 < \cdots < a_n$. For any nonempty subset $S \subset N$ the minimum in S, $\min S$, is defined with respect to $<$ in the obvious way. Furthermore, define a lexicographic order, $\prec$, among the subsets of N, and hence in $\mathcal{F}$, with respect to the order $<$ on N in the usual manner; for instance, $\{a_1, a_2, a_5\} \prec \{a_1, a_3, a_4\} \prec \{a_2, a_4\}$.

While we are interested in counting the partitions of N into k members of $\mathcal{F}$, it turns out to be useful to consider ordered k-partitions $(S_1, S_2, \ldots, S_k)$ of N with the members from $\mathcal{F}$ and listed in the lexicographic order, that is, $S_i \prec S_j$ when $i < j$. We denote by $\mathcal{L}_k$ the set of such *lexicographically ordered k-partitions*, treating N and $\mathcal{F}$ as fixed. Since for any k-partition of N, the ordering of its members into the lexicographic order is unique, we have the following.

Lemma 1. *The number of partitions of N into k members of $\mathcal{F}$ equals the cardinality of $\mathcal{L}_k$.*

The lexicographic order implies certain constraints on the tuples $(S_1, S_2, \ldots, S_k) \in \mathcal{L}_k$, which amount to a reduction in the number of subsets of N that need be considered by a dynamic programming algorithm similar to (1). For example, the first set S_1 obviously must contain the smallest element of N. In general, the ith set S_i must contain the smallest element of N not contained by the preceding sets $S_1, S_2, \ldots, S_{i-1}$. Let $\mathcal{R}_j$ denote the family of sets S that can be expressed as the union of j such sets $S_1, S_2, \ldots, S_j$. Formally, we define the family of relevant sets $\mathcal{R}_j$, for $j = 1, 2, \ldots, n$, by the recurrence

$$\mathcal{R}_1 = \{X : X \in \mathcal{F}, \min N \in X\}\,;$$
$$\mathcal{R}_j = \{Y \cup X : Y \in \mathcal{R}_{j-1}, X \in \mathcal{F}, Y \cap X = \emptyset, \min N \setminus Y \in X\}\,.$$

We proceed by defining, for each $j = 1, 2, \ldots, n$, a set function g_j that associates any set $S \subseteq N$ with the number of ordered partitions $(S_1, S_2, \ldots, S_j)$ of S into j members of $\mathcal{F}$ such that the following condition holds:

$$\min N \setminus (S_1 \cup S_2 \cup \cdots \cup S_{i-1}) \in S_i \quad \text{for all } i = 1, 2, \ldots, j\,. \tag{2}$$

We note that for $S = N$, this condition is satisfied if and only if $(S_1, S_2, \ldots, S_j)$ is a lexicographically ordered partition of N. Thus, $g_k(N)$ equals the cardinality of $\mathcal{L}_k$. Our modified dynamic programming algorithm evaluates $g_k(N)$ using the following recurrence.

Lemma 2. *Let $S \subseteq N$. Then*

$$g_1(S) = [S \in \mathcal{R}_1] = [a_1 \in S] \tag{3}$$

and

$$g_j(S) = \sum_{Y \subseteq S} g_{j-1}(Y)\,[S \setminus Y \in \mathcal{F}]\,[\min N \setminus Y \in S \setminus Y]\,. \tag{4}$$

Proof. The first equality (3) holds by the definition of $\mathcal{R}_1$.

We then prove the recurrence (4). For any $Y \subseteq S$, define $g_j(S;Y)$ as the number of ordered partitions $(S_1, S_2, \ldots, S_j)$ of S into j members of $\mathcal{F}$ satisfying (2) and $S_1 \cup S_2 \cup \cdots \cup S_{j-1} = Y$. We note that

$$g_j(S;Y) = g_{j-1}(Y)\,[S \setminus Y \in \mathcal{F}]\,[\min N \setminus Y \in S \setminus Y]\;.$$

Because every $(S_1, S_2, \ldots, S_j)$ determines a unique Y, we have $g_j(S) = \sum_{Y \subseteq S} g_j(S;Y)$. □

It remains to analyze the time complexity of computing the values $g_j(S)$ for all relevant sets S via the recurrence (3–4). Straightforward induction shows that each g_j vanishes outside $\mathcal{R}_j$. Thus, the number of additions, multiplications and basic set operations of a straightforward implementation that first computes $g_1(S)$ for all $S \in \mathcal{R}_1$, then $g_2(S)$ for all $S \in \mathcal{R}_2$, and so on, is proportional to

$$\Big(|\mathcal{R}_1| + |\mathcal{R}_2| + \cdots + |\mathcal{R}_k|\Big)|\mathcal{F}|\;. \tag{5}$$

In the remainder of this section we derive upper bounds for this expression.

We begin with the special case where every member of the set family contains *exactly* 2 elements. In this case we have $|\mathcal{R}_j| \le \binom{n-j}{j}$, because each set in $\mathcal{R}_j$ is of size $2j$ and must contain the first j elements $a_1, a_2, \ldots, a_j$ and exactly j other elements from $\{a_{j+1}, a_{j+2}, \ldots, a_n\}$. Now, we make use of the following well-known relations[2] of the diagonal sums of the binomial coefficients, the Fibonacci sequence (F_n), and the golden ratio $\varphi = (1+\sqrt{5})/2$:

$$\sum_{j=0}^{n} \binom{n-j}{j} = F_{n+1} = \Big(\varphi^{n+1} - (1-\varphi)^{n+1}\Big)\Big/\sqrt{5} < \varphi^n\;, \tag{6}$$

This suffices for proving the bound $O^*(\varphi^n)$ for (5), and hence Theorem 2.

It is easy to generalize the bound $O^*(\varphi^n)$ to the case where every member of the set family contains *at most* 2 elements. In this case we have $|\mathcal{R}_j| \le \sum_{s=j}^{2j} \binom{n-j}{s-j} \le \sum_{t=0}^{j} \binom{n-t}{t}$, because each set in $\mathcal{R}_j$ is of size at most $2j$ and must contain the first j elements $a_1, a_2, \ldots, a_j$ and at most j other elements from $\{a_{j+1}, a_{j+2}, \ldots, a_n\}$. Thus, by (6), the sum $|\mathcal{R}_1| + |\mathcal{R}_2| + \cdots + |\mathcal{R}_k|$ is at most $k\varphi^n$.

We finally turn to the case of an arbitary size bound r. In this case we have $|\mathcal{R}_j| \le \sum_{s=j}^{rj} \binom{n-j}{s-j}$, because each set in $\mathcal{R}_j$ is of size at most rj and must contain the first j elements $a_1, a_2, \ldots, a_j$ and 0 to $rj - j$ other elements from $\{a_{j+1}, a_{j+2}, \ldots, a_n\}$. Now, the above analysis for $r = 2$ seems not to extend to $r > 2$, as it relies heavily on the special property of the diagonal sums of binomial coefficients. We therefore resort to a somewhat less accurate analysis, making use of the following specialization of the Hoeffding bounds:

[2] The author was pointed to these relations by two anonymous reviewers.

Theorem 3 (Hoeffding [5]). *Let $X_1, X_2, \ldots, X_n$ be independent Bernoulli trials with* $\Pr\{X_i = 1\} = \mu_i$ *for* $i = 1, 2, \ldots, n$. *Let* $X = \sum_{i=1}^{n} X_i$, $\mu = \sum_{i=1}^{n} \mu_i$, *and* $0 < t < 1 - \mu/n$. *Then*

$$\Pr\{X \leq \mu - tn\} \leq \exp[-2nt^2] \ .$$

Substituting $\mu_i \equiv 1/2$ and $t = 1/2 - k/n$ gives us a useful bound:

Corollary 1. *If* $n > 2k$, *then*

$$\sum_{j=0}^{k} \binom{n}{j} \leq 2^n \exp\Big[-2n\Big(\frac{1}{2} - \frac{k}{n}\Big)^2 \Big] \ .$$

We are now ready to prove the following lemma, which completes the proof of Theorem 1.

Lemma 3. *Let n and r be natural numbers. Then*

$$\sum_{s=j}^{jr} \binom{n-j}{s-j} < 2^{n\lambda_r} \ , \quad \textit{with } \lambda_r = \frac{r-1}{\sqrt{(r-1/2)^2 - \ln\sqrt{2}}} \ .$$

Proof. We consider two cases. First, suppose $jr - j \geq (n-j)/2$. Then $j \geq n/(2r-1)$, and we can bound the sum of the binomial coefficients above by $2^{n-j} \leq 2^{n(2r-2)/(2r-1)}$; the claim follows.

In the remaining case, suppose $jr - j < (n-j)/2$. Now it is handy to use $\ell = r - 1$. By Corollary 1,

$$\sum_{i=0}^{j\ell} \binom{n-j}{i} \leq 2^{n-j} \exp\left[-2(n-j)\left(\frac{1}{2} - \frac{j\ell}{n-j}\right)^2 \right] \ .$$

Letting $n - j = xn$, with $2\ell/(2\ell+1) \leq x \leq 1$, and

$$\psi(x) = x\left[\ln 2 - 2\left(\frac{1}{2} + \ell - \frac{\ell}{x}\right)^2 \right]$$

the bound becomes simply $\exp\big[n\psi(x)\big]$.

We next bound $\psi(x)$ in the relevant range. The derivative of $\psi(x)$ is

$$\psi'(x) = \ln 2 - 2\left(\frac{1}{2} + \ell - \frac{\ell}{x}\right)^2 - x4\left(\frac{1}{2} + \ell - \frac{\ell}{x}\right)\frac{\ell}{x^2} \ .$$

In terms of a new variable $y = \ell/x$, write

$$\begin{aligned} \psi'(\ell/y) &= \ln 2 - 2\left(\frac{1}{2} + \ell - y\right)^2 - 4\left(\frac{1}{2} + \ell - y\right)y \\ &= \ln 2 - 2\left(\frac{1}{2} + \ell - y\right)\left(\frac{1}{2} + \ell + y\right) \ . \end{aligned}$$

Solving for $\psi'(\ell/y) = 0$ yields

$$(\ln 2)/2 - \left(\frac{1}{2} + \ell\right)^2 + y^2 = 0$$

$$y^2 = \left(\frac{1}{2} + \ell\right)^2 - \ln\sqrt{2}\,.$$

Thus, $\psi(x)$ is maximized at

$$\tilde{x} = \frac{\ell}{\sqrt{(1/2+\ell)^2 - \ln\sqrt{2}}} > \frac{\ell}{1/2+\ell} = \frac{2\ell}{2\ell+1}\,.$$

Now we may bound $\psi(\tilde{x})$ as

$$\psi(\tilde{x}) < \tilde{x}\ln 2 = \frac{\ell\ln 2}{\sqrt{(1/2+\ell)^2 - \ln\sqrt{2}}}\,.$$

Recalling $\ell = r - 1$ we arrive at the claimed bound. □

Acknowledgements

The author is grateful to Andreas Björklund, Thore Husfeldt, and Petteri Kaski for valuable discussions, and to four anonymous reviewers for suggestions that helped to improve the presentation.

References

1. Björklund, A., Husfeldt, T.: Exact algorithms for Exact Satisfiability and Number of Perfect Matchings. Algorithmica 52, 226-249 (2008)
2. Björklund, A., Husfeldt, T., Koivisto, M.: Set partitioning via inclusion–exclusion. SIAM J. Comput., Special Issue for FOCS 2006, to appear
3. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets Möbius: fast subset convolution. In: 39th ACM Symposium on Theory of Computing (STOC 2007), pp. 67–74. ACM Press (2007)
4. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symb. Comput. 9, 251–280 (1990)
5. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. American Stat. Assoc. 58, 13–30 (1963)
6. Valiant, L.G.: The complexity of computing the permanent. Theor. Comput. Sci. 8, 189–201 (1979)
7. Williams, R.: A new algorithm for optimal 2-constraint satisfaction and its implications. Theor. Comput. Sci. 348, 357–365 (2005)