Fourier Meets Möbius: Fast Subset Convolution

Andreas Björklund
Lund University
Department of Computer Science
P.O.Box 118, SE-22100 Lund, Sweden
andreas.bjorklund@anoto.com

Thore Husfeldt
Lund University
Department of Computer Science
P.O.Box 118, SE-22100 Lund, Sweden
thore.husfeldt@cs.lu.se

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

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

ABSTRACT

We present a fast algorithm for the *subset convolution problem*: given functions f and g defined on the lattice of subsets of an n-element set N, compute their *subset convolution* $f*g$, defined for all $S \subseteq N$ by

$$(f * g)(S) = \sum_{T \subseteq S} f(T)g(S \setminus T)\,,$$

where addition and multiplication is carried out in an arbitrary ring. Via Möbius transform and inversion, our algorithm evaluates the subset convolution in $O(n^2 2^n)$ additions and multiplications, substantially improving upon the straightforward $O(3^n)$ algorithm. Specifically, if the input functions have an integer range $\{-M, -M+1, \ldots, M\}$, their subset convolution over the ordinary sum–product ring can be computed in $\tilde{O}(2^n \log M)$ time; the notation $\tilde{O}$ suppresses polylogarithmic factors. Furthermore, using a standard embedding technique we can compute the subset convolution over the max–sum or min–sum *semiring* in $\tilde{O}(2^n M)$ time.

To demonstrate the applicability of fast subset convolution, we present the first $\tilde{O}(2^k n^2 + nm)$ algorithm for the Steiner tree problem in graphs with n vertices, k terminals, and m edges with bounded integer weights, improving upon the $\tilde{O}(3^k n + 2^k n^2 + nm)$ time bound of the classical Dreyfus–Wagner algorithm. We also discuss extensions to recent $\tilde{O}(2^n)$-time algorithms for covering and partitioning problems (Björklund and Husfeldt, FOCS 2006; Koivisto, FOCS 2006).

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.1 [**Discrete Mathematics**]: Combinatorics; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Algorithms, Theory

Keywords

Convolution, Möbius transform, Steiner tree

STOC'07, June 11–13, 2007, San Diego, California, USA.

1. INTRODUCTION

1.1 Background and Main Result

Many hard computational problems admit a recursive solution via a convolution-like recursion step over the subsets of an n-element ground set N. More precisely, for every $S \subseteq N$, one computes the "solution" $h(S)$ defined by

$$h(S) = \sum_{T \subseteq S} f(T)g(S \setminus T)\,, \tag{1}$$

where $f(T)$ and $g(S \setminus T)$ are previously computed solutions for the subproblems specified by T and $S \setminus T$, and the arithmetic is carried out in an appropriate semiring; the most common examples in applications being perhaps the integer sum–product ring and the integer max–sum semiring. Given f and g, a direct evaluation of h for all $S \subseteq N$ requires $\Omega(3^n)$ semiring operations. To our knowledge, this is also the fastest known evaluation approach until the present work.

In a first attempt to improve upon the direct evaluation, the convolution analogy suggests the natural approach to evaluate (1) as a product of some type of Fourier transforms of f and g via a fast Fourier transform (FFT) and its inverse—in general, this approach has proven to be spectacularly successful in domains ranging from signal processing to number theory; see Maslen and Rockmore [17] for a survey of generalized FFTs. For example, considering a slightly different convolution operation of the form

$$h'(S) = \sum_{T \subseteq N} f\big(T\big)g\big(S \Delta T\big)\,, \tag{2}$$

where $S\Delta T = (S \setminus T) \cup (T \setminus S)$, the Fourier approach immediately yields an evaluation approach requiring $O(n2^n)$ ring operations via the fast Fourier transform on $\mathbb{Z}_2^n$, the elementary Abelian group of order 2^n. However, the convolution (1) is "truncated" from $T \subseteq N$ to $T \subseteq S$, which in effect renders the operation somewhat incompatible with group-theoretic Fourier transforms and associated "natural" convolution operations performed over the entire group. A simple zero-padding trick allows one to evaluate (1) via the FFT on $\mathbb{Z}_3^n$ (equivalently, the classical n-dimensional FFT with padding on each dimension), but this unfortunately does not improve upon the direct evaluation strategy.

A second attempt at analogy will prove to be more successful. Indeed, the truncation to $T \subseteq S$ and the sum over all $T \subseteq S$ in (1) suggests a connection with the classical Möbius transform [1, 20, 25] on the lattice of subsets of N, which is further motivated by the fact that a fast algorithm is known for evaluating the Möbius transform and its inverse on the subset lattice.

It turns out that—in analogy with the Fourier approach—the evaluation of (1) in $O(n^2 2^n)$ ring operations can be achieved via a product ("convolution over rank") of "ranked" extensions of the classical Möbius transforms of f and g on the subset lattice, followed by a "ranked" Möbius inversion. This is the main result of this paper.

THEOREM 1. *The subset convolution over an arbitrary ring can be evaluated in $O(n^2 2^n)$ ring operations.*

Furthermore, the basic fast convolution operation admits considerable extensions and variations, which we believe will find applications beyond the ones we proceed to outline in what follows.

1.2 Application to Specific Computational Problems

Besides the algebraic complexity we also study the implementation of the fast subset convolution algorithm on the ordinary sum–product ring of integers. The model of computation assumed in our analyses is the random access machine with the restriction that arithmetic operations (including comparison) are considered unit-time only for integers of constant size. To avoid cumbersome expressions in runtime bounds we may use the notation $\tilde{O}$ to hide polylogarithmic factors, that is, we may denote $\tilde{O}(\tau)$ when we have $O(\tau \log^d \tau)$ for some constant d in the familiar Landau notation.

Our main result easily implies the following.

THEOREM 2. *The subset convolution over the integer sum–product ring can be computed in $\tilde{O}(2^n \log M)$ time, provided that the range of the input functions is $\{-M, -M+1, \ldots, M\}$.*

Combinatorial optimization problems usually concern the max–sum or min–sum *semiring*. While our fast subset convolution algorithm does not directly apply to semirings where additive inverses need not exist, we can, fortunately, embed the integer max–sum (min–sum) semiring into the integer sum–product ring.

THEOREM 3. *The subset convolution over the integer max–sum (min–sum) semiring can be computed in $\tilde{O}(2^n M)$ time, provided that the range of the input functions is $\{-M, -M+1, \ldots, M\}$.*

As an illustrative application of fast subset convolution, we accelerate the classical Dreyfus–Wagner algorithm [6] for the Steiner tree problem: given an undirected graph $G = (V, E)$, a weight $w(e) > 0$ for each edge $e \in E$, and a set of vertices $K \subseteq V$, find a minimum-weight subgraph H of G that connects the vertices in K. The Dreyfus–Wagner algorithm runs in $\tilde{O}(3^k n + 2^k n^2 + nm)$ time, where $n = |V|$, $m = |E|$, and $k = |K|$. We give the first $\tilde{O}(2^k n^2 + nm)$-time algorithm, provided that the edge weights are small integers.

The Dreyfus–Wagner algorithm and its variants play a key role in solving various related problems. For example, the Dreyfus–Wagner algorithm has recently been used as a subroutine in fixed parameter tractable algorithms for certain vertex cover problems [13], as well as for near-perfect phylogenetic tree reconstruction [4]. Regarding rectilinear Steiner trees (RSTs), Ganley [11] writes: "The algorithm of Dreyfus and Wagner is probably the most popular used to date for computing optimal RSTs in practice." Furthermore, other "hierarchical partitioning" algorithms similar to that of Dreyfus and Wagner seem to appear in the literature with no explicit connection to the Steiner tree problem; we will consider in some detail a recent algorithm by Scott, Ideker, Karp, and Sharan [22] for detecting signaling pathways in protein interaction networks. Our improvement via fast subset convolution concerns all these variants and applications, subject to the constraint that edge weights can be represented by small integers.

We also note that many classical graph partitioning problems [12] can be solved by counting all valid partitions via recursive application of subset convolution (over the integer sum–product ring). Thus, the present technique can be seen as a generalization of the authors' previous work [3, 16] based on inclusion–exclusion and applies to a wider family of partitioning problems. For example, we can now solve extended partitioning problems, such as finding *all* k-colorable induced subgraphs of a given n-vertex graph, in $\tilde{O}(2^n)$ total time.

1.3 Related Research and Discussion

Möbius transform and inversion play a central role in combinatorial theory, particularly in the theory of partially ordered sets, subset lattices being special cases [1, 20, 25]. The fast Möbius transform and inversion algorithms on the subset lattice can be considered folklore; Kennes [15] gives a formal treatment, but the algorithm is, in essence, that of Yates [27] for multiplying a vector of size 2^n by a Kronecker product of n matrices of size 2×2 in $O(n2^n)$ operations. As far as we know, the connection of (ranked) Möbius inversion and subset convolution has not been studied until the present work.

For partitioning problems, we (the first two authors and the last author, independently) recently found two mutually different inclusion–exclusion algorithms [3, 16], which anticipated the results of the present paper. Yet, these earlier results, even when combined, do not immediately yield the fast subset convolution algorithm. What remained to be discovered was, in essence, the role of (fast) ranked Möbius inversion.

For the Steiner tree problem, Fuchs, Kern, and Wang [8] presented an $O(2.684^k p(n))$ algorithm, and Mölle, Richter, and Rossmanith [18] found an algorithm that, for any fixed $\epsilon > 0$, runs in $O((2+\epsilon)^k p(n))$ time, where $p(n)$ is a polynomial function of n. Unfortunately, the degree of $p(n)$

grows rapidly when ϵ approaches zero, which renders the algorithm impractical for small ϵ; in a subsequent work [7], the degree of $p(n)$ is improved to $12\sqrt{\epsilon^{-1}\ln\epsilon^{-1}}$, resulting in bounds like $O(2.5^k n^{14.2})$ and $O(2.1^k n^{57.6})$. Our accelerated Dreyfus–Wagner algorithm is not only theoretically faster, but may also have practical value when k is large enough (say, $k > 25$).

The idea of embedding the integer max–sum or min-sum semiring into the sum–product ring is not new. Some well-known examples are Yuval's [28] and others' [10, 23, 24] approaches to compute shortest paths via (fast) matrix multiplication. In our case the embedding technique provides a more substantial gain. Indeed, compared to fast subset convolution, fast matrix multiplication algorithms involve large constant factors and their practical value is not clear; the exponential speedup offered by fast matrix multiplication is currently by the ratio $3/2.376$ [5] and cannot exceed $3/2$, whereas the exponential speedup offered by fast subset convolution is by the ratio $\log 3/\log 2 > 1.58 > 3/2$.

1.4 Organization

The next section is devoted to proving our main theorem (Theorem 1); in addition, we introduce some variants of the subset convolution problem together with corresponding fast algorithms. In Section 3 we give short proofs of Theorems 2 and 3 concerning the implementation in the sum–product ring and the max–sum and min–sum semirings, which are essential for the applications. We consider the Steiner tree problem in detail in Section 4; also other applications to extended partitioning and hypergraph problems are illustrated, but in somewhat less detail.

2. FAST SUBSET CONVOLUTION OVER A RING

Throughout this section we assume that R is an arbitrary (possibly noncommutative) ring and that N is a set of n elements, $n \geq 0$. Let f (respectively, g) be a function that associates with every subset $S \subseteq N$ an element $f(S)$ (respectively, $g(S)$) of the ring R.

2.1 Subset Convolution

Define the *convolution* $f * g$ for all $S \subseteq N$ by

$$(f * g)(S) = \sum_{T \subseteq S} f(T)g(S \setminus T)\,, \tag{3}$$

or, equivalently, in a more symmetric form

$$(f * g)(S) = \sum_{\substack{U,\,V \subseteq S \\ U \cup V = S \\ U \cap V = \emptyset}} f(U)g(V)\,. \tag{4}$$

It follows that the convolution operation is associative (and commutative if R is commutative).

2.2 Möbius Transform and Möbius Inversion on the Subset Lattice

We recall the classical Möbius transform and inversion formulas on the subset lattice together with their fast evaluation algorithms. The *Möbius transform* of f is the function $\hat{f}$ that associates with every $X \subseteq N$ the ring element

$$\hat{f}(X) = \sum_{S \subseteq X} f(S)\,. \tag{5}$$

Given the Möbius transform $\hat{f}$, the original function f may be recovered via the *Möbius inversion* formula

$$f(S) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} \hat{f}(X)\,. \tag{6}$$

The *fast Möbius transform* [15, 27] is the following algorithm for computing the Möbius transform (5) in $O(n2^n)$ ring operations. By relabeling if necessary, we may assume that $N = \{1, 2, \ldots, n\}$. To compute $\hat{f}$ given f, let initially

$$\hat{f}_0(X) = f(X)$$

for all $X \subseteq N$, and then iterate for all $j = 1, 2, \ldots, n$ and $X \subseteq N$ as follows:

$$\hat{f}_j(X) = \begin{cases} \hat{f}_{j-1}(X) & \text{if } j \notin X\,, \\ \hat{f}_{j-1}(X \setminus \{j\}) + \hat{f}_{j-1}(X) & \text{if } j \in X\,. \end{cases} \tag{7}$$

It is straightforward to verify by induction on j that this recurrence gives $\hat{f}_n(X) = \hat{f}(X)$ for all $X \subseteq N$ in $O(n2^n)$ ring operations. The inversion operation (6) can be implemented in a similar fashion. To compute f given $\hat{f}$, let initially

$$f_0(S) = \hat{f}(S)$$

for all $S \subseteq N$, and then iterate for all $j = 1, 2, \ldots, n$ and $S \subseteq N$ as follows:

$$f_j(S) = \begin{cases} f_{j-1}(S) & \text{if } j \notin S\,, \\ -f_{j-1}(S \setminus \{j\}) + f_{j-1}(S) & \text{if } j \in S\,. \end{cases} \tag{8}$$

Then we have $f_n(S) = f(S)$ for all $S \subseteq N$.

2.3 Ranked Möbius Transform and Inversion

The *ranked Möbius transform* of f is the function $\hat{f}$ that associates with every $k = 0, 1, \ldots, n$ and $X \subseteq N$ the ring element

$$\hat{f}(k, X) = \sum_{\substack{S \subseteq X \\ |S| = k}} f(S)\,. \tag{9}$$

In particular, the classical Möbius transform of f is obtained in terms of the ranked transform by taking the sum over k, that is, $\hat{f}(X) = \sum_{k=0}^{|X|} \hat{f}(k, X)$. For the ranked transform, inversion is achieved simply by

$$f(S) = \hat{f}(|S|, S)\,, \tag{10}$$

or, in a somewhat more redundant form,

$$f(S) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} \hat{f}(|S|, X)\,. \tag{11}$$

This latter expression, rather than the former one, provides the key to fast evaluation of the subset convolution (3). Namely, we will "invert" a function that, in general, cannot be represented via ranked Möbius transform but via a convolution (over rank) of two such transforms.

To set the stage, it is immediate that the ranked transform (9) can be computed in $O(n^2 2^n)$ ring operations by carrying out the fast transform (7) independently for each $k = 0, 1, \ldots, n$. Similarly, the ranked inversion (11) can be computed in $O(n^2 2^n)$ ring operations by carrying out the fast inversion (8) independently for each $k = 0, 1, \ldots, n$.

2.4 Fast Subset Convolution

For two ranked Möbius transforms, $\hat{f}$ and $\hat{g}$, define the convolution $\hat{f} \circledast \hat{g}$ for all $k = 0, 1, \ldots, n$ and $X \subseteq N$ by

$$(\hat{f} \circledast \hat{g})(k, X) = \sum_{j=0}^{k} \hat{f}(j, X)\hat{g}(k-j, X)\,. \tag{12}$$

Note that this convolution operation is over the rank parameter rather than over the subset parameter.

It now holds that the inversion operation (11) applied to $\hat{f} \circledast \hat{g}$ gives $f * g$. Indeed, first observe by (9) and (12) that for any $S \subseteq N$ we have

$$\begin{aligned}
&\sum_{X \subseteq S} (-1)^{|S \setminus X|} (\hat{f} \circledast \hat{g})(|S|, X) \\
&= \sum_{X \subseteq S} (-1)^{|S \setminus X|} \sum_{j=0}^{|S|} \hat{f}(j, X)\hat{g}(|S|-j, X) \\
&= \sum_{X \subseteq S} (-1)^{|S \setminus X|} \sum_{j=0}^{|S|} \sum_{\substack{U,\,V \subseteq X \\ |U| = j \\ |V| = |S| - j}} f(U)g(V)\,.
\end{aligned} \tag{13}$$

Because X ranges over all subsets of S, it follows that for any ordered pair (U, V) of subsets of S satisfying $|U| + |V| = |S|$, the term $f(U)g(V)$ occurs in the sum with sign $(-1)^{|S \setminus X|}$ exactly once for every X satisfying $U \cup V \subseteq X \subseteq S$. No other terms occur in the sum. Thus, collecting the terms associated with each pair (U, V) together, the coefficient of $f(U)g(V)$ is, by the Binomial Theorem,

$$\sum_{x=|U \cup V|}^{|S|} \binom{|S| - |U \cup V|}{x - |U \cup V|} (-1)^{|S|-x} = \begin{cases} 1 & \text{if } |U \cup V| = |S|\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Because $|U| + |V| = |S|$ and $|U \cup V| = |S|$ imply $U \cup V = S$ and $U \cap V = \emptyset$, it follows that (13) and (4) agree. In other words,

$$(f * g)(S) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} (\hat{f} \circledast \hat{g})(|S|, X)\,. \tag{14}$$

Given f and g, we can now evaluate $f * g$ in $O(n^2 2^n)$ ring operations by first computing the fast ranked Möbius transform of f and g, then taking the convolution (12) of the transforms $\hat{f}$ and $\hat{g}$, and inverting the result using fast ranked Möbius inversion. This establishes Theorem 1.

2.5 Variants and Extensions

There are two immediate ways to relax the subset convolution (4). First, the *covering* product is defined for all $S \subseteq N$ by

$$(f *_{\mathrm{c}} g)(S) = \sum_{\substack{U,\,V \subseteq S \\ U \cup V = S}} f(U)g(V)\,. \tag{15}$$

Second, the *packing* product is defined for all $S \subseteq N$ by

$$(f *_{\mathrm{p}} g)(S) = \sum_{\substack{U,\,V \subseteq S \\ U \cap V = \emptyset}} f(U)g(V)\,. \tag{16}$$

Given f and g, the covering product (15) can be evaluated in $O(n2^n)$ ring operations by computing the Möbius transforms $\hat{f}$ and $\hat{g}$, taking the elementwise (Hadamard) product $(\hat{f}\hat{g})(X) = \hat{f}(X)\hat{g}(X)$ of the transforms, and inverting the result using fast Möbius inversion. Indeed, observe first that

$$\sum_{X \subseteq S} (-1)^{|S \setminus X|} (\hat{f}\hat{g})(X) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} \sum_{U,V \subseteq X} f(U)g(V)\,.$$

Now, for each ordered pair (U, V) of subsets of S, the coefficient of the term $f(U)g(V)$ is 1 if $U \cup V = S$ and 0 otherwise. Thus,

$$(f *_{\mathrm{c}} g)(S) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} (\hat{f}\hat{g})(X)\,. \tag{17}$$

Given f and g, the packing product $f *_{\mathrm{p}} g$ can be evaluated in $O(n^2 2^n)$ ring operations by first computing the subset convolution $f * g$ and then convolving the result with the vector $\vec{1}$ with all entries equal to 1. Indeed, based on (4) it is not difficult to check that

$$f *_{\mathrm{p}} g = f * g * \vec{1} = f * \vec{1} * g = \vec{1} * f * g.$$

Besides the immediate extensions (15) and (16), also somewhat more subtle variants are possible. For example, using (4) and (15), define the *intersecting covering product* for all $S \subseteq N$ by

$$(f *_{\mathrm{ic}} g)(S) = \sum_{\substack{U,\,V \subseteq S \\ U \cup V = S \\ U \cap V \neq \emptyset}} f(U)g(V)\,. \tag{18}$$

A fast evaluation algorithm is now immediate from the observation $f *_{\mathrm{ic}} g = f *_{\mathrm{c}} g - f * g$. Also more precise control over the allowed intersection cardinalities $|U \cap V| = \ell$ besides the $\ell = 0$ ($f * g$) and $\ell > 0$ ($f *_{\mathrm{ic}} g$) cases can be obtained by modifying (12); however, we will not enter into detailed discussion. Some further variations are possible by restricting the domain, e.g., to any hereditary family of subsets of N; we omit the details.

3. MODEL OF COMPUTATION AND THE CHOICE OF RING

Up to this point we have worked with an abstract ring R, and have considered only the number of ring operations (addition, subtraction, multiplication) required to carry out the computations. To arrive at a more accurate analysis of the required computational effort, we must choose a concrete ring R, fix a representation for its elements, and evaluate the required effort in a model that parallels the operation of an actual physical computer. In what follows, the model of computation is the random access machine with the restriction that arithmetic operations (including comparison) are considered unit-time only for constant-size integers. In this model, two b-bit integers can be added, subtracted, and compared in $O(b)$ time, and multiplied in $O(b \log b \log\log b) = \tilde{O}(b)$ time [21], recently improved to $b \log b 2^{O(\log^* b)}$ [9].

3.1 Integer Sum–Product Ring

We prove Theorem 2. We consider the subset convolution; similar argumentation applies to the other variants in Section 2.5. By Theorem 1, we know that the subset convolution can be computed in $O(n^2 2^n)$ ring operations. It is thus sufficient to notice that any intermediate results, for which ring operations are performed, are $O(n \log M)$-bit integers. To see this, note first that the ranked Möbius transform of

an input function can be computed with integers between $-M2^n$ and $M2^n$. Given this we note that the convolution of ranked transforms can be computed with $O(n \log M)$-bit integers. Finally, the ranked Möbius inversion is computed by adding (and subtracting) $O(n \log M)$-bit integers $O(2^n)$ times.

3.2 Integer Max–Sum and Min–Sum Semirings

We prove Theorem 3. We consider the case of max–sum semiring; similar argumentation applies to the min–sum semiring. Without loss of generality we assume that the range of the input functions is $\{0, 1, \ldots, M\}$; otherwise, we may first add M to each value of both input functions, compute the convolution, and finally subtract $2M$ to get the correct output.

Let f and g be the two input functions. Let $\beta = 2^n + 1$ and $M' = \beta^M$. Define new mappings f' and g' from the subsets of N to $\{0, 1, \ldots, M'\}$ by $f' = \beta^f$ and $g' = \beta^g$. By Theorem 2 we can compute the subset convolution $f' * g'$ over the integer sum–product ring in $\tilde{O}(2^n \log M') = \tilde{O}(2^n M)$ time. It remains to show that we can, for all $S \subseteq N$, efficiently deduce the value of $\max_{T \subseteq S}\{f(T)+g(S\setminus T)\}$ given the value of $\sum_{T \subseteq S} f'(T)g'(S \setminus T)$.

We observe that, for all $S \subseteq N$, we have a polynomial representation

$$\begin{aligned}(f' * g')(S) &= \sum_{T \subseteq S} \beta^{f(T)+g(S\setminus T)} \\ &= \alpha_0(S) + \alpha_1(S)\beta + \cdots + \alpha_{2M}(S)\beta^{2M}\,,\end{aligned}$$

where, due to the choice of β, each coefficient $\alpha_r(S)$ is uniquely determined and equals the number of subsets T of S for which $f(T) + g(S \setminus T) = r$. Thus, for each $S \subseteq N$, we can find the largest r for which $\alpha_r(S) > 0$ in $\tilde{O}(M)$ time. This completes the proof.

4. APPLICATIONS

4.1 The Steiner Tree Problem

The Steiner tree problem is a classical NP-hard problem. Given an undirected graph $G = (V, E)$, a weight $w(e) > 0$ for each edge $e \in E = E(G)$, and a set of vertices $K \subseteq V = V(G)$, the task is to find a subgraph H of G that connects the vertices in K and has the minimum total weight $\sum_{e \in E(H)} w(e)$ among all such subgraphs of G. Because the edge weights are positive, an optimal subgraph H is necessarily a tree with leaves in K. In what follows a *Steiner tree* always refers to such an optimal subgraph.

To be able to apply Theorem 3, convolution over the min-sum ring, we assume in what follows that the edge weights $w(e)$ are integers from $\{1, 2, \ldots, M\}$. Furthermore, to simplify some expressions, we assume that M is a constant.

4.1.1 Dreyfus–Wagner Recursion

Dreyfus and Wagner [6] discovered a beautiful dynamic programming algorithm for finding a Steiner tree in $\tilde{O}(3^k n + 2^k n^2 + nm)$ time, where $n = |V|$, $m = |E|$, and $k = |K|$.

The key idea in the Dreyfus–Wagner algorithm is that a Steiner tree H connecting a given subset of vertices $Y \subseteq V$ in G has the following optimal decomposition property, assuming $|Y| \geq 3$. For every $q \in Y$, there exists a vertex $p \in V$, a nonempty proper subset $D \subset Y \setminus \{q\}$, and a decomposition $E(H) = E(H_1) \cup E(H_2) \cup E(H_3)$ of the edges such that (a) H_1 is a Steiner tree connecting $\{p, q\}$ in G, (b) H_2 is a Steiner tree connecting $\{p\} \cup D$ in G, and (c) H_3 is a Steiner tree connecting $\{p\} \cup (Y \setminus (D \cup \{q\}))$ in G. (See Dreyfus and Wagner [6] for a proof.) Note that the decomposition may be degenerate, e.g., we can have $p = q$, implying that H_1 is empty.

The optimal decomposition property enables the following *Dreyfus–Wagner recursion*. For a vertex subset $Y \subseteq V$, denote by $W(Y)$ the total weight of a Steiner tree connecting Y in G. To set up the base case, observe that for $|Y| \leq 1$ the weight $W(Y) = 0$ and for $|Y| = 2$ the weight $W(Y)$ can be determined by a shortest-path computation based on the edge weights $w(e)$. For $|Y| \geq 3$ the optimal decomposition property implies that we have for all $q \in Y$ and $X = Y \setminus \{q\}$ the recursion

$$W(\{q\} \cup X) = \min\left\{W\big(\{p, q\}\big) + g_p\big(X\big) \,:\, p \in V\right\}, \quad (19)$$

$$g_p(X) = \min\left\{W\big(\{p\} \cup D\big) + W\big(\{p\} \cup (X \setminus D)\big) \,:\, \emptyset \subset D \subset X\right\}. \quad (20)$$

The Steiner tree problem can be solved by computing the weight $W(K)$ via this recursion. A bottom-up evaluation of $W(K)$ relying on dynamic programming takes the claimed $\tilde{O}(3^k n + 2^k n^2 + nm)$ time; first all-pairs shortest paths are computed in $\tilde{O}(n^2 + nm)$ time (in $O(n^2 \log n + nm)$ basic operations) using, e.g., Johnson's algorithm [14]. Once the values $W(\{p\} \cup Y)$ and $g_p(Y)$ for all $Y \subset K$ and $p \in V$ have been computed and stored, an actual Steiner tree is easy to construct by tracing backwards a path of optimal choices in (19) and (20) [6]; this costs only $O(2^k + kn)$ simple operations, that is, $\tilde{O}(2^k \log n + kn)$ time.

4.1.2 Expediting the Dreyfus–Wagner Recursion

We apply the fast subset convolution over the min–sum semiring to expedite the evaluation of the Dreyfus–Wagner recursion in (20). However, we cannot simply replace (20) by fast subset convolution as each $g_p(X)$ is defined in terms of other values $g_r(Z)$, for $Z \subset X$ and $r \in V$, which need to be precomputed. To this end, we carry out the computations in a level-wise manner.

For each level $\ell = 2, 3, \ldots, k-1$ in turn, assume the value $W(\{q\} \cup X)$ has been computed and stored for all $X \subset K$ with $|X| \leq \ell - 1$ and $q \in V \setminus X$. To compute $g_p(X)$ for each $p \in V$ and $X \subset K$ with $|X| = \ell \geq 2$, define the function f_p for all $X \subseteq K$ by

$$f_p(X) = \begin{cases} W(\{p\} \cup X) & \text{if } 1 \leq |X| \leq \ell - 1\,, \\ \infty & \text{otherwise}\,. \end{cases} \quad (21)$$

Here we let ∞ in (21) denote an integer that is sufficiently large to exceed the weight of any tree in G; for example, $(n-1)M + 1$ suffices. Applying the subset convolution over the min–sum semiring, it is now immediate from (20) and (21) that $g_p(X) = (f_p * f_p)(X)$ holds for all $X \subseteq K$ with $|X| \leq \ell$. Thus, by Theorem 3, we can compute $g_p(X)$ for all $p \in V$ and $X \subset K$ with $|X| = \ell$ using n evaluations of the subset convolution with integers bounded by nM, which leads to $\tilde{O}(2^k n^2)$ total time; note that the $\tilde{O}$ notation hides a factor of k^3. In fact, we can do even better and save a factor of k by replacing the subset convolution with the covering product over the min–sum semiring. To see this, observe that because $W(Z) \leq W(Y)$ holds whenever

$Z \subseteq Y \subseteq V$, we have that (20) can also be computed as $g_p(X) = (f_p *_c f_p)(X)$, that is,

$$g_p(X) = \min\big\{W\big(\{p\} \cup T\big) + W\big(\{p\} \cup U\big) \;:\; \emptyset \subset T,U \subset X,\;\; T \cup U = X\big\}\,.$$

Once the values $g_p(X)$ have been computed for all $p \in V$ and $X \subset K$ with $|X| = \ell$, it is easy to compute $W(\{q\} \cup X)$ for all $X \subset K$ and $q \in V \setminus X$ with $|X| = \ell$ in $\tilde{O}\big(\binom{k}{\ell} n^2\big)$ time using (19). Computing the above steps for all levels $\ell = 2, 3, \ldots, k-1$ takes $\tilde{O}\big(2^k n^2 + nm\big)$ total time, including the time needed for computing all-pairs shortest paths. Finally, a Steiner tree can be constructed within the same time bound (see Section 4.1.1). We have thus established the following theorem, which we state in a form without the assumption that M is constant.

THEOREM 4. *The Steiner tree problem with edge weights in $\{1, 2, \ldots, M\}$ can be solved in $\tilde{O}(2^k n^2 M + nm \log M)$ time.*

4.2 A Rooted Tree Model for Signaling Pathways

Scott, Ideker, Karp, and Sharan [22] consider various models for signaling pathways in protein interaction networks. One of the two more general models they introduce is based on rooted trees, and leads to the following network problem. Given an undirected graph $G = (V, E)$, a weight $w(e)$ for each edge $e \in E$, a vertex subset $I \subseteq V$, and a positive integer k, the task is to find for each vertex $v \in V$ a tree of the minimum total weight among all k-vertex subtrees in G that are rooted at v and in which every leaf belongs to I.

Scott et al. [22] apply the color coding method of Alon, Yuster, and Zwick [2], which proceeds by carrying out a sequence of randomized trials. In each trial, every vertex v is given independently and uniformly at random a color $c(v) \in \{1, 2, \ldots, k\}$, and the following subtask is solved: for each vertex $v \in V$ and subset $S \subseteq \{1, 2, \ldots, k\}$ that contains $c(v)$, find a minimum-weight subtree with $|S|$ vertices that is (a) rooted at v, (b) contains a node of each color in S, and (c) in which every leaf belongs to I. Scott et al. give the following recurrence for the associated minimum weight, denoted by $W(v, S)$:

$$W(v, S) = \min\big\{A(v, S),\, B(v, S)\big\}\,,$$

where

$$A(v, S) = \min\big\{W(u, S \setminus \{c(v)\}) + w(u, v) \;:\; c(u) \in S \setminus \{c(v)\}\big\}\,,$$

$$B(v, S) = \min\big\{W(v, T) + W(v, U) \;:\; T \cap U = \{c(v)\},\, T \cup U = S\big\}\,,$$

with $W(v, \{c(v)\}) = 0$ if $v \in I$ and $W(v, \{c(v)\}) = \infty$ otherwise. A direct evaluation of this recurrence can be carried out in $\tilde{O}(3^k m)$ time [22], where $m = |E|$.

Armed with fast subset convolution, we can speed up the evaluation of the recurrence to $\tilde{O}(2^k m)$ time, assuming that the edge weights are small integers. Namely, proceeding simultaneously for all sets S of a given cardinality, the computation of $B(v, S)$ can be reduced to subset convolution over the integer min–sum semiring; the transformation is analogous to the one used in Section 4.1.2, so we omit details.

Scott et al. [22] also consider a different model based on two-terminal series-parallel graphs. In this case, too, the original $\tilde{O}(3^k n^2)$ algorithm can be accelerated to an $\tilde{O}(2^k n^2)$ algorithm by using fast subset convolution.

4.3 Partitioning Problems and Extensions

Consider the generic problem of partitioning an n-element set N into k disjoint subsets that each satisfy some desired property specified by an indicator function f on the subsets of N. Given, N, k, and f as input, the task is to decide whether there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of N such that $f(S_c) = 1$ for each $c = 1, 2, \ldots, k$. Many classical graph partitioning problems are of this form. For example, in graph coloring $f(S) = 1$ if and only if S is an independent set in the input graph with the vertices N. Likewise, in domatic partitioning f is the indicator of dominating sets.

Recently we [3, 16] discovered two different algorithms that solve the generic partitioning problem using the principle of inclusion and exclusion in $\tilde{O}(2^n)$ time, provided that $f(S)$ can be evaluated for all $S \subseteq N$ in $\tilde{O}(2^n)$ total time. Using fast subset convolution we obtain yet another $\tilde{O}(2^n)$ algorithm. Indeed, we observe that the number of valid partitions of N is given by $f^{*k}(N)$, where

$$f^{*k} = \underbrace{f * f * \cdots * f}_{k \text{ times}}\,.$$

Thus, we can count the valid partitions by $k-1$ subset convolutions, or even better, in $O(\log k)$ convolutions by using the doubling trick.

What is more, we can solve considerable extensions of partitioning problems within the same runtime bound. For example, we can find a maximal k-colorable induced subgraph (in fact, all such subgraphs) in $\tilde{O}(2^n)$ time by computing $f^{*k}(S)$ for all vertex subsets $S \subseteq N$. In a similar fashion, but using the packing product, we can decide whether the input graph G contains k disjoint cliques each of size at least ℓ in $\tilde{O}(2^n)$ time: we check if $f^{*_{\mathrm{p}}k}(N) > 0$, where $f(S) = 1$ if S is a clique in G with $|S| \geq \ell$, and $f(S) = 0$ otherwise.

Fast subset convolution allows us to solve not only flat partitioning problems but also *hierarchical* partitioning problems in $\tilde{O}(2^n)$ time. Consider, for example, a branching process that partitions the ground set N in a tree-structured manner, as follows. With probability α, a node $S \subseteq N$ is split uniformly at random into two proper subsets $T \subset S$ and $S \setminus T \subset S$, which are then further partitioned recursively; with the remaining probability $1 - \alpha$, the branching terminates at S, and S becomes a leaf of the tree. With each possible leaf $L \subseteq N$ we associate a number $f(L)$, and by $g(N)$ we denote the expected value of the product of $f(L)$ over all leafs of the (random) tree. Then $g(N)$ can be solved through a recursion for $S \subseteq N$:

$$g(S) = (1 - \alpha) f(S) + \alpha \cdot \frac{1}{2^{|S|} - 2} \sum_{\emptyset \subset T \subset S} g(T) g(S \setminus T)\,.$$

Using fast subset convolution we can compute $g(S)$ for all $S \subseteq N$ in a total of $\tilde{O}(2^n)$ arithmetic operations.

4.4 Spanning Problems in Hypergraphs

We conclude this section by illustrating more subtle applications to two NP-hard hypergraph problems (see Polzin and Daneshmand [19] and Warme [26]). We begin by recalling the appropriate hypergraph terminology. A *hypergraph*

is a pair $\mathcal{H} = (V, \mathcal{E})$, where V is a finite set and $\mathcal{E}$ is a set consisting of subsets of V. A hypergraph $\mathcal{J} = (W, \mathcal{F})$ is a *subhypergraph* of $\mathcal{H}$ if $W \subseteq V$ and $\mathcal{F} \subseteq \mathcal{E}$. A subhypergraph is *spanning* if $V = W$. A *path* in a hypergraph $\mathcal{H}$ is a sequence $(x_1, E_1, x_2, E_2, \ldots, E_\ell, x_{\ell+1})$ such that (a) $x_1, x_2, \ldots, x_{\ell+1} \in V$ are all distinct, (b) $E_1, E_2, \ldots, E_\ell \in \mathcal{E}$ are all distinct, and (c) $x_i, x_{i+1} \in E_i$ for all $i = 1, 2, \ldots, \ell$. A path *joins* x_1 to $x_{\ell+1}$. A hypergraph is *connected* if for all distinct $x, y \in V$ there exists a path joining x to y. A connected hypergraph is a *tree* if for all distinct $x, y \in V$ the path joining x to y is unique.

The *minimum connected spanning subhypergraph* (MCSH) problem asks, given a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a weight $w(E) > 0$ for each hyperedge $E \in \mathcal{E}$, to produce a connected spanning subhypergraph of $\mathcal{H}$ that has the minimum total weight, or to assert that none exists. The *minimum spanning tree* (MSTH) problem is otherwise similar to the MCSH problem, but in addition it is required that the subhypergraph must be a tree.

Assuming that $w(E) \in \{1, 2, \ldots, M\}$ for all $E \in \mathcal{E}$, both the MCSH problem and the MSTH problem can be solved in time $\tilde{O}(2^n M)$ using variants of the fast subset convolution over the min–sum semiring, where $n = |V|$. Indeed, define the function f for all $E \subseteq V$ by

$$f(E) = \begin{cases} w(E) & \text{if } E \in \mathcal{E}\,, \\ \infty & \text{otherwise}\,. \end{cases}$$

To solve the MCSH problem, we employ the intersecting covering product (18). Define the kth power of the intersecting covering product for all $k = 2, 3, \ldots$ by

$$f^{*_{\mathrm{ic}}k} = f *_{\mathrm{ic}} \left(f^{*_{\mathrm{ic}}(k-1)}\right), \quad f^{*_{\mathrm{ic}}1} = f.$$

Here the order in which the products are evaluated is relevant because the intersecting covering product is *not* associative. Now observe that (a) a MCSH can be constructed by augmenting a connected subhypergraph of $\mathcal{H}$ one hyperedge at a time, and (b) at most $n - 1$ hyperedges occur in a MCSH of $\mathcal{H}$. Thus, $f^{*_{\mathrm{ic}}k}(V) < \infty$ is the minimum weight of a connected spanning subhypergraph of $\mathcal{H}$ consisting of k hyperedges. By storing the functions $f^{*_{\mathrm{ic}}k}$ for each $k = 1, 2, \ldots, n-1$, the actual MCSH can be determined by tracing back the computation one edge at a time. To solve the MSTH problem, replace the intersecting covering product (18) with an intersecting covering product that in addition requires the cardinality of the intersection to be exactly 1; such a product can be obtained by a minor modification of (12).

Acknowledgments

This research was supported in part by the Academy of Finland, Grants 117499 (P.K.) and 109101 (M.K.).

5. REFERENCES

[1] M. Aigner, *Combinatorial Theory*, Springer, Berlin, 1979.

[2] N. Alon, R. Yuster, U. Zwick, Color-coding, *J. ACM* 42 (1995) 844–856.

[3] A. Björklund, T. Husfeldt, Inclusion–exclusion algorithms for counting set partitions, in: Proc. 47th IEEE Symposium on Foundations of Computer Science (Berkeley, Oct. 22–24, 2006), IEEE Computer Society, Los Alamitos, CA, 2006, pp. 575–582.

[4] G.E. Blelloch, K. Dhamdhere, E. Halperin, R. Ravi, R. Schwartz, S. Sridhar, Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction, in: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, 33rd International Colloquium (Venice, July 10–14, 2006), Proceedings, Part I, Lecture Notes in Computer Science 4051, Springer, Berlin, 2006, pp. 667–678.

[5] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, *J. Symbolic Comput.* 9 (1990) 251–280.

[6] S.E. Dreyfus, R.A. Wagner, The Steiner problem in graphs, *Networks* 1 (1971/72) 195–207.

[7] B. Fuchs, W. Kern, D. Mölle, S. Richter, P. Rossmanith, X. Wang, Dynamic programming for minimum Steiner trees, *Theory Comput. Syst.*, to appear.

[8] B. Fuchs, W. Kern, X. Wang, Speeding up the Dreyfus–Wagner algorithm for minimum Steiner trees, *Math. Meth. Oper. Res.*, to appear.

[9] M. Fürer, Faster integer multiplication, in: Proc. 39th ACM Symposium on Theory of Computing (San Diego, June 11–13, 2007), these proceedings.

[10] Z. Galil, O. Margalit, All pairs shortest paths for graphs with small integer length edges, *J. Comput. System Sci.* 54 (1997) 243–254.

[11] J.L. Ganley, Computing optimal rectilinear Steiner trees: a survey and experimental evaluation, *Discrete Appl. Math.* 90 (1999) 161–171.

[12] M. Garey, D. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., San Francisco, CA, 1979.

[13] J. Guo, R. Niedermeier, S. Wernicke, Parametrized complexity of vertex cover variants, in: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures, 9th International Workshop (Waterloo, Canada, Aug. 15–17, 2005), Lecture Notes in Computer Science 3608, Springer, Berlin, 2005, pp. 36–48.

[14] D.B. Johnson, Efficient algorithms for shortest paths in sparse networks, *J. ACM* 24 (1977) 1–13.

[15] R. Kennes, Computational aspects of the Moebius transform of a graph, *IEEE Transactions on Systems, Man, and Cybernetics* 22 (1991) 201–223.

[16] M. Koivisto, An $O^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion exclusion, in: Proc. 47th IEEE Symposium on Foundations of Computer Science (Berkeley, Oct. 22–24, 2006), IEEE Computer Society, Los Alamitos, CA, 2006, pp. 583–590.

[17] D.K. Maslen, D.N. Rockmore, Generalized FFTs—a survey of some recent results, in: L. Finkelstein, W.M. Kantor (Eds.), *Groups and Computation, II*, American Mathematical Society, Providence, RI, 1997, pp. 183–237.

[18] D. Mölle, S. Richter, P. Rossmanith, A faster algorithm for the Steiner tree problem, in: B. Durand, W. Thomas (Eds.), 23rd Symposium on Theoretical Aspects of Computer Science (Marseille,

Feb. 23–25, 2006), Lecture Notes in Computer Science 3884, Springer, Berlin, 2006, pp. 561–570.

[19] T. Polzin, S.V. Daneshmand, On Steiner trees and minimum spanning trees in hypergraphs, *Oper. Res. Lett.* 31 (2003) 12–20.

[20] G.-C. Rota, On the foundations of combinatorial theory. I. Theory of Möbius functions. *Z. Wahrscheinlichkeitstheorie und verw. Gebiete* 2 (1964) 340–368.

[21] A. Schönhage, V. Strassen, Schnelle Multiplikation großer Zahlen, *Computing* 7 (1971) 281–292.

[22] J. Scott, T. Ideker, R.M. Karp, R. Sharan, Efficient algorithms for detecting signaling pathways in protein interaction networks, *J. Comput. Biol.* 13 (2006) 133–144.

[23] R. Seidel, On the all-pairs-shortest-path problem in unweighted undirected graphs, *J. Comput. System Sci.* 51 (1995) 400–403.

[24] A. Shoshan, U. Zwick, All pairs shortest paths in undirected graphs with integer weights, in: Proc. 40th Symposium on Foundations of Computer Science (New York, Oct. 17–19, 1999), IEEE Computer Society, Los Alamitos, CA, 1999, pp. 605–614.

[25] R.P. Stanley, *Enumerative Combinatorics*, Vol. I, Cambridge University Press, Cambridge, 1997.

[26] D.M. Warme, Spanning Trees in Hypergraphs with Applications to Steiner Trees, Ph.D. Thesis, University of Virginia, 1998.

[27] F. Yates, *The Design and Analysis of Factorial Experiments*, Technical Communication No. 35, Commonwealth Bureau of Soil Science, Harpenden, UK, 1937.

[28] G. Yuval, An algorithm for finding all shortest paths using $N^{2.81}$ infinite-precision multiplications, *Inform. Process. Lett.* 4 (1976) 155–156.