Planar Negative k-Cycle

Paweł Gawrychowski[*] Shay Mozes[†] Oren Weimann[‡]

Abstract

Given an edge-weighted directed graph G, the Negative-k-Cycle problem asks whether G contains a negative-weight cycle with at most k edges. For $k = 3$ the problem is known as the NegativeTriangle problem and is equivalent to all-pairs shortest paths (and to min-plus matrix multiplication) and solvable in $O(n^3)$ time. In this paper, we consider the case of directed planar graphs. We show that the Negative-k-Cycle problem can be solved in $\min\{O(nk^2 \log n), O(n^2 \log n)\}$ time. Assuming the min-plus convolution conjecture, we then show, for $k > n^{1/3}$ that there is no algorithm polynomially faster than $O(n^{1.5}\sqrt{k})$, and for $k \leq n^{1/3}$ that our $O(nk^2 \log n)$ upper bound is essentially tight. The latter gives the first non-trivial tight bounds for a planar graph problem in P. Our lower bounds are obtained by introducing a natural problem on matrices that generalizes both min-plus matrix multiplication and min-plus convolution, and whose complexity lies between the complexities of these two problems.

1 Introduction

In this paper we present upper and lower bounds for the Negative-k-Cycle problem in planar graphs. In this problem, we are asked whether a given edge-weighted directed graph G contains a directed cycle with at most k edges whose total weight is negative. This is a natural problem that generalizes both the NegativeTriangle problem (where $k = 3$) and the NegativeCycle problem (where $k = n$).

In general graphs, the NegativeTriangle problem is solvable in $O(n^3)$ time and is equivalent to all-pairs shortest-paths (APSP), min-plus matrix multiplication, and many other problems [4, 20] in the sense that if one of them can be solved in truly subcubic $O(n^{3-\epsilon})$ time then all of them can. In sparse graphs, the NegativeTriangle problem is solvable in $O(m^{1.5})$ time [11] and the NegativeCycle problem is solvable in $O(mn)$ time. Recently, Orlin et al. [18] gave a randomized $O(mn \log n)$ time algorithm that finds the smallest k for which there is a negative-k-cycle, and Lincoln et al. [15] proved that (conditioned on the hardness of the minimum-weight k-Clique problem) when $m = \Theta(n^{1+1/k})$ there is no $O(n^2 + mn^{1-\epsilon})$ algorithm for Negative-$(2k + 1)$-Cycle.

In planar graphs, the situation is very different. The NegativeTriangle problem is solvable in linear $O(n)$ time and the NegativeCycle problem is solvable in near-linear $\tilde{O}(n)$ time [10, 12, 17]. Here is a simple $O(n)$ algorithm for NegativeTriangle: Every planar graph contains a vertex v of degree at most 6, check every two neighbors of v if they form a negative triangle with v, remove v from the graph and recurse. This simple trick works for $k = 3$ but for other values of k it is entirely unclear what is the complexity of the Negative-k-Cycle problem. In this paper we set out to resolve this problem. In particular,

Can Negative-k-Cycle *in planar graphs be solved in near-linear time?*

Until this paper, the only known bound for PlanarNegative-k-Cycle was the $O(n^{1.5}k)$ time algorithm by Williamson and Subramani [22]. Our first result is a faster algorithm:

Theorem 1.1. PlanarNegative-k-Cycle *can be solved in $O(nk^2 \log n)$ time deterministically and in $O(n^2 \log n)$ time with constant probability.*

The above running times strictly dominate [22] for any $k \neq \sqrt{n}$ and match [22] for $k = \sqrt{n}$. But can one do better? We use tools from fine-grained complexity in order to show that probably not. The two obvious candidates for showing such hardness results are min-plus matrix multiplication and min-plus convolution:

MinPlusMult
Input: $n \times n$ matrices A, B, C with integer entries in $[-W, W]$.
Output: Does $A[i][k] + B[k][j] \geq C[i][j]$ hold for every i, j, k?

[*]University of Wrocław, Poland. gawry@cs.uni.wroc.pl.
[†]The Interdisciplinary Center Herzliya, Israel. smozes@idc.ac.il. Supported partially by ISF grant 592/17.
[‡]University of Haifa, Israel. oren@cs.haifa.ac.il. Supported partially by ISF grant 592/17.

MinPlusConv
Input: n-length sequences a, b, c with integer entries in $[-W, W]$.
Output: Does $\min_{x+y=z} a[x] + b[y] \geq c[z]$ hold for every z?

Observe that the above problems are the decision versions (also called the "Lower Bound" versions) of min-plus multiplication and min-plus convolution. Cygan et al. [9] conjectured that min-plus convolution cannot be solved in $O(n^{2-\epsilon} \operatorname{polylog} W)$ time and proved that this implies that the decision version MinPlusConv cannot be solved in $O(n^{2-\epsilon} \operatorname{polylog} W)$ time. Vassilevska Williams and Williams [21] conjectured that APSP cannot be solved in $O(n^{3-\epsilon} \operatorname{polylog} W)$ time and proved that this implies that the decision version MinPlusMult cannot be solved in $O(n^{3-\epsilon} \operatorname{polylog} W)$ time. The min-plus convolution conjecture has already been used to establish quite a few lower bounds [1,3,5,6,8,9,13,14]. It is particularly appealing because it implies both the 3-SUM conjecture and the APSP (or equivalently MinPlusMult) conjecture (and therefore it implies all the lower bounds that follow from these two very popular conjectures, see [19]). Namely, Bremner et al. [7] showed that the min-plus convolution conjecture implies that MinPlusMult cannot be solved in $O(n^{3-\epsilon} \operatorname{polylog} W)$ time, and Cygan et al. [9] showed that it implies that 3-SUM cannot be solved in $O(n^{2-\epsilon} \operatorname{polylog} W)$ time.

The first conditional lower bound in planar graphs was given by Abboud and Dahlgaard [2] for the problem of maintaining a (dynamic) planar graph subject to edge insertions and deletions as well as distance queries between any two nodes. Abboud and Dahlgaard showed that, assuming the MinPlusMult conjecture, no algorithm can support both updates and queries in $O(n^{1/2-\epsilon})$ time. Very recently, Abboud et al. [1] established the first conditional lower bound for a *static* planar graph problem. They showed that, assuming the MinPlusConv conjecture, the Sparsest Cut problem (and also the related Minimum Quotient and Minimum Bisection problems) cannot be solved in $O(n^{2-\epsilon})$ time on a planar graph with n vertices, unit vertex-weights, and total edge cost $C = n^{O(1)}$. However, this does not match their upper bound of $O(n^{3/2} W \log(CW))$, as for instances obtained in the lower bound this is $O(n^{5/2} \log n)$.

Consider the following simple reduction (inspired but different from Abboud et al. [1]) from MinPlusConv to PlanarNegative-k-Cycle with $k = n$: Let $a'[i] = a[i] - i \cdot M$, $b'[i] = b[i] - i \cdot M$ and $c'[i] = -c[n-1-i] + (n-1-i)M$, where M is sufficiently large, say $M = 3W + 1$. Then finding $x + y = z$ such that $a[x] + b[y] < c[z]$ is equivalent to finding x, y, z such that $x + y + z \leq n - 1$ and $a'[x] + b'[y] + c'[z] < 0$. The graph consists of three paths (1) $u_0, u_1, \ldots, u_n$, (2) $v_0, v_1, \ldots, v_n$, and (3) $w_0, w_1, \ldots, w_n$. We identify u_n with v_0, v_n with w_0 and w_n with u_0, and add zero-weight edges (u_i, u_{i+1}), (v_i, v_{i+1}) and (w_i, w_{i+1}), for every $i = 0, 1, \ldots, n$. Finally, we add an edge (u_0, u_{n-i}) with weight $a'[i]$, (v_0, v_{n-i}) with weight $b'[i]$ and (w_0, w_{n-i}) with weight $c'[i]$. See Figure 1.

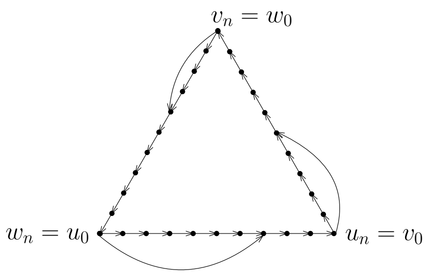

Figure 1: Illustration of the reduction for $k = \Theta(n)$.

The obtained planar graph on $O(n)$ nodes contains a negative cycle consisting of at most $n + 2$ edges if and only if there are x, y, z such that $a[x] + b[y] < c[z]$. Consequently, we should not hope for an $O(n^{2-\epsilon})$ time algorithm for the PlanarNegative-k-Cycle problem with $k = n$. In particular, this excludes a near-linear time algorithm. However, because this reduction uses a large value of k, it might be still possible that there exists, say, an $O(nk)$ time algorithm. This raises the following question:

Can Negative-k-Cycle *in planar graphs be solved in $\tilde{O}(nk)$ time?*

To determine the complexity of Negative-k-Cycle for the case of sublinear values of k, we introduce a natural variant of matrix multiplication that generalizes both min-plus matrix multiplication and min-plus convolution. We hope that this new variant will be useful to study the fine-grained complexity of other problems with two parameters.

MinPlusConvMult
Input: $n \times n$ matrices A, B, C whose entries are sequences of length s of integers in $[-W, W]$.
Output: Does $\min_{x+y=z} A[i][k][x] + B[k][j][y] \geq C[i][j][z]$ hold for every i, j, k, z?

Clearly, when $n = 1$ MinPlusConvMult degenerates to MinPlusConv and when $s = 1$ it degenerates to MinPlusMult. Algorithmically, the naive $O(n^3)$ algorithm for MinPlusConv and the naive $O(n^2)$ algorithm for MinPlusMult imply a naive $O(n^3s^2)$ algorithm for MinPlusConvMult. In terms of complexity as a function of the size of the input, without any restrictions on the parameters n and s, MinPlusConvMult is not easier than MinPlusConv and MinPlusMult. However, in order to prove the hardness of Negative-k-Cycle, we will be interested in the case where s is restricted to be polynomial in n. When this is the case, we show that, informally speaking, the complexity of MinPlusConvMult lies between min-plus convolution and min-plus matrix multiplication. More precisely, we show the following:

Theorem 1.2. *For any $\alpha, \epsilon > 0$ there exists $\epsilon' > 0$ s.t. when $s = n^\alpha$ an $O(n^3s^{2-\epsilon})$ time algorithm for* MinPlusConvMult *implies an $O(n^{2-\epsilon'})$ time algorithm for* MinPlusConv*, and an $O(n^{3-\epsilon})$ time algorithm for* MinPlusMult *implies an $O(n^3s^{2-\epsilon'})$ time algorithm for* MinPlusConvMult*.*

We then show a reduction from MinPlusConvMult (with parameters n, s) to PlanarNegative-k-Cycle (with parameters $k = n + s$, and $n' = n^2s$). By appropriately adjusting the parameters, this results in the following theorem:

Theorem 1.3. *Assuming the min-plus convolution conjecture, for $k > n^{1/3}$ there is no algorithm for* PlanarNegative-k-Cycle *polynomially faster than $O(n^{1.5}\sqrt{k})$, and for $k \leq n^{1/3}$ there is no algorithm polynomially faster than $O(nk^2)$.*

In particular, for $k \leq n^{1/3}$ we obtain tight upper and lower bounds (up to sub-polynomial terms). This is the first non-trivial tight bound for a planar graph problem in P.

If one is willing to believe in the hardness of MinPlusMult but not MinPlusConv, our reduction is still meaningful, as it implies the following theorem:

Theorem 1.4. *Assuming the APSP conjecture, there is no algorithm for* PlanarNegative-k-Cycle *polynomially faster than $O(nk)$.*

2 Upper Bound

In this section we prove Theorem 1.1. Let G' be the unweighted undirected planar graph obtained by disregarding the weights and the directions of all edges of G. We can assume that G' is connected. Let T be a BFS tree in G' rooted at an arbitrary vertex, and let $d(v)$ denote the depth of vertex v in T. We define the *i-th slice* of T to contain all vertices v such that $d(v) \in [i \cdot k, i \cdot k + 2k)$. Notice that the slices are not disjoint, but every vertex belongs to at most two slices. Consider a directed cycle C with at most k edges in G. Let $v_1, v_2, \ldots, v_\ell$ denote the vertices of C ordered such that $d(v_1) \leq d(v_2) \leq \cdots \leq d(v_\ell)$. Then, by construction of G' and since T is a BFS tree we have that $d(v_j) \in [d(v_1), k + d(v_1)]$ for all j. This implies that all vertices of C belong to the $\lfloor d(v_1)/k \rfloor$-th slice. Consequently, it is enough to solve the problem separately for the subgraph of G induced by the nodes of every slice. We next give an $O(n'k^2)$ time solution for a subgraph of size n' leading to an overall running time of $O(nk^2)$.

Consider the subgraph G_i induced in G by the nodes of the i-th slice, and let G'_i be the unweighted undirected planar graph obtained by disregarding the weight and the directions of all edges of G_i. Again, we can assume that G'_i is connected by adding extra edges incident to a new node. By construction, there exists a spanning tree T_i of G'_i of depth $O(k)$. It is well known [16] that under such assumption G'_i and any of its subgraphs admit a balanced vertex separator of size $O(k)$ that can be found in linear time. We run a divide-and-conquer procedure that works as follows: After having found in linear time a balanced vertex separator S of size $O(k)$, for every node $u \in S$ we check in $O(n'k)$ time if there exists a negative directed cycle consisting of at most k edges and containing u. Then, we remove S and recurse on the obtained smaller subgraphs. Solving the recurrence we obtain that the running time is $O(n'k^2 \log n)$.

It remains to explain how to check in $O(n'k)$ time if there exists a negative directed cycle consisting of at most k edges and containing u. This is done by simply running k iterations of Bellman-Ford. More formally, we proceed in k phases. In the i phase we calculate for every v the distance $d_i(v)$ from u to v consisting of at most i edges. Initially, $d_0(u) = 0$ and $d_0(v) = \infty$ for every $v \neq u$. The values $d_{i+1}(v)$ can be computed from the values $d_i(v)$ in $O(n')$ time by relaxing all the edges. Overall, the running time is $O(n'k)$ and at the end $d_0(u) < 0$ if and only if there is a negative cycle with at most k edges that contains u.

Finally, we explain how to obtain the alternative $O(n^2 \log n)$ bound in the theorem's statement using standard random sampling. We do this in $\log k$ phases. The i'th phase checks if there is a negative k'-cycle for $k' \in [2^i, 2^{i+1}]$. The i'th phase begins by randomly sampling each vertex of G with probability $\frac{1}{2^i}$. If there exists a negative k'-cycle for $k' \in [2^i, 2^{i+1}]$, then with constant probability one of the sampled vertices belongs to this cycle. Therefore, as explained above, we can find this cycle by running 2^{i+1} iterations of Bellman-Ford from each sampled vertex. This takes

$O(\frac{n}{2^i} \cdot n \cdot 2^{i+1}) = O(n^2)$ time, so over all phases we get $O(n^2 \log n)$ time.

3 Lower Bound

In order to consider the running time as a function of n, we assume that in a MinPlusConvMult instance $s = n^\alpha$ for some constant $\alpha > 0$. We start by proving Theorem 1.2 that, informally speaking, says that MinPlusConvMult lies between MinPlusConv and MinPlusMult. This is established through two reductions, both inspired by the reduction from min-plus convolution to APSP [7].

Lemma 3.1. *For any $\alpha, \epsilon > 0$, if* MinPlusConvMult *can be solved in $O(n^{3+2\alpha-\epsilon})$ time then* MinPlusConv *can be solved in $O(n^{2-\epsilon/(2+\alpha)})$ time.*

Proof. We will reduce an instance of MinPlusConv on sequences of length n to m instances of MinPlusConvMult on matrices of size $m \times m$ and $s = 2n^{\alpha/(2+\alpha)}$, so $m = s^{1/\alpha} = n^{1/(2+\alpha)}$. By substituting the assumed complexity of MinPlusConvMult, we then obtain that the complexity of MinPlusConv is

$$\begin{aligned} O(m \cdot m^{3+2\alpha-\epsilon}) &= O(n^{1/(2+\alpha)} \cdot n^{(3+2\alpha-\epsilon)/(2+\alpha)}) \\ &= O(n^{2-\epsilon/(2+\alpha)}). \end{aligned}$$

Let $\ell = n/m^2 = n^{\alpha/(2+\alpha)}$. We start with partitioning all sequences into $n/\ell = n^{2/(2+\alpha)}$ blocks of length ℓ, and use a_i, b_i, c_i to denote the i-th block of the sequence $(a[i])_{i=0}^{n-1}$, $(b[i])_{i=0}^{n-1}$, $(c[i])_{i=0}^{n-1}$, respectively. Further, let a'_i, b'_i be blocks of length 2ℓ obtained from a_i, b_i by appending ℓ copies of ∞, and let c'_i be a block of length 2ℓ obtained by concatenating c_i and c_{i+1} (or appending ℓ copies of $-\infty$ if c_{i+1} does not exist). Recall that our task is to check if $\min_{x+y=z} a[x] + b[y] \geq c[z]$ holds for every z, or equivalently $a[x] + b[y] \geq c[x + y]$ holds for every x, y. Because $\lfloor (x+y)/\ell \rfloor$ is either $\lfloor x/\ell \rfloor + \lfloor y/\ell \rfloor$ or $\lfloor x/\ell \rfloor + \lfloor y/\ell \rfloor + 1$, this is now equivalent to checking if, for every i, j and x, y we have $a'_i[x] + b'_j[y] \geq c'_{i+j}[x + y]$.

We now define the instances of MinPlusConvMult. In all instances, $A[i][k]$ is simply $a'_{(i-1)m+k}$. In the t-th instance, for $t = 0, 1, \ldots, m-1$, $B[k][j]$ is $b'_{t+(j-1)m-k}$ and $C[i][j]$ is $c'_{t+(i+j-2)m}$, where the nonexistent a'_i, b'_i and c'_i are assumed to consist of 2ℓ copies ∞, ∞ and $-\infty$, respectively. We need to argue that solving all these instances is equivalent to checking the aforementioned condition.

Assume that in the t-th instance of MinPlusConvMult we have i, j, x, y such that $A[i][k][x] + B[k][j][y] < C[i][j][x + y]$. By substituting the definitions, this implies

$$a'_{(i-1)m+k}[x] + b'_{t+(j-1)m-k}[y] < c'_{t+(i+j-2)m}[x + y]$$

so we have $a'_{i'}[x] + b'_{j'}[y] < c'_{i'+j'}[x+y]$ for some i', j', x, y.

Now assume that we have $a'_i[x] + b'_j[y] < c'_{i+j}[x + y]$ for some i, j, x, y. Set $t = (i + j) \bmod m$ and $k = (i - 1) \bmod m + 1$. Then, both $i - k$ and $j - t + k$ are nonnegative divisible by m. We can therefore find i', j' such that $(i' - 1)m + k = i$, $t + (j' - 1)m - k = j$ and $t + (i' + j' - 2)m = i + j$. Again by substituting the definitions, this implies

$$A[i'][k][x] + B[k][j'][y] < C[i'][j'][x + y]$$

for some i', j', k, x, y. ∎

Lemma 3.2. *For any $\alpha, \epsilon > 0$, if* MinPlusMult *can be solved in $O(n^{3-\epsilon})$ time then* MinPlusConvMult *can be solved in $O(n^{3+2\alpha-\epsilon(1+\alpha/2)})$ time.*

Proof. We will reduce an instance of MinPlusConvMult on $n \times n$ matrices with entries being sequences of length $s = n^\alpha$ to $\sqrt{s}$ instances of MinPlusMult on $n\sqrt{s} \times n\sqrt{s}$ matrices. By substituting the assumed complexity of MinPlusMult, we then obtain that the complexity of MinPlusConvMult is

$$O(n^{\alpha/2} n^{(1+\alpha/2)(3-\epsilon)}) = O(n^{3+2\alpha-\epsilon(1+\alpha/2)}).$$

Bremner et al. [7, Theorem 10] showed how to reduce MinPlusConv to MinPlusMult. Their reduction can be thought of as turning an instance of MinPlusConv of sequences of length s into $\sqrt{s}$ instances of MinPlusMult of $\sqrt{s} \times \sqrt{s}$ matrices. Our reduction proceeds in two steps. In the first step, we follow the same approach as [7] for each sequence in the MinPlusConvMult instance. The result of this process is a set of $\sqrt{s}$ matrices $\{A_t[1..n][1..n]\}_{t=0}^{\sqrt{s}-1}$, where the entry $A_t[i][j]$ is the t'th $\sqrt{s} \times \sqrt{s}$ matrix from the reduction in [7] (and similarly for B and C). Doing so guarantees the following for every i, j, k: there exist x, y such that $A[i][k][x] + B[k][j][y] < C[i][j][x + y]$ if and only if there exist t and i', j', k' such that $A_t[i][k][i'][k'] + B_t[k][j][k'][j'] < C_t[i][j][i'][j']$.

In the second step we convert the $n \times n \times \sqrt{s} \times \sqrt{s}$ matrices A_t, B_t, C_t into $n\sqrt{s} \times n\sqrt{s}$ matrices A'_t, B'_t, C'_t by setting

$$\begin{aligned} A'_t[(i-1)\sqrt{s} + i'][(k-1)\sqrt{s} + k'] &= A_t[i][k][i'][k'], \\ B'_t[(k-1)\sqrt{s} + k'][(j-1)\sqrt{s} + j'] &= B_t[k][j][k'][j'], \\ C'_t[(i-1)\sqrt{s} + i'][(j-1)\sqrt{s} + j'] &= C_t[i][j][i'][j']. \end{aligned}$$

By definition, there exist i, j, k, i', j', k' such that $A_t[i][k][i'][k'] + B_t[k][j][k'][j'] < C_t[i][j][i'][j']$ if and only if there exist i, j, k such that $A'_t[i][k] + B'_t[k][j] < C_t[i][j]$. Consequently, solving all the obtained instances of MinPlusMult is equivalent to solving the original instance of MinPlusConvMult.

We note that a weaker conclusion that MinPlusConvMult can be solved in $O(n^{3+2\alpha-\epsilon})$ time can be obtained by iterating over every x, y (and $z = x + y$) and creating a MinPlusMult instance corresponding to choosing $A[i][j][x]$, $B[j][k][y]$ and $C[i][k][z]$. □

Having shown that MinPlusConvMult is between MinPlusConv and MinPlusMult, we now proceed to show a reduction from MinPlusConvMult to PlanarNegative-k-Cycle.

Lemma 3.3. *Given an instance of* MinPlusConvMult*, we can construct in linear time an equivalent instance of* PlanarNegative-k-Cycle *on $O(n^2 s)$ nodes and $k = O(n + s)$.*

Before we proceed to the proof, let us analyze the implications of Lemma 3.3 on the complexity of PlanarNegative-k-Cycle on n nodes and $k = n^\beta$, assuming the min-plus convolution conjecture of Cygan et al. [9]:

Conjecture 3.1. *For any $\epsilon > 0$,* MinPlusConv *cannot be solved in $O(n^{2-\epsilon} \operatorname{polylog} W)$ time.*

For $\beta > 1/3$ we proceed as follows. We consider MinPlusConvMult with $s = n^\alpha = n^{2\beta/(1-\beta)}$. By Lemma 3.3 it can be reduced to PlanarNegative-k-Cycle on $N = O(n^{2+2\beta/(1-\beta)})$ nodes and $k = O(n^{2\beta/(1-\beta)})$. Conjecture 3.1 together with Lemma 3.1 implies that there is no $O(n^{3+2\alpha-\epsilon}) = O(n^{3+4\beta/(1-\beta)-\epsilon}) = O(N^{3/2}k^{1/2-\epsilon'})$ time algorithm for this case. For $\beta \le 1/3$ we proceed slightly differently. We consider MinPlusConvMult with $s = n^\alpha = n^{3\beta}$. By partitioning each matrix into $s \times s$ blocks, such an instance can be reduced using a standard technique to $(n/s)^3$ instances of MinPlusConvMult on $s \times s$ matrices with the same s. By Lemma 3.3, each of the smaller instances can be reduced to PlanarNegative-k-Cycle on $O(s^3)$ nodes and $k = O(s)$. These instances can be then glued together to obtain a single graph on $N = O(n^3)$ nodes and $k = O(s)$. Conjecture 3.1 together with Lemma 3.1 imply that there is no $O(n^{3+2\alpha-\epsilon}) = O(Nk^{2-\epsilon})$ time algorithm for this case. These two cases together establish Theorem 1.3.

Lemma 3.3 can be also be used in conjunction with the (perhaps more believable) APSP conjecture:

Conjecture 3.2. *For any $\epsilon > 0$,* MinPlusMult *cannot be solved in $O(n^{3-\epsilon} \operatorname{polylog} W)$ time.*

We consider an instance of MinPlusMult. Let $\ell = n^{3\alpha/(2\alpha+1)}$. By partitioning each matrix into $\ell \times \ell$ blocks and considering the blocks containing the sought $A[i][k]$, $B[k][j]$ and $C[i][j]$, such an instance can be reduced to $(n/\ell)^3$ instances of MinPlusMult on $\ell \times \ell$ matrices. Because MinPlusMult can be treated as MinPlusConvMult with $s = 1$, By Lemma 3.3, each of the smaller matrices can be reduced to PlanarNegative-k-Cycle on $O(\ell^2)$ nodes and $k = O(\ell)$. These instances can be then glued together to obtain a single graph on $N = O(n^3/\ell)$ nodes and $k = O(\ell)$. Conjecture 3.2 implies that there is no $O(Nk^{1-\epsilon})$ time algorithm for this case and establishes Theorem 1.4.

We proceed to the proof of Lemma 3.3. We first describe the auxiliary gadgets, then the main gadget, and then finally explain how to compose three copies of the main gadget to obtain the reduction.

The sequence gadget. Given a sequence $(a[i])_{i=0}^{s-1}$ and parameters M, ℓ, where (informally speaking) ℓ is a small constant and M is a very large weight, the sequence gadget $S(a, M, \ell)$ is constructed as follows. We create a directed path $v_0 \to v_1 \to \ldots \to v_{\ell \cdot s+2}$ with $\ell \cdot s + 3$ vertices. Then, for every $i = 0, 1, \ldots, s-1$ we add the edge (v_0, v_{i+1}) with weight $a[i]$. Finally, we set the weight of the edge (v_s, v_{s+1}) to be $-M$, and the weights of the remaining edges on the path to be 0. See Figure 2. We call v_0 the source and $v_{\ell \cdot s+2}$ the sink. It is easy to verify that, for every $i = 0, 1, \ldots, s-1$, there is a path from the source to the sink consisting of $\ell \cdot s - i + 2$ edges and total weight $a[i] - M$, and there are no other source-to-sink paths.

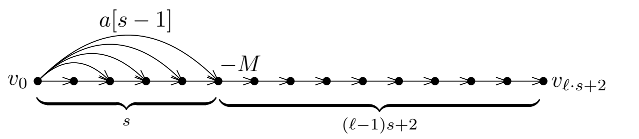

Figure 2: Sequence gadget.

The grid gadget. Given a matrix $A[1..n][1..n]$ consisting of sequences of length s and parameters M, M' and ℓ, where (informally speaking) ℓ is a small constant and $M \gg M'$, the grid gadget $G(A, M, M', \ell)$ is obtained as follows. We start with creating an $(n+1) \times (n+1)$ grid consisting of nodes $v_{i,j}$, for all $i, j = 0, 1, \ldots, n$ such that $i + j > 0$. For every $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, n-1$ we add an edge $(v_{i,j}, v_{i,j+1})$. For all $i, j = 1, 2, \ldots, n$ we add an edge $(v_{i,j}, v_{i-1,j})$. The nodes $s_i = v_{i,0}$ are called sources, and the nodes $t_j = v_{0,j}$ are called targets. Then, for all $i, j = 1, 2, \ldots, n$ we subdivide the edge $(v_{i,j-1}, v_{i,j})$ by inserting a new node $v'_{i,j}$, and similarly for all $i, j = 1, 2, \ldots, n$ we subdivide the edge $(v_{i,j}, v_{i-1,j})$ by inserting a new node $v''_{i,j}$. We set the weights of all edges to be 0. Finally, we insert a copy of the sequence gadget $S(A[i][j], M'', \ell)$ with $M'' = M + (i+j)M'$, identifying its source and target nodes with $v'_{i,j}$ and $v''_{i,j}$, respectively. See Figure 3.

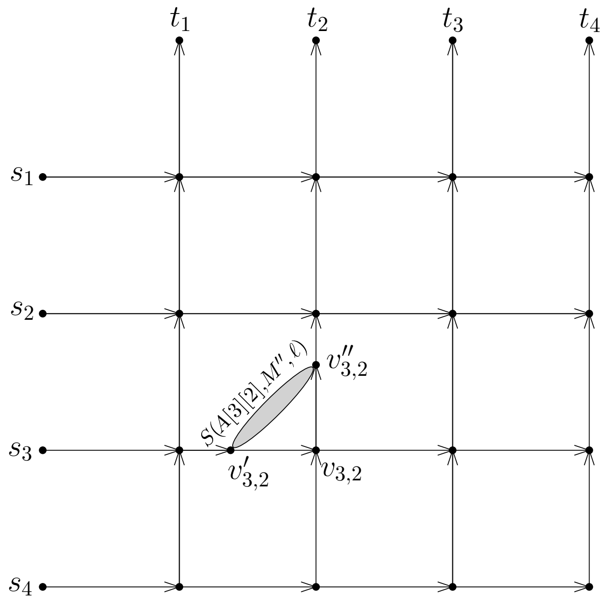

Figure 3: Grid gadget.

Consider a path from s_i to t_j. Such path is one of three types depending on the number of sequence gadgets visited by the path (where intersecting the gadget only at its source or sink is not considered to be visiting):

1. The path does not visit any sequence gadget and therefore consists of $2(i+j)$ edges and its total weight is 0.

2. The path visits exactly one sequence gadget and therefore consists of $2(i+j) + \ell \cdot s - k$ edges for some $k \in \{0, 1, \ldots, s-1\}$. Assume that the visited sequence gadget has been created for $v_{i',j'}$ (where $i' \le i$ and $j' \le j$ by the construction of the grid). The total weight of the path is therefore $-M-(i'+j')M' + A[i'][j'][k]$. Choosing a sufficiently large $M \gg M'$ thus guarantees that the total weight of such a path is either at least $-M-(i+j)M'+M'/2$ or equal to $-M-(i+j)M' + A[i][j][k]$.

3. The path visits at least two sequence gadgets and therefore consists of at least $2(i+j) + 2\ell \cdot s$ edges.

Composing the gadgets. Before we explain how to compose the gadget, we reformulate MinPlus-ConvMult as to make it more suitable for the reduction. First, by negating the entries, we obtain an equivalent formulation in which we are given matrices $A[1..n][1..n]$, $B[1..n][1..n]$ and $C[1..n][1..n]$ consisting of sequences of length s and we seek i, j, k, x, y such that $A[i][k][x] + B[k][j][y] + C[i][j][z] < 0$ and $z = x + y$. Second, by choosing a sufficiently large m and defining $A'[i][j][x] = A[i][j][x] + m \cdot x$, $B'[k][j][y] = B[k][j][y] + m \cdot y$, and $C'[j][i][z] = C[i][j][s-1-z] - m(s-1-z)$, we get an equivalent problem of finding i, j, k, x, y, z such that $A'[i][k][x] + B'[k][j][y] + C'[j][i][z] < 0$ and $s - 1 \le x + y + z$.

We now explain how to compose the gadgets. We create three grid gadgets $G(A', M_A, M, 4)$, $G(B', M_B, M, 2)$ and $G(C', M_C, M, 1)$, where $M_A \gg M_B \gg M_C \gg M$. We use s_i^A and t_i^A to the denote the i-th source and the i-th sink of the grid gadget $G(A, M_A, M, 4)$, and similarly for the other grid gadgets. Let $d = 12n + 6s + 2$. We connect t_i^A to s_i^B with a path consisting of $4(n-i) + d$ edges, and make the total weight of these edges $M_A + 2i \cdot M$. Similarly, we connect t_i^B to s_i^C with a path consisting of $4(n-i) + d$ edges, and make the total weight of these edges $M_B + 2i \cdot M$. Finally, we connect t_i^C to s_i^A with a path consisting of $4(n-i) + d$ edges, and make the total weight of these edges $M_C + 2i \cdot M$. See Figure 4.

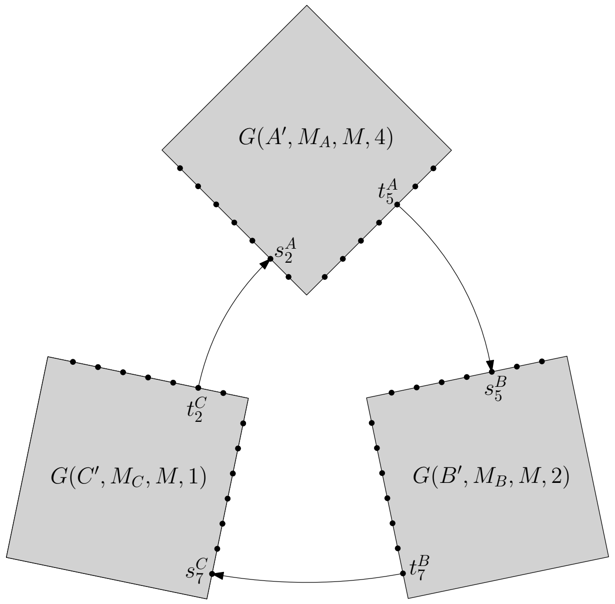

Figure 4: Overall structure of the reduction.

Consider a cycle consisting of at most k edges where $k = 12n + 6s + 1 + 3d$. Because $k < 4d$ and each path connecting a sink to its corresponding source in the next grid consists of at least d edges, any cycle consisting of at most k edges must have the following structure: There exist i, j, k such that the cycle consists of the following six parts:

1. a path from s_i^A to t_k^A in $G(A', M_A, M, 4)$ consisting of at least $2(i+k)$ edges,

2. the unique path from t_k^A to s_k^B consisting of $4(n-k)+d$ edges of total weight $M_A + 2k \cdot M$,

3. a path from s_k^B to t_j^B in $G(B', M_B, M, 2)$ consisting of at least $2(k+j)$ edges,

4. the unique path from t_j^B to s_j^C consisting of $4(n-j)+d$ edges of total weight $M_B + 2j \cdot M$,

5. a path from s_j^C to t_i^C in $G(C', M_C, M, 1)$ consisting of at least $2(j+i)$ edges,

6. the unique path from t_i^C to s_i^A consisting of $4(n-i)+d$ edges of total weight $M_C + 2i \cdot M$.

For a sufficiently large M_A, any negative cycle consisting of at most k edges needs to visit at least one sequence gadget inside $G(A', M_A, M, 4)$ in order to cancel out the M_A weight. Visiting at least two such sequence gadgets increases the number of edges on the cycle to at least $12n + 8s + 3d > k$. Therefore, any negative cycle consisting of at most k edges visits exactly one sequence gadget inside $G(A', M_A, M, 4)$. By repeating this reasoning on $G(B', M_B, M, 2)$ and $G(C', M_C, M, 1)$ we obtain that, in fact, such a cycle visits exactly one sequence gadget inside each grid gadget. Next, recall that the total weight of the part of the cycle inside $G(A', M_A, M, 4)$ is either at least $-M_A - (i+k)M + M/2$ or equal to $-M_A - (i+k)M + A'[i][k][x]$. Therefore, for a sufficiently large M, it must be equal to $-M_A - (i+k)M + A'[i][k][x]$ (and consist of $2(i+k) + 4s - x$ edges). By repeating this reasoning on $G(B', M_B, M, 2)$ and $G(C', M_C, M, 1)$ we conclude that the cycle consists of $12n + 7s - x - y - z + 3d$ edges and its total weight is $A'[i][k][x] + B'[k][j][y] + C'[j][i][z]$. The existence of such a negative cycle implies that $12n + 7s - x - y - z + 3d \le k$, which is equivalent to $s - 1 \le x + y + z$ and $A'[i][k][x] + B'[k][j][y] + C'[j][i][z] < 0$ as required. In the other direction, it is easy to verify that if i, j, k, x, y, z are such that $A'[i][k][x] + B'[k][j][y] + C'[j][i][z] < 0$ and $s - 1 \le x + y + z$ then there is a negative cycle (that passes through s_i^A, t_k^A, s_k^B, t_j^B s_j^C, and t_i^C) consisting of at most k edges. This concludes the proof of Lemma 3.3.

References

[1] Amir Abboud, Vincent Cohen-Addad, and Philip N. Klein. New hardness results for planar graph problems in P and an algorithm for sparsest cut. In *52nd STOC*, pages 996–1009, 2020.

[2] Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *57th FOCS*, pages 477–486, 2016.

[3] Amir Abboud, Shon Feller, and Oren Weimann. On the fine-grained complexity of parity problems. In *47th ICALP*, volume 168, pages 5:1–5:19, 2020.

[4] Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *26th SODA*, pages 1681–1697, 2015.

[5] Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. Better approximations for tree sparsity in nearly-linear time. In *28th SODA*, pages 2215–2229, 2017.

[6] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Saeed Seddighin, and Cliff Stein. Fast algorithms for knapsack via convolution and prediction. In *50th STOC*, pages 1269–1282, 2018.

[7] David Bremner, Timothy M. Chan, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, John Iacono, Stefan Langerman, Mihai Patrascu, and Perouz Taslakian. Necklaces, convolutions, and $X + Y$. *Algorithmica*, 69(2):294–314, 2014.

[8] Karl Bringmann. Approximating subset sum is equivalent to min-plus-convolution. *CoRR*, abs/1912.12529, 2019.

[9] Marek Cygan, Marcin Mucha, Karol Węgrzycki, and Michał Włodarczyk. On problems equivalent to $(\min, +)$-convolution. *ACM Trans. Algorithms*, 15(1):14:1–14:25, 2019.

[10] Jittat Fakcharoenphol and Satish Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *Journal of Computer and System Sciences*, 72:868–889, 2006.

[11] Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.

[12] Philip N. Klein, Shay Mozes, and Oren Weimann. Shortest paths in directed planar graphs with negative lengths: a linear-space $O(n \log^2 n)$-time algorithm. In *20th SODA*, pages 236–245, 2009.

[13] Marvin Künnemann, Ramamohan Paturi, and Stefan Schneider. On the fine-grained complexity of one-dimensional dynamic programming. In *44th ICALP*, volume 80, pages 21:1–21:15, 2017.

[14] Eduardo S. Laber, Wilfredo B. Roncalla, and Ferdinando Cicalese. On lower bounds for the maximum consecutive subsums problem and the $(\min, +)$-convolution. In *11th ISIT*, pages 1807–1811, 2014.

[15] Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *29th SODA*, pages 1236–1252, 2018.

[16] Richard J. Lipton and Robert E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Math*, 36:177–189, 1979.

[17] Shay Mozes and Christian Wulff-Nilsen. Shortest paths in planar graphs with real lengths in $O(n \log^2 n / \log \log n)$. In *18th ESA*, pages 206–217,

2010.

[18] James B. Orlin, K. Subramani, and Piotr J. Wojciechowski. Randomized algorithms for finding the shortest negative cost cycle in networks. *Discret. Appl. Math.*, 236:387–394, 2018.

[19] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, 2018.

[20] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018. Preliminary version in FOCS 2010.

[21] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018.

[22] Matthew D. Williamson and K. Subramani. On the negative cost girth problem in planar networks. *J. Discrete Algorithms*, 35:40–50, 2015.