SCIENCE CHINA
Information Sciences

• LETTER •

August 2024, Vol. 67, Iss. 8, 189101:1–189101:2
https://doi.org/10.1007/s11432-023-4080-3

On optimal streaming kernelization algorithms

Hao FENG[1], Wei YANG[1] & Jianer CHEN[2*]

[1]*School of Computer Science, Guangzhou University, Guangzhou* 510006*, China;*
[2]*Department of Computer Science and Engineering, Texas A&M University, College Station, TX* 77843*, USA*

Received 5 July 2023/Revised 7 March 2024/Accepted 18 June 2024/Published online 16 July 2024

The streaming model has been a popular model in big data computation. Streaming kernelization algorithms can be regarded as data compression processes on streaming data. In this study, we give a general method for developing computational lower bounds for streaming kernelization algorithms that is applicable to a large class of computational problems. As an example, we use the method to prove computational lower bounds for the well-known problem d-Hitting-Set. This result shows that a streaming kernelization algoroithm we recently developed for the famous NP-hard problem Vertex-Cover is optimal in all complexity measures, including space, update-time, and kernel size.

A parameterized problem Q is a decision problem with instances of form (x, k), where the parameter k is an integer. A kernelization algorithm K_Q of Q on an input (x, k) constructs (x_0, k_0) satisfying $|x_0|, k_0 \leqslant h(k)$ for a fixed function h, such that (x, k) is a yes-instance of Q iff (x_0, k_0) is a yes-instance of Q, where x_0 is the kernel. A deterministic context-sensitive problem (DCS) is a problem solvable by an $O(n)$-space deterministic algorithm. Note that many parameterized problems of interests are DCS.

Inputs of a streaming model are given as a stream of input elements. A streaming algorithm must follow the order of the element arrivals to process the input. The efficiency of a streaming algorithm is evaluated based on its space and update-time (i.e., the processing time per input element).

Theorem 1. A DCS parameterized problem Q has an $O(s(k))$-space bounded randomized streaming kernelization algorithm producing kernels of size $O(s(k))$ iff Q is solvable by an $O(s(k))$-space bounded randomized streaming algorithm with the same success probability.

Proof. ($\Leftarrow$) An $O(s(k))$-space randomized streaming algorithm A_Q solving the problem Q exactly gives the following streaming kernelization algorithm for Q: on a stream of an instance (x, k) of Q, (1) call the algorithm A_Q to decide in space $O(s(k))$ if (x, k) is a yes-instance; then (2) construct a proper trivial instance of size $O(1)$. This algorithm is obviously an $O(s(k))$-space bounded randomized streaming kernelization algorithm for Q that has the same success probability and constructs a kernel of size $O(1) = O(s(k))$.

($\Rightarrow$) An $O(s(k))$-space randomized streaming kernelization algorithm K_Q for the problem Q that constructs kernels of size $O(s(k))$ can be used to solve Q exactly, as follows: on a stream of an instance I, call the streaming kernelization algorithm K_Q to construct, in space $O(s(k))$, an equivalent instance I' of size $O(s(k))$, and store I' in space $O(s(k))$. Now we are able to solve the instance I' using also $O(s(k))$ space because Q is a DCS problem. This gives an $O(s(k))$-space randomized streaming algorithm that has the same success probability and solves the problem Q exactly.

Theorem 1 suggests a very effective method for developing lower bounds for streaming kernelization algorithms.

Corollary 1. If a DCS parameterized problem Q has space complexity $\Omega(s(k))$ for randomized streaming algorithms, then the problem Q has no streaming kernelization algorithms with the same success probability that run in $o(s(k))$ space and construct a kernel of size $o(s(k))$.

Lower bounds in space complexity for streaming algorithms have been developed recently for certain DCS problems [1], which, by Corollary 1, can help develop lower bounds for streaming kernelization algorithms. In the following, we extend the techniques in [1] and present some stronger space lower bounds on streaming algorithms for further DCS parameterized problems.

A set S is a d-set if S consists of exactly d elements. Let $\mathcal{C}$ be a collection of d-sets. A set H is a hitting set of size k for $\mathcal{C}$ if H consists of k elements such that for every d-set S in $\mathcal{C}$, $S \cap H \neq \emptyset$. Consider the following.

Parameterized d-Hitting-Set (p-dHS): given a collection $\mathcal{C}$ of d-sets and k, is there a hitting set H of size k for $\mathcal{C}$?

Our lower bounds will be derived based on the one-way communication mode, which consists of two randomized algorithms $\mathscr{A}$ and $\mathscr{B}$ [2]. To compute a 2-variable function $\phi(x, z)$, $\mathscr{A}$ is given the input x (but not z) and allowed to send $\mathscr{B}$ a single message $M(x)$, and $\mathscr{B}$ based on the input z (but not knowing x) and the message $M(x)$ from $\mathscr{A}$ computes the value $\phi(x, z)$. In this model, we measure the communication complexity of the protocol by the size of the message $M(x)$. The protocol works correctly with probability p if the output computed by $\mathscr{B}$ is equal to $\phi(x, z)$ with probability at least p for all x and z. Consider the well-known problem as given below.

The Index problem: The input to the protocol is (x, z), where $x = x_1x_2 \cdots x_n \in \{0, 1\}^n$ and $z \in \{1, 2, \ldots, n\}$, and compute the value $\phi(x, z) = x_z$.

Proposition 1 ([2]). For any constant $p > 1/2$, a communication protocol that solves Index with probability p must have communication complexity of $\Omega(n)$ bits.

We show how to use a streaming algorithm for problem p-

* Corresponding author (email: chen@cse.tamu.edu)

dHS to solve the INDEX problem. For an instance (x, z) of INDEX, where $x = x_1x_2\cdots x_n \in \{0,1\}^n$ and $z \in \{1, 2, \dots, n\}$, let $h = \lceil \sqrt[d]{n}\, \rceil$, and fix an injection π from $\{1, 2, \dots, n\}$ to the set $\{\langle b_1, b_2, \dots, b_d\rangle \mid 1 \leqslant b_i \leqslant h, 1 \leqslant i \leqslant d\}$ of ordered d-tuples. Suppose $\pi(z) = \langle a_1, a_2, \dots, a_d\rangle$.

Let $U = \{v_{i,b}, v_{i,b,t} \mid 1 \leqslant i \leqslant d, 1 \leqslant b \leqslant h, 1 \leqslant t \leqslant d\}$. Define a collection $\mathcal{C}_{x,z}$ of d-sets of U as follows:

(G1) For each $1 \leqslant i \leqslant d$ and each $b_i \neq a_i$, there is a d-set $\{v_{i,b_i,1}, \dots, v_{i,b_i,i-1}, v_{i,b_i}, v_{i,b_i,i+1}, \dots, v_{i,b_i,d}\}$ in $\mathcal{C}_{x,z}$. There are totally $d(h-1)$ d-sets in group (G1).

(G2) For each bit x_y of x such that $x_y = 1$, where $\pi(y) = \langle b_1, b_2, \dots, b_d\rangle$, there is a d-set $\{v_{1,b_1}, v_{2,b_2}, \dots, v_{d,b_d}\}$ in $\mathcal{C}_{x,z}$. The total number of d-sets in group (G2) is equal to the number of 1-bits in x.

A proof for Lemma 1 below can be found in Appendix A.

Lemma 1. The collection $\mathcal{C}_{x,z}$ has a hitting set of size $d(h-1)$ if and only if $x_z = 0$.

We are now prepared to present and prove the following theorem.

Theorem 2. Any randomized streaming algorithm that solves the P-dHS problem with probability p, where $p > 1/2$ can be any constant, uses space of $\Omega(k^d)$ bits.

Proof. Let $\mathcal{S}_{x,z}$ be a stream of d-sets for the collection $\mathcal{C}_{x,z}$ defined above, which is given by a sequence of d-sets in group (G2) (in arbitrary order), followed by a sequence of d-sets in group (G1) (in arbitrary order). Let $A_{\rm hit}$ be any randomized streaming algorithm for the problem P-dHS. Thus, the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, d(h-1))$ will decide if the collection $\mathcal{C}_{x,z}$ has a hitting set of size $d(h-1)$.

We construct a communication protocol with randomized algorithms $\mathscr{A}$ and $\mathscr{B}$ for the INDEX problem, as follows. On an instance (x, z) of INDEX, algorithm $\mathscr{A}$ takes the input x, generates all the d-sets in group (G2) for the collection $\mathcal{C}_{x,z}$ ($\mathscr{A}$ can do so because it knows which bit of x is 1), then runs the streaming algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, d(h-1))$ for the P-dHS problem on the generated d-sets in group (G2), until it reads the last d-set in group (G2). Then, algorithm $\mathscr{A}$ sends the memory contents $M(x)$ of its computation to algorithm $\mathscr{B}$. Upon receiving the message $M(x)$ from $\mathscr{A}$, algorithm $\mathscr{B}$ generates the d-sets in group (G1) ($\mathscr{B}$ can do so because it knows the value z so also the values $\pi(z) = \langle a_1, a_2, \dots, a_d\rangle$), and then uses the memory contents $M(x)$ of $\mathscr{A}$'s computation to continue the execution of the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, d(h-1))$, starting from the first d-set in the d-sets it generated for group (G1). Therefore, $\mathscr{B}$ will be able to complete the execution of the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, d(h-1))$. By Lemma 1, $\mathscr{B}$ will correctly conclude $x_z = 0$ if and only if the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, d(h-1))$ claims that the collection $\mathcal{C}_{x,z}$ has a hitting set of size $d(h-1)$. This gives the protocol for the INDEX problem, whose success probability is equal to that of the algorithm $A_{\rm hit}$ for the P-dHS problem.

Now, suppose that $A_{\rm hit}$ is any randomized streaming algorithm that solves P-dHS with probability p for a constant $p > 1/2$. Then, the above communication protocol solves INDEX with probability p. According to Proposition 1, in this case, the message $M(x)$ sent from $\mathscr{A}$ to $\mathscr{B}$ has size at least $\Omega(|x|) = \Omega(n)$ bits. Since the message $M(x)$ sent from $\mathscr{A}$ to $\mathscr{B}$ is the memory content of the execution of the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, k)$ for P-dHS, where $k = d(h-1)$, as a result, the algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, k)$ uses memory space of at least $\Omega(n)$ bits. Since $k = d(h-1)$, $h = \lceil \sqrt[d]{n}\, \rceil$, and d is a constant, we have $n = \Omega(k^d)$. Thus, the randomized streaming algorithm $A_{\rm hit}(\mathcal{S}_{x,z}, k)$ that solves the problem P-dHS takes space of at least $\Omega(k^d)$ bits. The proof of the theorem is now completed since $A_{\rm hit}$ is an arbitrary randomized streaming algorithm for the problem P-dHS.

The P-2HS problem is the famous parameterized VERTEX COVER problem (abbr. P-VC) that determines if a given graph has k vertices that can cover all edges of the graph. Moreover, it is easy to see that the problem P-dHS is DCS for each fixed d: on an instance $(\mathcal{C}, k)$ of P-dHS, simply enumerate all subcollections of k d-sets in $\mathcal{C}$ to check if any of them is a hitting set for the collection $\mathcal{C}$. Thus, by Theorem 2 and Corollary 1, we obtain the following lower bound for streaming kernelization algorithms for the problem P-VC.

Theorem 3. For any constant $p > 1/2$, there is no randomized streaming kernelization algorithm that solves the problem P-VC with probability p, runs in space $o(k^2)$, and constructs a kernel of size $o(k^2)$.

Our recent research [3] has given a streaming kernelization algorithm for the problem P-VC. The algorithm has space complexity $O(k^2)$ and update-time $O(1)$, and constructs kernels of size $O(k^2)$. By Theorem 3, this algorithm is optimal in all complexity measures, including space, update time, and kernel size.

We give a few remarks on our lower-bound results. Theorem 3 gives the first lower-bound result simultaneously on space complexity and kernel size for streaming kernelization algorithms for the problem P-VC. Chitnis et al. [1] presented a space lower bound $\Omega(k^2)$ for streaming algorithms that solve the problem P-VC, which does not imply Theorem 3 because solving a problem is obviously more difficult than kernelizing the problem. Moreover, Theorem 3 is not implied either by the lower-bound result in [4] that unless the polynomial-time hierarchy collapses, no deterministic polynomial-time kernelization algorithms can construct a kernel of size $O(k^{2-\epsilon})$ for the problem P-VC for any constant $\epsilon > 0$. The lower bound on kernel size for P-VC given in [4] relies on an unproved complexity theory conjecture (i.e., the polynomial-time hierarchy does not collapse), while our lower-bound result in Theorem 3 holds true unconditionally, i.e., without needing any complexity theory conjectures. In fact, our Theorem 3 presents a stronger lower bound for kernel size (on a more restricted space-bounded streaming model): (1) the lower bound holds true unconditionally; (2) the lower bound holds true for randomized algorithms; (3) our lower bound $\Omega(k^2)$ is strictly larger than the lower bound $\Omega(k^{2-\epsilon})$ —— there are functions such as $k^2/\log k$ that are not in $O(k^{2-\epsilon})$ for any constant $\epsilon > 0$ but are in $o(k^2)$; and (4) it excludes the possibility of superpolynomial time kernelization algorithms with $o(k^2)$ space complexity.

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chitnis R, Cormode G, Hajiaghayi M, et al. Parameterized streaming: maximal matching and vertex cover. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, 2015. 1234–1251

2 Roughgarden T. Communication complexity (for algorithm designers). FNT Theor Comput Sci, 2015, 11: 217–404

3 Chen J, Chu Z, Guo Y, et al. Space limited linear-time graph algorithms on big data. Theor Comput Sci, 2024, 993: 114468

4 Dell H, van Melkebeek D. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. J ACM, 2014, 61: 1–27