Exercise XI, Computational Complexity 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

Set-Intersection, and Applications

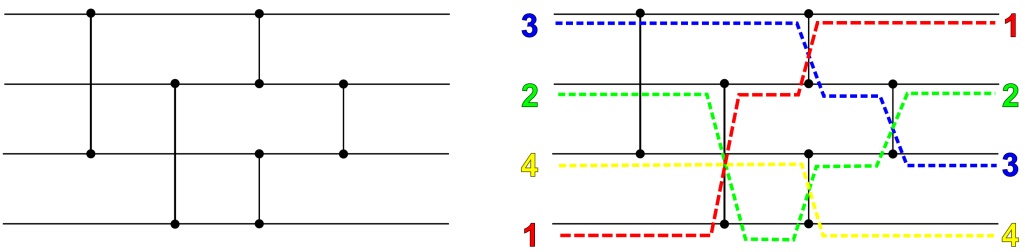
1 Consider the directed s–t connectivity function $\textsc{Conn}\colon \{0,1\}^{n\times n} \to \{0,1\}$ where the input $x \in \{0,1\}^{n\times n}$ is the adjacency matrix of a graph $G = ([n], E)$ (that is, $(i,j) \in E$ iff $x_{ij} = 1$) and $\textsc{Conn}(x) = 1$ iff this graph has a directed path from vertex $s = 1$ to vertex $t = n$. Note that $\textsc{Conn}$ is monotone. Show that the monotone KW game for $\textsc{Conn}$ can be solved with a $O(\log^2 n)$-bit deterministic protocol. (*Hint: What would Savitch do?*)

Solution: Let us suppose that Alice has an $s-t$ path $P = (s, v_2, v_3, \ldots, v_k, t)$ and Bob has no such $s-t$ path. Alice first sends $v_{k/2}$ (communicating $\log(n)$ bits). Notice that since Bob has no $s-t$ path in the graph represented by the input, it must be that it has no $s-v_{k/2}$ path or $v_{k/2}-t$ path. So Bob simply sends which parts it lacks (communicating 1 bit). Now, the problem is halved and Alice and Bob can simply repeat this trick until the path consists of a single edge which Alice holds and Bob not. Note that the communication cost is $\leq O(\log(k)\log(n)) \leq O(\log^2(n))$.

2 Recall Problem X-1: Alice has $A \subseteq [n]$, $|A| \geq n/2$, Bob has $B \subseteq [n]$, $|B| < n/2$, and they want to output $i \in A \setminus B$.

(a) Observe that this problem is essentially the monotone Karchmer–Wigderson game for the n-bit majority function $\textsc{Maj}_n\colon \{0,1\}^n \to \{0,1\}$

(b) A *sorting network*[1] consists of n wires together with *comparator gates* that given two input numbers (a,b) output the pair $(\min(a,b), \max(a,b))$. On input a list of n integers, the network outputs the integers in sorted order. Explain how a sorting network of *depth* d (maximum number of comparisons in any input-to-output path) can be used to construct a monotone circuit of depth d for $\textsc{Maj}_n$.



(picture credit: Wikipedia)

[1] `https://en.wikipedia.org/wiki/Sorting_network`

(c) The famous AKS network has depth $O(\log n)$. Assuming this, conclude that there is a $O(\log n)$-bit deterministic protocol for Problem X-1.

Solution:

(a) It the KW-game for $\textsc{Maj}_n$, Alice receives $x \in f^{(-1)}(1)$, i.e. $|x| \geq n/2$ and Bob receives $y \in f^{-1}(0)$, i.e. $|y| < n/2$. Their goal is then to find $i \in [n]$ such that $x_i > y_i$. We can see elements of $\{0,1\}^n$ as representing subsets of $[n]$ so that both problems are equivalent.

(b) Let S be a sorting network for n integers of depth d. Note that if we restrict the inputs of S to be 0 or 1 instead of integers, this sorting network can be re-expressed using only $\vee$ and $\wedge$ gates. Indeed, for any two $a, b \in \{0,1\}$, $\max(a,b) = a \vee b$ and $\min(a,b) = a \wedge b$. Thus, there exists a monotone circuit C of depth d that sorts the inputs bits. Finally, for any $x \in \{0,1\}^n$, $\textsc{Maj}_n(x) = 1$ if and only if $|x| \geq n/2$ if and only if $o_{n/2} = 1$ where o_i is the i-th output of C.

(c) Using the AKS network and the above characterization, we know that there exists a monotone circuit of depth $\log(n)$ for $\textsc{Maj}_n$. Since monotone circuit depth characterizes monotone KW games, it means that that monotone KW-game complexity of $\textsc{Maj}_n$ is $O(\log(n))$, we can then conclude using part a.

3 Consider the following problem: Alice holds a graph $G_A = ([n], E_A)$, Bob holds a graph $G_B = ([n], E_B)$, and their goal is to decide whether $G_A \cup G_B = ([n], E_A \cup E_B)$ is connected.

(a) Find a deterministic protocol of cost $O(n \log n)$ for this problem.

(b) Prove an $\Omega(n)$ lower bound by a reduction from set-intersection SI.

(*Formally, a reduction from SI_n to $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ consists of a pair of functions $A\colon \{0,1\}^n \to \mathcal{X}$ and $B\colon \{0,1\}^n \to \mathcal{Y}$ such that $SI_n(x,y) = f(A(x), B(y))$ for all x, y.*)

Solution:

(a) Alice can simply send a list of her connected components at cost $O(n \log(n))$ (basically a spanning forest) and Bob then computes if his part of the graph connects those connected components together or not.

(b) The transformed graph will have $k = 3n$ vertices and consists of three groups of vertices: $V_A = \{a_i\}_{i \in [n]}$, $V_b = \{b_i\}_{i \in [n]}$ and $V_C = \{c_i\}_{i \in [n]}$. The reduction works as follows (see Figure 1 for an example. Alice has all V_a connected by a single path and for any $i \in [n]$, if $x_i = 1$, then a_i is connected to m_i and if $x_i = 0$, then m_i is connected to b_i. Bob has all V_b connected by a single path and for each $i \in [n]$, if $y_i = 1$ then b_i is connected to m_i by an edge. Note that the resulting union of Alice and Bob's graphs has no isolated vertex and is connected if and only if there exists an index $i \in [n]$ with $x_i = y_i = 1$.

4 *Set-intersection is "*NP*-complete."* Let $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be any two-party function with non-deterministic communication complexity $\mathrm{N}_1(f) = k$. Show that f reduces to the set-intersection function SI_n with input length $n = 2^k$.

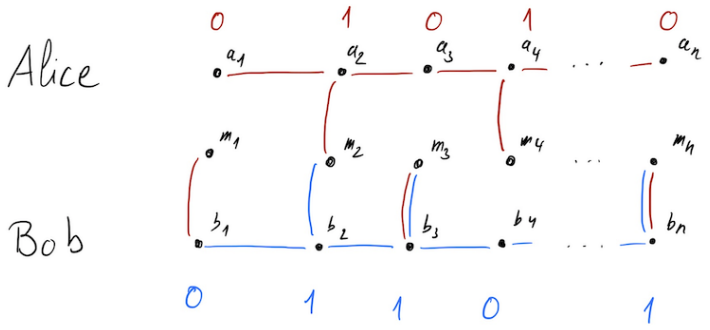


Figure 1. Reduction of set intersection to graph connectivity. Alice's input and edges are in red while Bob ones are in blue. In this example, the graph is connected and indeed both Alice and Bob share element 2.

Solution: Recall that $2^{\mathrm{N}_1(f)}$ is equal to the minimum number of rectangles needed to cover the 1-entries of M_f, the communication matrix of f. Now the reductions $A : \{0,1\}^n \to \{0,1\}^{2^k}$ and $B : \{0,1\}^n \to \{0,1\}^{2^k}$ are simply the indicator vector of which rectangle are hit by x and y. More precisely, let $(R_i)_{i\in[2^k]} \subseteq \{0,1\}^n \times \{0,1\}^n$ be a 1-cover of M_f and define A, B as:

$$A(x) = \{i \in [2^k] : x \text{ appears in } R_i\}$$
$$B(y) = \{i \in [2^k] : y \text{ appears in } R_i\}$$

Then, for any $x, y \in \{0,1\}^n$:

$$f(x,y) = 1 \iff \exists i \in [2^k] : (x,y) \in R_i \iff \mathrm{SI}_{2^k}(A(x), B(y))$$