Theory of Computation

Final Examination on January 13, 2015
Fall Semester, 2014

Problem 1 (20 points) Does **IP** contain all languages that have uniformly polynomial circuits?

Ans: Yes. **P** equals the class of languages with uniformly polynomial circuits. Furthermore, any language in **P** can be decided by an interactive proof system where the verifier simply decides the language itself and ignores the provers messages. So $\mathbf{P} \subseteq \mathbf{IP}$. ■

Problem 2 (30 points) Design a zero-knowledge proof protocol for 3 Colorability.

Ans: See pp. 695–696 of the slides. ■

Problem 3 (30 points) Suppose that there are n jobs to be assigned to m machines. Let t_i be the sorted running time for job $i \in \{1, \ldots, n\}$ in descending order, which means $t_1 > t_2 > \cdots > t_n$, A be an assignment where $A[i] = j$ means that job i is assigned to machine $j \in \{1, \ldots, m\}$, and $T[j] = \sum_{A[i]=j} t_i$ be the total running time for machine j. The makespan of A is the maximum time that any machine is busy, or

$$\text{makespan}(A) = \max_j T[j].$$

The Sorted Load Balance problem is to find the assignment which has minimal makespan over all assignments A, denoted by OPT. It is known to be NP-hard. Consider the following algorithm for Sorted Load Balance:

1: $T[j] \leftarrow 0$ for $j = 1, 2, \ldots, m$
2: **for** $i \leftarrow 1$ to n **do**
3: Let min be the j such that $T[j]$ is the smallest (with ties broken arbitrarily)
4: $A[i] \leftarrow$ min
5: $T[\text{min}] \leftarrow T[\text{min}] + t_i$
6: **end for**
7: **return** A

Show that this algorithm for Sorted Load Balance is an approximation algorithm which returns a solution that is at most $(3/2) \times OPT$. (You may use the fact that $T[j] - t_i \leq OPT$ for machine j and job i.)

Ans: Suppose $n \leq m$. Then the algorithm is trivially optimal and the claim holds. Now suppose $n > m$. Assume that the optimal algorithm assigns the first m jobs to distinct machines. Then job $m+1$ must be paired with one of the first m jobs. Recall that each of the first m jobs has a running time at least t_m. So $t_m + t_{m+1} \leq OPT$, and therefore $2t_{m+1} < OPT$, which implies

$$t_{m+1} < OPT/2.$$

On the other hand, assume that two (say i and j, where $i < j$) of the first m jobs are assigned by the optimal algorithm to the same machine. Then $t_i + t_j \leq OPT$. It implies that $2t_j < OPT$, which further implies $t_{m+1} \leq OPT/2$ because $t_{m+1} < t_j$.

Let machine j^* be the busiest machine after running our greedy algorithm. This means $T[j^*]$ equals the makespan, i.e., the largest among all $T[j]$s. Let i^* be the last job assigned to machine j^*. Suppose $i^* \leq m$. Then machine j^* has only one job i^* because each job of the first m jobs is assigned to distinct machines. Since t_1 is the largest running time among the first m jobs, this implies that $i^* = 1$ and $T[j^*] = t_1$. Recall that $t_i \leq OPT$ for all i. So $T[j^*] = t_1 = OPT$ and the claim holds. Now suppose $i^* > m$. Then $t_{i^*} < t_{m+1} \leq OPT/2$. By the hint, $T[j^*] \leq OPT + t_{i^*} \leq (3/2) \times OPT$. Hence, the claim is proved. ■

Problem 4 (20 points) Argue that if all monotone languages in P have polynomial monotone circuits, then P $\neq$ NP.

Ans: See p. 804 of the slides. ■