AN OPTIMIZATION PROBLEM CONCERNING CUSTOMER'S RANKING OF GOODS

THINH D. NGUYEN

ABSTRACT. We prove the NP-hardness of an optimization problem originating from algorithmic game theory. In this problem, we have a list containing each customer's ranking of some goods from a set U of items in a store. We want to select a subset $L \subseteq U$ of goods from which each customer will buy their highest ranked item. The objective is to maximize the sum of the rank values of bought items.

1. PROBLEM STATEMENT

The name we coin for our problem comes from the idea that, in our problem, we are trying to recommend the best items to our customers. This fact is reflected in the objective function, since we can regard the rank value of a bought item as the buyer's satisfaction.

Definition 1. (RECOMMENDED ITEMS): **Given:**

- ground set $U = [n] = \{1, \dots, n\}$ given by n ,
- m rankings given as total orders $\langle S_i, \sigma_i \rangle$ where $S_i \subseteq U$ ($1 \leq i \leq m$),
- weight vector for U given by $w \in \mathbb{R}^n$.

Goal: Find a subset $L \subseteq U$ maximizing the following sum:

$$(1) \quad r(L) = \sum_{i \in [m], S_i \cap L \neq \emptyset} w(t_i(L))$$

where $t_i(L)$ is the highest ranked item in $L \cap S_i$ according to σ_i .

Claim 2. *We have that 3-CNF-SAT $\leq_p$ RECOMMENDED ITEMS*

In this claim, the hardness of 3-CNF-SAT is known since [1]. Now, we prove our claim.

Proof. Suppose, we are given a 3-CNF-SAT instance consisting of: m DNF clauses $(\varphi_1, \dots, \varphi_m)$ over n variables $(x_1, \dots, x_n)$.

Description of our reduction: Create a set of items composed of two items for every variable: $x_i, \bar{x}_i$, corresponding to a *True* assignment to either the variable x_i or its negation, plus one auxiliary item t . Let the price of all items $\{x_i, \bar{x}_i\}_{i=1, \dots, n}$ be 1, and the price of t be 1.5.

Create two sets of consumers:

Set 1: (*Validity rankings*) this set of consumers will encode the validity constraints on the assignments to $x_1, \dots, x_n$. Namely, that exactly one out of every

Key words and phrases. optimization, ranking, set, subset.
Perebor.

$\{x_i, \bar{x}_i\}$ is set to *True* (i.e., taken by the algorithm). For every $i = 1, \dots, n$ create four partial rankings:

$$\sigma_{i1} : x_i \succ t$$

$$\sigma_{i2} : \bar{x}_i \succ t$$

$$\sigma_{i3} : x_i \succ \bar{x}_i$$

$$\sigma_{i4} : x_i \succ \bar{x}_i$$

Taking t can never decrease the payoff, so we assume that it's always chosen. If both x_i and $\bar{x}_i$ are selected, we get a payoff 4. If none of them is selected, we get a payoff of 3. If one of them is chosen, we get a payoff of 4.5.

Set 2: (*Clause rankings*) For each clause of the form $\varphi_j = \ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$, we create a ranking: $\sigma_j : \ell_{j1} > \ell_{j2} > \ell_{j3}$. If φ_j is satisfied, it means that at least one of the items corresponding to ℓ_{j1} , ℓ_{j2} , and ℓ_{j3} is selected, which gives an extra payoff of 1 from ranking σ_j .

Now, the 3CNF formula is satisfiable if and only if there is a set of items that gives a payoff of $m + 4.5n$. $\square$

2. CONCLUSION

As long as we study a mathematical conjecture, we should encourage ourselves of having enough labouring hours on popular maths books. Then, reading some articles on theory of computing like [3] is a good practice. Only after that, could we think of the ultimate final for all mathematics sciences.

REFERENCES

1. Michael R. Garey, David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*
2. David S. Johnson, *The NP-Completeness Column: An Ongoing Guide*. pp.393-405
3. Phan Dinh Dieu, Le Cong Thanh, Le Tuan Hoa, *Average Polynomial Time Complexity of Some NP-Complete Problems*, Theor. Comput. Sci. 46(3): pp.219-237 (1986)
Current address: Department of Mathematics, Moscow State University
Email address: kosmofarmer@yandex.com