This is a review submitted to Mathematical Reviews/MathSciNet.

Reviewer Name: Lin, Chuang-Chieh

Mathematical Reviews/MathSciNet Reviewer Number: 80385

Address:
Dept. of Comp. Sci. & Information Engineering
Tamkang University
Tamsui (New Taipei City)
TAIWAN (R.O.C.)
158778@mail.tku.edu.tw

Author: Bampis, Evripidis; Christou, Dimitris; Escoffier, Bruno; NGuyen, Kim Thang

Title: Online learning for min-max discrete problems.

MR Number: MR4471607

Primary classification: 68W27

Secondary classification(s): 68W20 68W25

Review text:

In general I like this note a lot. It encompasses a variety of topics, such as computational complexity (NP, RP, NP-complete, APX-hard, etc.), randomized and approximation algorithms, online learning, no regret (a.k.a. vanishing regret), (sub-)gradient descent with projection, follow the (regularized/perturbed) leader (FTL/FTRL/FTPL), as well as the min-max version of NP problems. Obviously, this article has a certain breadth and depth.

The authors study discrete nonlinear combinatorial optimization problems in an online setting. We can consider the vertex cover problem (VC) as an illustrating example. Given a graph $G = (V, E)$, a vertex cover of G is a vertex subset $V' \subseteq V$ such that every edge in E has at least one endpoint in V'. One can easily comprehend that any superset of V' is a feasible vertex cover. When each vertex $v \in V$ of is assigned with a nonnegative weight $w(v)$, the min-max version of VC is a kind of discrete nonlinear optimization problem which asks for a feasible solution V' such that the *maximum* vertex weight in V' is *minimized.* We should be aware that such a min-max version of an NP-hard problem is not always difficult, as mentioned by the authors. Indeed, given the graph G and the parameter k as the upper bound on the maximum vertex weight of the solution, one can just pick all the vertices of weight at most k and check whether it is a vertex cover. Let us denote such a set by W. If W is a vertex cover indeed, then we are done. If we would like to seek for a solution with smaller $k' \leq k$, we can check all vertex weights $\{w(v) \mid V\}$ and repeat the above procedure by dichotomic or binary searching, then the solution can be polynomially solved.

When we are allowed to solve the min-max version of VC iteratively, the problem turns out to be of online learning setting. The loss in each iteration is the maximum vertex weight of the computed feasible solution, hence it is discrete and non-convex. By iteratively updating the solution via gradient descent or FTRL plus projection onto the feasible set, if necessary, hopefully we can find the optimal solution of the min-max VC. In general, for online learning problems, one seeks for "no-regret", which means the average aggregated loss over time approaches that from an offline optimum (or we say the best in hindsight). Yet, there are problems which are hard to solve even for the offline setting. Hence, the relax notion of regret, called α-regret, is proposed as

$$R_T^\alpha = \sum_{t=1}^{T} \ell(x^t, y^t) - \alpha \sum_{t=1}^{T} \ell(x^*, y^*),$$

where ℓ is the loss function, x^t and y^t are the output of a (randomized) algorithm and state (or environmental feedback) at iteration t, respectively, and x^* is the offline optimal solution for minimizing the accumulated loss over T iterations. Hence, one can view the coefficient α here as a possible approximation ratio of the problem. The goal is to have an α-regret to be $R_T^\alpha = \mathrm{poly}(n) T^c = O(T^c)$, where $\mathrm{poly}(n)$ is a polynomial function of the input of size n and $0 \leq c < 1$ is a constant to represent the sublinear dependency of T. Note that the time horizon T can be much larger than n. To me, it reminds me that we usually neglect the computational complexity in generating the output of an onine learning algorithm, such as follow-the-leader, or best-response, etc., which could be computationally expensive. It might be natural that, to have a very small regret, one must be able to produce the solution which is very close to the offline optimum, for most of the time horizon. Hence, we can roughly come out an idea that it might not easy to have vanishing regret or α-regret for an NP-hard problem or an APX-hard problem.

As concluded by the author, I feel that this work has potential in extending the online learning framework to objectives other than min-max version of NP-problems. However, one must deal with the computation of the gradients or sub-gradients with reasonable constraint, such as Lipschitz continuity. To sum up, I like this work. Readers can even learn the background of a variety of topics due to the self-contained summary of introductions on their backgrounds. The deductions are very intuitive and straightforward, but the implication can have much impact.

Comments to the MR Editors (not part of the Review Text):
Sorry for the late submission of the review. I like this work and thanks for the invitation.