Max-Min Optimization for Lipschitz-Continuous Functions

Aurélien Delage[1], Olivier Buffet[2], Jilles-Steeve Dibangoye[1]

[1] Univ. Lyon, INSA Lyon, DISP, EA 4570, F-69100 Villeurbanne, France
{aurelien.delage,jilles-steeve.dibangoye}@insa-lyon.fr

[2] Univ. Lorraine, INRIA, CNRS, LORIA, F-54000, Nancy, France
{olivier.buffet}@loria.fr

Keywords : *Max-Min optimization, Lipschitz Optimization, Optimization over simplexes*

1 Introduction

For a given function $f : X \times Y \rightarrow \mathbb{R}$, the problem of computing $\max_x \min_y f(x, y)$ (referred to as a max-min problem) arises in various domains. For example, max-min problems are linked to Nash equilibria in game theory [2]. But it also models complex physics phenomena such as aerodynamic optimization [3]. However, if f is only Lipschitz-continuous, one can not use any method relying on gradient computations. As far as we know, no method exists to solve max-min problems for Lipschitz functions that guarantees a finite-time convergence property towards a global optimum. We propose to adapt Munos' Deterministic Optimistic Optimization (DOO) algorithm [1] to the max-min case.

2 Background

Let $f : X \rightarrow \mathbb{R}$ be a Lipschitz function defined over $X \subset \mathbb{R}^n$. DOO solves $\max_{x\in X} f(x)$ for a given error $\epsilon > 0$. It creates a non-uniform covering of the space definition with a finite number of areas where the function is controlled. Such areas, called subdivisions, are defined as balls of radius $\rho > 0$ centered in a point $x \in \mathbb{R}^n$ for a given norm $\|\cdot\|$. The Lipschitz property of f allows to upper-bound its value in any ball of radius ρ around a point $\tilde{x}$ by $f(\tilde{x}) + \lambda \cdot \rho$. The smaller the radius, the closer the upper-bound is to the values of f.

Let us assume that X is such that (i) there is an analytic way to create a first paving and (ii) each subdivision can be subdivided again. The algorithm (i-1) starts with the first paving of X, and (i-2) computes the upper-bound in every subdivision of the paving. Then, each of its iteration will consist in (ii-1) selecting the most promising subdivision according to the higher upper-bound value, (ii-2) subdividing it and (ii-3) computing the upper-bound values of each new subdivision. The algorithm is given in Algorithm 1 and Figure 1 gives an illustration of an application of DOO.

Algorithm 1: DOO

Fct DOO($\Big[\mathcal{D} \rightarrow \mathbb{R}\,;\, x \mapsto f(x)\Big], \epsilon,$)
  input : $f : \mathbb{R} \rightarrow \mathbb{R}$ is a λ-Lipschitz function
  Initialize $\mathcal{I}$ and $(R_i)_{i\in\mathcal{I}}$ s.t. $\mathcal{D} \subseteq \cup_{i\in\mathcal{I}} R_i$
  while *time* $\leq$ *budget* **do**
    $i^* \leftarrow \arg\max_{i\in\mathcal{I}} f(x_i) + r_i$
    Subdivide R_{i^*} into $\cup_{j\in\mathcal{I}^*} R_j \quad (\supseteq R_{i^*})$ $\forall j \in \mathcal{I}^*, x_j \leftarrow Center(R_j)$
    $\mathcal{I} \leftarrow [\mathcal{I} \setminus i^*] \cup \mathcal{I}^*$
  return $\langle \arg \,\&\, \max_{x_i} f(x_i)\rangle$

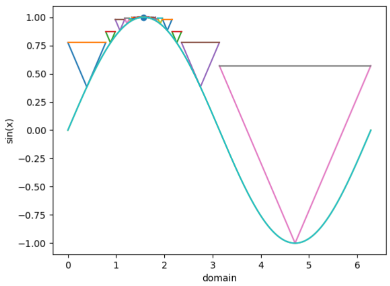

FIG. 1: Example of DOO applied to $x \mapsto sin(x)$ on $[0, 2\pi]$. The interval is tiled with subdivisions (*i.e.* here, intervals) of different size where the function *sin* is upper-bounded by Lipschitz cones summed up by an upper-bound constant. The cones along with the upper-bound form the triangle shapes.

As the algorithm iterates, the approximation error shrinks with the radius around the most promising areas. Thus, Munos was able to bound the difference between the highest value of f and DOO's returned value after running out of given budget.

3 Contributions

We are interested in computing $\max_{x \in X} \min_{y \in Y} f(x, y)$, where f is Lipschitz with finite time convergence guarantees. First, we prove that one can derive a finite-time convergent algorithm from DOO (contrary to bounding the error according to a given budget). The function $f_x : x \mapsto \min_y f(x, y)$ is also Lipschitz, so that one can use two nested DOO processes, *i.e.*, (1) an outer ϵ_1-optimal DOO maximizing the function $x \mapsto \min_y f(x, y)$, using the solution of (2) an inner ϵ_2-optimal DOO minimizing the function $y \mapsto f(x, y)$ for fixed x. We prove the finite-time convergence of this process towards an ϵ-optimal ($\epsilon = \epsilon_1 + \epsilon_2$) value by adapting Munos' proof. Finally, we show that one can analytically subdivide the probability simplex (which corresponds to mixed strategies in game theory) using $n-1$-dimensional hypercubes. Such a subdivision is analytical, which means that an algorithm has an explicit way of subdividing interesting areas of the simplex. We then derive a full algorithm to solve the Lipschitz optimization problem and evaluate it experimentally.

4 Conclusion

This papers proposes a method to deal with max-min Lipschitz optimization problems that require a finite-time convergence towards a global optimum guarantee. We focused on the optimization over simplexes, derived an algorithm and evaluated it. We hope our method can be useful whenever such a problem arises in various domains. Finally, let us point out that our approach can be straightforwardly generalized to any bi-level optimization problem, with any two different (Lipschitz) criteria $f^1(x, y)$ and $f^2(x, y)$.

References

[1] Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[2] John F Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[3] Zhili Tang and Lianhe Zhang. Nash equilibrium and multi criterion aerodynamic optimization. *Journal of Computational Physics*, 314:107–126, 2016.