THE COMBINATORIAL GENERATION OF OBJECTIVE TARGETS AND CONSTRAINTS FOR LARGE-SCALE TESTING OF OPTIMIZATION ROUTINES

ORIT DAVIDOVICH

Abstract. Our primary motivation is the large-scale testing and performance analysis of constrained optimization algorithms. To that end, we wish to randomly generate pairs (f, Ω) consisting of a continuous objective target f and a convex feasibility region Ω contained in its domain. Our challenge is to produce (f, Ω) in such a way that the true solution of the associated constrained optimization problem can be established combinatorially without recourse to an optimization algorithm.

1. Problem Statement

We wish to randomly generate pairs (f, Ω), where f is a continuous piecewise linear (PWL) function defined in a bounded, closed, convex domain $\mathrm{dom}(f) \subseteq \mathbb{R}^d$ and $\Omega \subseteq \mathrm{dom}(f)$ is a convex d-polytope. Our challenge is to produce (f, Ω) in such a way that the constrained optimization problem

$$\text{find } x^* \in \arg\min_{x \in \Omega} f(x) \tag{1}$$

can be resolved combinatorially without recourse to an optimization algorithm (similarly, for a maximization problem).

Our motivation is the large-scale testing and performance analysis of constrained optimization algorithms that seek to solve (1). Of particular interest to us are data-driven optimization algorithms that take a pair (D, Ω) of data D and constraints Ω as input. Such algorithms take an end-to-end approach that combines learning and optimization [7, 10, 11, 22, 23]. Given (f, Ω) and a hypothesis on the distribution of data points in $\mathrm{dom}(f)$, a pair (D, Ω) can easily be derived.

A continuous PWL function is in essence a combinatorial gadget. The domain of each of its affine pieces is a polytope. Subdividing polytopes into simplices, a continuous PWL function can be derived from a triangulation of its domain and set of real values, one for each triangulation vertex. The random generation of continuous PWL functions then becomes the random generation of triangulations [3, 6, 14, 15].

2. Triangulations

Let $S \subseteq \mathbb{R}^d$ be an n point configuration with $n \geq d+1$ and assume the affine span of S is $\mathbb{R}^d$. A full triangulation of S is an abstract d-dimensional *simplicial complex* $\mathcal{T}$ with vertex set S. The (finite) set of i-simplices of $\mathcal{T}$ is denoted by $\mathcal{T}^i$. A full triangulation satisfies $\mathcal{T}^0 = S$. The geometric realization of $\mathcal{T}$, defined as an abstract topological space, is denoted by $\|\mathcal{T}\|$.

Date: October 12, 2021.

We add a requirement that relates triangulations as abstract combinatorial gadgets to concrete embeddings in $\mathbb{R}^d$. We consider only those triangulations for which the following map $\iota : \|\mathcal{T}\| \to \mathbb{R}^d$ is an embedding: for each $\tau \in \mathcal{T}^i$ and $v \in \tau$, ι maps the vertex of the standard i-simplex $\|\tau\|$ labeled by v to v and restricts to an affine map on $\|\tau\|$. By abuse of notation, we use $\|\mathcal{T}\|$ also for the image of ι. Hence, we say that $\mathcal{T}$ is a *convex* triangulation when $\|\mathcal{T}\|$ is.

3. Piece-wise Linear Functions

Consider a pair $(\mathcal{T}, \mathcal{V})$ consisting of a triangulation $\mathcal{T}$ and a set $\mathcal{V} = \{r_v \in \mathbb{R} \,|\, v \in \mathcal{T}^0\}$. It gives rise to a continuous PWL function, which we will denote by $f_{(\mathcal{T},\mathcal{V})}$. Its domain is the bounded, closed $\mathrm{dom}(f) = \|\mathcal{T}\|$. To evaluate f at $v \in \mathrm{dom}(f)$, consider a simplex $\tau = \{v_1, \ldots, v_{d+1}\} \in \mathcal{T}^d$ such that $v \in \|\tau\|$ together with its associated values $r = (r_1, \ldots, r_{d+1})$ from $\mathcal{V}$. Define the homogenization of $v \in \mathbb{R}^d$ to be $\bar{v} = (v, 1)$. When simplices are non-degenerate, we are guaranteed a unique solution $\bar{x} \in \mathbb{R}^{d+1}$ to $\bar{A}_\tau \, \bar{x} = r$, where $\bar{A}_\tau$ is the matrix with rows $\{\bar{v}_1, \ldots, \bar{v}_{d+1}\}$, and we set $f(v) = \bar{x} \cdot \bar{v}$.

The random generation of continuous PWL functions $f_{(\mathcal{T},\mathcal{V})}$ is thus tantamount to the random generation of triangulations $\mathcal{T}$. There is a rich literature on triangulating polygons, polytopes, and point configurations [1, 4, 8, 9, 21]. The software TOPCOM defines the state of the art of triangulation generation and enumeration given a fixed point configuration [19, 13], and there are other triangulation generation schemes that produce optimal triangulations in some desired sense [2, 5]. Our purpose is different; we do not seek to solve the combinatorial problem of listing or enumerating all triangulations for a given point configuration, nor do we seek to produce optimal triangulations. Instead, our challenge is to generate sufficiently rich, that is, sufficiently random collections of continuous PWL functions.

In a computation environment, a simplex $\tau \in \mathcal{T}^d$ is numerically degenerate with respect to system tolerance tol, if it has effective zero volume, that is, $\mathrm{vol}(\tau) = |\det(\bar{A}_\tau)|/d! < \mathrm{tol}$. This prompts the need for some form of triangulation regularization [20]. Consider the set of volumes $\{\mathrm{vol}(\tau)/\mathrm{vol}(\mathcal{T})\}_{\tau \in \mathcal{T}^d}$ of a triangulation $\mathcal{T}$. It can be matched to a partition of $[0, 1]$, albeit not uniquely. Regardless of the partition produced, the distribution of the minimal volume will be equivalent to the distribution of the minimum in a random choice of $\#\mathcal{T}^d - 1$ points in $[0, 1]$, which is well known to be $\mathrm{Beta}(1, \#\mathcal{T}^d - 2)$. We formulate triangulation regularization as follows.

Definition 3.1 (Regularization Condition)**.** Given $\epsilon \in (0, 1)$, a triangulation $\mathcal{T}$ is subject to the ϵ-regularization condition if

$$\mathrm{CDF}_{\mathrm{MinVol}}(\min\{\mathrm{vol}(\tau)/\mathrm{vol}(\mathcal{T})\}_{\tau \in \mathcal{T}^d}) > \epsilon \tag{2}$$

where $\mathrm{MinVol} \sim \mathrm{Beta}(1, \#\mathcal{T}^d - 2)$.

4. Feasibility Region

We now turn to the generation of convex polytopes $\Omega \subseteq \mathrm{dom}(f)$, i.e., the feasibility region introducing constraints into the optimization problem

(1). Scenarios $\Omega \subsetneq \mathrm{dom}(f)$ are of particular interest when extending the generation of (f, Ω) to the generation of $(\mathcal{D}, \Omega)$. In real-world applications, governed by changing business consideration, historical business record may contain data points no longer considered feasible. Another reason to incorporate unfeasible data points into a synthetic data generation scheme is the testing of algorithms that seek to learn, perturb, or optimize the feasibility region itself.

Let $f = f_{(\mathcal{T},\mathcal{V})} : \|\mathcal{T}\| \to \mathbb{R}$ and consider a convex triangulation $\mathcal{T}_\circ$ that satisfies the property:

$$\forall \tau \in \mathcal{T}_\circ^d,\ \exists \tau' \in \mathcal{T}^d : \|\tau\| \subseteq \|\tau'\| \tag{3}$$

Setting $\Omega = \|\mathcal{T}_\circ\|$, we can easily solve (1), since

$$\arg\min_{x\in\Omega} f(x) = \arg\min_{v\in\mathcal{T}_\circ^0} f(v)$$

Given $\mathcal{T}$, our goal is then to produce $\mathcal{T}_\circ$ that satisfies Property (3). Generating f and Ω separately, and then producing $\mathcal{T}_\circ$ via intersection $\{\Omega \cap \|\tau\|\}_{\tau\in\mathcal{T}}$ is akin to solving multiple LP problems, since the intersection of affine hyperplanes must be constrained [16, 17]. This defeats the purpose of our combinatorial generation scheme. We therefore require a different approach.

5. Algorithm

Our proposed algorithm melds the generation of f and Ω with successive ϵ-regularized pulling and placing (or pushing) of new vertices. Crucially, the triangulation, updated at each incremental step, is maintained hierarchically.

Definition 5.1 (DAG of Simplices)**.** A DAG $i, t : E \to V$ of d-simplices has nodes $\tau = \{v_1, \ldots, v_{d+1}\} \subseteq \mathbb{R}^d$ in general position and strict inclusion of geometric realizations $\|\tau\| \subsetneq \|\tau'\|$ for edges $e = (\tau', \tau) \in E$ with $i(e) = \tau', t(e) = \tau$.

Our algorithm takes as input: (i) a random variable X over $\mathbb{R}^d$ to sample triangulation vertices, (ii) a random variable y over $\mathbb{R}$ to sample their respective values, (iii) an integer $n \geq d+1$ for the number of triangulation vertices, (iv) $p \in [0, 1]$ for Bernoulli trials, and (v) $\epsilon \in (0, 1)$ for regularization. It produces a list $L = [G_1, \ldots, G_N]$ of DAGs of d-simplices and a dictionary $\mathcal{V}$ of values, one for each unique vertex of some node $\tau \in V(G_i)$. The list L is constructed so that, for every $1 \leq m \leq N$, we have a convex triangulation $\mathcal{T}_m$ with $\mathcal{T}_m^d = \cup_{i\leq m}\mathrm{leaves}(G_i)$. In addition, $\mathcal{T}_m$ is a subtriangulation of $\mathcal{T}_{m'}$ for $m \leq m'$. The sequence $[\mathcal{T}_1, \ldots, \mathcal{T}_N]$ allows us to establish Theorem 5.2, which delivers what we have set out to do.

Theorem 5.2. *For $\rho \in [0, 1]$ and input $n \geq d + 1$, Algorithm 1 gives rise to (f, Ω) and $v_{\min}, v_{\max} \in \Omega$, where f is a continuous PWL function defined over a bounded, closed, convex domain $dom(f) \subseteq \mathbb{R}^d$ and $\Omega \subseteq dom(f)$ is a convex d-polytope, such that,*

(i) $vol(\Omega)/vol(dom(f)) \geq \rho$,
(ii) $\min_\Omega f = f(v_{\min})$ *and* $\max_\Omega f = f(v_{\max})$,
(iii) the solutions $v_{\min}, v_{\max}$ *are produced in* $\mathcal{O}(n)$.

Algorithm 1 PWL Functions with Constraints

Input: X r.v. in $\mathbb{R}^d$, y r.v. in $\mathbb{R}$, $n \geq d+1$, $p \in [0,1], \epsilon \in (0,1)$
Output: $(L = [G_1, \ldots, G_N], \mathcal{V})$

$\tau \leftarrow$ sample $\{v_1, \ldots, v_{d+1}\}$ from X
$\mathcal{V} \leftarrow \{v_i : r_i\}_{i=1}^{d+1}$, r_i sampled from y
$L \leftarrow [\text{DAG}(\text{roots} = \tau)]$
$i \leftarrow 0$
while $i < n - (d+1)$ **do**
  sample c from $\text{Bern}(p)$
  if $c = 0$ **then**
    sample v from X
    if $\exists G \in L : v \in G$ **then**
      $\tau \leftarrow$ minimal in G w.r.t. v
      $\tau_v \leftarrow$ subdivide τ w.r.t. v
      if $\textsc{RegCond}(\tau_v, \epsilon)$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} + \{v : \text{ sample } y\}$
        $\text{children}(\tau) \leftarrow \tau_v^d$
        $i \leftarrow i + 1$
      end if
    else
      $\text{facets}(v) \leftarrow \{F \,|\, F \text{ facet of } \text{Conv}(L) \text{ visible to } v\}$
      $\text{nodes}(F) \leftarrow \{\tau \,|\, \exists G \in L : \tau \in G \text{ minimal w.r.t } F\}$
      $\text{leaves}(F) = \cup_{\tau \in \text{nodes}(F)} \text{leaves}(\tau)$
      $\text{facets}(F) \leftarrow \{F' \,|\, \exists \varsigma \in \text{leaves}(F) : F' \in \varsigma^{d-1}, \|F'\| \subseteq \|F\|\}$
      $\tau_{F'} \leftarrow F' \cup \{v\}, F' \in \text{facets}(F), F \in \text{facets}(v)$
      if $\textsc{RegCond}(\text{leaves}(F) \cup \{\tau_{F'}\}_{F' \in \text{facets}(F)}, \epsilon), \forall F \in \text{facets}(v)$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} + \{v : \text{ sample } y\}$
        $L \leftarrow L + [\text{DAG}(\text{roots} = \{\tau_{F'} | F' \in \text{facets}(F), F \in \text{facets}(v)\})]$
        $i \leftarrow i + 1$
      end if
    end if
  else
    sample $v \in \cup_i \|G_i\|^{d-1}$
    $G_v \leftarrow \{G_i \,|\, v \in G_i\}$
    $\min(G_v) \leftarrow \{\tau \,|\, \exists G \in G_v : \tau \in G \text{ minimal w.r.t. } v\}$
    $\tau_v \leftarrow$ subdivide τ w.r.t. v, $\forall \tau \in \min(G_v)$
    if $\textsc{RegCond}(\tau_v, \epsilon), \forall \tau \in \min(G_v)$ **then**
      $\mathcal{V} \leftarrow \mathcal{V} + \{v : \text{ sample } y\}$
      $\text{children}(\tau) \leftarrow \tau_v^d, \forall \tau \in \min(G_v)$
      $i \leftarrow i + 1$
    end if
  end if
end while

References

1. D. Avis and H. ElGindy, *Triangulating simplicial point sets in space*, Proceedings of the Second Annual Symposium on Computational Geometry (New York, NY, USA), SCG '86, Association for Computing Machinery, 1986, pp. 133–141.
2. C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, *The quickhull algorithm for convex hulls*, ACM Trans. Math. Softw. **22** (1996), no. 4, 469–483.
3. P. Caputo, F. Martinelli, A. Sinclair, and A. Stauffer, *Random lattice triangulations: Structure and algorithms*, Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '13, Association for Computing Machinery, 2013, pp. 615—-624.
4. B. Chazelle, *Triangulating a simple polygon in linear time*, Discrete & Computational Geometry **6** (1991), no. 3, 485–524.
5. B. Delaunay, *Sur la sphère vide. a la mémoire de georges voronoï*, Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na **6** (1934), 793–800.
6. M. Demirtas, L. McAllister, and A. Rios-Tascon, *Bounding the kreuzer-skarke landscape*, arXiv preprint arXiv:2008.01730 (2020).
7. P. L. Donti, B. Amos, and J. Z. Kolter, *Task-based end-to-end model learning in stochastic optimization*, Proceedings of the 31st International Conference on Neural Information Processing Systems (Red Hook, NY, USA), NIPS'17, Curran Associates Inc., 2017, pp. 5490–5500.
8. H. Edelsbrunner, *Algorithms in combinatorial geometry*, Monographs in Theoretical Computer Science. An EATCS Series, vol. 10, Springer-Verlag Berlin Heidelberg, 1987.
9. H. Edelsbrunner, F. P. Preparata, and D. B. West, *Tetrahedrizing point sets in three dimensions*, Symbolic and Algebraic Computation (Berlin, Heidelberg) (P. Gianni, ed.), Springer Berlin Heidelberg, 1989, pp. 315–331.
10. A. N. Elmachtoub and P. Grigas, *Smart "predict, then optimize"*, arXiv preprint arXiv:1710.08005 (2020).
11. F. Fang, T. Nguyen, R. Pickles, W. Lam, G. Clements, B. An, A. Singh, M. Tambe, and A. Lemieux, *Deploying PAWS: Field optimization of the protection assistant for wildlife security*, Proceedings of the Twenty-Eighth AAAI Conference on Innovative Applications of Artificial Intelligence (Palo Alto, CA, USA), IAAI'16, The AAAI Press, 2016, pp. 3966–3973.
12. J. E. Goodman, J. O'Rourke, and C. D. Tóth (eds.), *Handbook of discrete and computational geometry*, 3rd ed., Chapman and Hall/CRC, 2017.
13. C. Jordan, M. Joswig, and L. Kastner, *Parallel enumeration of triangulations*, The Electronic Journal of Combinatorics **25** (2018), no. 3, P3.6.
14. L. McShine and P. Tetali, *On the mixing time of the triangulation walk and other catalan structures*, in Pardalos et al. [18], pp. 147–160.
15. M. Molloy, B. Reed, and W. Steiger, *On the mixing rate of the triangulation walk*, in Pardalos et al. [18], pp. 179–190.
16. D. M. Mount, *Geometric intersection*, in Goodman et al. [12], pp. 1113–1134.
17. D. E. Muller and F. P. Preparata, *Finding the intersection of two convex polyhedra*, Theoretical Computer Science **7** (1978), no. 2, 217–236.
18. P. Pardalos, S. Rajasekaran, and J. Rolim (eds.), *Randomization methods in algorithm design*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 43, DIMACS-AMS, 1999.
19. J. Rambau, *TOPCOM: Triangulations of point configurations and oriented matroids*, Mathematical Software (A. M. Cohen, X.-S. Gao, and N. Takayama, eds.), World Scientific, 2002, pp. 330–340.
20. J. Ruppert, *A delaunay refinement algorithm for quality 2-dimensional mesh generation*, Journal of Algorithms **18** (1995), no. 3, 548–585.
21. R. Seidel, *A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons*, Computational Geometry **1** (1991), no. 1, 51–64.

22. H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, *COPE: Traffic engineering in dynamic networks*, Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (New York, NY, USA), SIGCOMM '06, Association for Computing Machinery, 2006, pp. 99—-110.
23. Bryan Wilder, Bistra Dilkina, and Milind Tambe, *Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 1658–1665.

IBM Research Lab, Haifa, Israel
E-mail address: orit.davidovich@ibm.com