Mathematical Programming manuscript No.
(will be inserted by the editor)

J. Guérin · P. Marcotte · G. Savard

An Optimal Adaptive Algorithm for the Approximation of Concave Functions ⋆

Received: date / Revised version: date

Abstract. We consider the piecewise linear approximation of a concave function representing the value function of a parameterized convex program. Using dynamic programming, we derive a procedure for selecting the knots at which an oracle provides the function value and one supergradient. The procedure is adaptive in that the choice of a knot is dependent on the choice of the previous knots. It is also optimal in that the approximation error, in the integral sense, is minimized in the worst case.

Key words. Dynamic programming. Approximation. Adaptive algorithm.

1. Introduction

The present work is motivated by a bicriterion network equilibrium problem modelled as a variational inequality (see Marcotte and Zhu [5]). In the linearization algorithm whose implementation is discussed in Marcotte, Nguyen and Tanguay [6], parametric shortest path problems have to be solved repeatedly. Since this is computationally costly, it is natural to consider the approximation of the value function defined by this parametric program by a piecewise linear function involving a small number of evaluation points (knots). In order to be consistent with the stopping criterion used in the linearization algorithm, the quality of the approximation has to be measured in the integral sense, i.e., with respect to the L_1 norm. This yields the problem of selecting the knots such as to minimize the approximation error, in the worst case.

In a general setting, consider a proper, concave function f defined over the interval $[0,1]$, normalized such that $f(0) = 0$ and $f(1) = 1$. At each point $\bar{x} \in (0,1)$ an oracle gives the value $f(\bar{x})$ and that of a supergradient $\xi \in \partial f(\bar{x})$, i.e., a point satisfying the inequality

$$f(x) \leq f(\bar{x}) + \xi(x - \bar{x}) \qquad \forall x \in [0,1].$$

J. Guérin: Département de mathématiques et génie industriel, École Polytechnique, C.P. 6079, succursale Centre-ville Montréal (Québec) H3C 3A7

P. Marcotte: CRT and DIRO, Université de Montréal, CP 6128, succursale Centre-Ville, Montréal, Canada H3C 3J7

G. Savard: GERAD and Département de mathématiques et génie industriel, École Polytechnique, C.P. 6079, succursale Centre-ville, Montréal (Québec) H3C 3A7

Mathematics Subject Classification (1991): 20E28, 20G40, 20C20

⋆ This work was partially supported by NSERC (Canada) and FCAR (Québec)

In what follows we will denote an arbitrary supergradient ξ by $f'(\bar{x})$, even when f is not differentiable at $\bar{x}$. From this information we derive the obvious under and over-approximations of f over the interval (0,1):

$$L(t) = \min\left\{\frac{f(\bar{x})}{\bar{x}}t, \frac{1-f(\bar{x})}{1-\bar{x}}(t-\bar{x}) + f(\bar{x})\right\}$$
$$U(t) = f(\bar{x}) + f'(\bar{x})(t-\bar{x})$$

which provide the error bound $\int_0^1 (U(t) - L(t))\, dt$. We propose to minimize the above error term through a sequential selection procedure for the knots, and prove that this procedure is optimal in the sense that it minimizes the error term in the worst case. The procedure is adaptive: the selection of a knot is dependent on the choice of the previous knots.

Novak [7] and Sonnevend [9] have shown that, for the problem of approximating the integral of a convex function using function values and derivatives, adaption does not improve the accuracy of the approximation with respect to passive algorithms, in the worst case. This integration problem is equivalent to ours and therefore adaption does not help here either. However, since the worst case is unlikely to occur in practice, an adaptive algorithm might be able to take advantage of favorable information to yield an improved approximation. This led Sukharev [10] to the definition of *sequentially optimal* algorithms, i.e., adaptive algorithms that make optimal use, at each step, of available information.

The algorithm that we propose in section 4 is not truly sequentially optimal since, although evaluation points are chosen according to previous information, they are selected in a left-to-right fashion. This may seem like a severe restriction, as would any other fixed ordering of the points, but since adaption does not help in the worst case no ordering can guarantee an better accuracy. The same is true for any other more complex scheme to determine the order in which the evaluation points are chosen. This does not mean that adaptive methods are not useful for our problem, but that the advantage of such methods can be only be seen in practice.

We have chosen the left-to-right order for simplicity. This gives an approximation method that may be less efficient in practice than a sequentially optimal algorithm, but such algorithms typically have high computational complexity, offsetting accuracy gain. In contrast, our algorithm has low complexity, namely $\mathcal{O}(n)$, where n denotes the number of evaluation points. This is the same as the complexity of Sonnevend's optimal passive algorithm.

Approximation algorithms based on the bounding functions L and U have been studied in the literature under the name of "sandwich algorithms", the difference between L and U being measured with respect to the uniform, L_1 or Hausdorff norm (the Hausdorff distance between the graphs of the functions L and U). At a given iteration of a sandwich algorithm, a knot that lies in the interval of largest estimated error is determined. Fruhwirth, Burkard and Rote [3] propose, in the case of the Hausdorff distance, three subdivision rules that achieve the optimal asymptotic bound $\mathcal{O}(n^{-2})$. A bound of the same order was also obtained by Burkard, Hamacher and Rote [2] for the uniform norm. In

Fig. 1. Upper and lower approximation of the function f ($n = 2$)

the paper by Rote [8] four subdivision rules are studied both from a theoretical and numerical point of view. However, two of the subdivision methods require, at each iteration, the solution of an optimization problem involving the function f itself; this violates a condition of our problem which states that no more than n function evaluations must be performed. Indeed, Yang and Goh [11] showed that, if f is easy to compute, the sandwich algorithm can dispense altogether with first-order (derivative or supergradient) information.

In discussing optimal sandwich methods, Rote mentions the problem of determining an evaluation strategy that minimizes the maximal error. Our analysis brings a partial answer to this problem and improves upon previous works in two important respects:

- We obtain both the optimal convergence rate and optimal selection rules for each n.
- Our result is parameter-free: no a priori information about the function to be approximated is required.

2. Problem definition

Let f be a proper concave function defined over the unit interval $[x_0, x_{n+1}] = [0, 1]$ and normalized so that $f(0) = 0$ and $f(1) = 1$. We wish to sequentially select n points $x_1, \ldots, x_n$ in order to minimize the measure

$$\mathcal{E}(x_1, \ldots, x_n) = \frac{1}{2} \int_0^1 [U(t) - L(t)]\, dt \tag{1}$$

where (see Figure 1)

$$L(t) = \min_{i=1,\ldots,n+1} \left\{ f(x_{\sigma(i)-1}) \frac{(t - x_{\sigma(i)})}{x_{\sigma(i)-1} - x_{\sigma(i)}} + f(x_{\sigma(i)}) \frac{(t - x_{\sigma(i)-1})}{x_{\sigma(i)} - x_{\sigma(i)-1}} \right\}$$

$$U(t) = \min_{i=0,\ldots,n+1} \{ f(x_i) + f'(x_i)(t - x_i) \}$$

and σ is the permutation that reorders the knots and the two endpoints from left to right:

$$0 = x_{\sigma(0)} < x_{\sigma(1)} < \ldots < x_{\sigma(n)} < x_{\sigma(n+1)} = 1.$$

Consider a class of functions $F \subset C[0, 1]$. Let $\mathcal{E}_n$ denote the worst-case error corresponding to an optimal selection of n knots ($n > 0$). More precisely, let A be the class of all algorithms that construct the approximation $\frac{U+L}{2}$ using the information $(f(x_1), f'(x_1), \ldots, f(x_n), f'(x_n))$ obtained by evaluating the function $f \in F$ and one of its supergradients at n points $x_1, \ldots, x_n$ of $[0, 1]$. We consider

here deterministic adaptive algorithms, where the choice of the point x_i may depend on the previous information $x_1, f(x_1), f'(x_1), \ldots, x_{i-1}, f(x_{i-1}), f'(x_{i-1})$.

Denote the application of $\alpha \in A$ to $f \in F$ by $\alpha(f)$. The optimal worst-case error for the approximation of functions from F in the L_1 norm is defined to be

$$\mathcal{E}_n(a,b) = \inf_{\alpha \in A} \sup_{f \in F} || f - \alpha(f) ||_1.$$

It is not difficult to show that the supremum in the above expression is exactly the integral on the right-hand side of (1).

In the terminology of Sukharev [10], we consider adaptive algorithms of the form $\alpha = (N, \phi)$, where the *information operator* is

$$N(f) = (f(x_1), f'(x_1), \ldots, f(x_n), f'(x_n))$$

and the *terminal operation* ϕ is defined by

$$\phi(N(f)) = \frac{U+L}{2}.$$

It can be shown that ϕ is *central* and thus optimal in the sense that for each f, and $N(f)$ already computed, it minimizes

$$\sup_{\tilde{f} \in F_f} || \tilde{f} - \alpha(f) ||_1,$$

where F_f is the subset of functions $\tilde{f}$ in F with $\tilde{f}(x_i) = f(x_i)$ and $\tilde{f}'(x_i) = f'(x_i)$ for all i.

The information operator N described above is imposed by the information available for our problem and the choice of the particular terminal operation ϕ is justified by its optimality. Therefore, the construction of an approximation algorithm reduces here to the choice of the n evaluation points $x_1, \ldots, x_n$. The optimal worst-case error $\mathcal{E}_n$ and the optimal evaluation points can be computed through the recursion

$$\mathcal{E}_{n-k}(z^k) = \min_{x_{k+1} \in (0,1)} \max_{(f(x_{k+1}), f'(x_{k+1})) \in C} = \mathcal{E}_{n-(k+1)}(z^{k+1})$$

$$k = 0, \ldots, n-1$$

where

$$z^k = \big(x_1, \ldots, x_k, f(x_1), \ldots, f(x_k), f'(x_1), \ldots, f'(x_k)\big)$$

and C represents the set of constraints that must be satisfied by the values of $f(x_{k+1})$ and $f'(x_{k+1})$ in order to be compatible with the first k functional and supergradient values of the concave function f.

The above system is in most likelihood too complex to be reduced to a closed form expression. For this reason, we limit our analysis to the identity permutation, i.e., the knots will be determined in a left-to-right fashion.

Fig. 2. Case $n = 1$.

Let $a = f'(0)$, $b = f'(1)$ and denote by $\mathcal{E}_n(a, b)$ the optimal worst-case error when knots are selected in a from left to right. By definition $\mathcal{E}_0(a, b)$ is equal to the area of the triangle OPT in Figure 2, i.e.

$$\mathcal{E}_0(a, b) = \frac{1}{2}\frac{(1-b)(a-1)}{a-b}.$$

In the case where $n = 1$, let x denote the evaluation point and set $v = f(x)$, $\mu = f'(x)$ [1] . Since the graph of f is entirely contained within the triangle OPT of Figure 2, the following requirements must be met by v and μ:

$$\begin{aligned} &x \le v \le ax && \text{if} \quad && x \in [0, \tfrac{1-b}{a-b}] \\ &x \le v \le bx + 1 - b && \text{if} \quad && x \in [\tfrac{1-b}{a-b}, 1] \\ &\frac{1-v}{1-x} \le \mu \le \frac{v}{x}. \end{aligned} \tag{2}$$

The error bound $\mathcal{E}_1(a, b)$ corresponds to the sum of the areas of the triangles OML and LNT of Figure 2 and can be expressed in term of $\mathcal{E}_0$:

$$\mathcal{E}_1(a, b) = \min_{x\in(0,1)} \max_{v} \max_{\mu} \left\{ xv\mathcal{E}_0\left(\frac{x}{v}a, \frac{x}{v}\mu\right) + (1-x)(1-v)\mathcal{E}_0\left(\frac{1-x}{1-v}\mu, \frac{1-x}{1-v}b\right)\right\},$$

where v and μ must satisfy the geometric constraints (2), and the scaling factors multiplying $\mathcal{E}_0$, a, b and μ are obtained by elementary geometric arguments. Now, for an arbitrary number n, the worst-case error term can be defined recursively as

$$\mathcal{E}_n(a, b) = \min_{x\in(0,1)} \max_{v} \max_{\mu} \left\{ xv\mathcal{E}_0\left(\frac{x}{v}a, \frac{x}{v}\mu\right) + (1-x)(1-v)\mathcal{E}_{n-1}\left(\frac{1-x}{1-v}\mu, \frac{1-x}{1-v}b\right)\right\} \tag{3}$$

with constraints (2). Any minimizer x of $\mathcal{E}_n(a, b)$ is called *optimal*. The next point of evaluation is then set to a minimizer of $\mathcal{E}_{n-1}\left(\frac{1-x}{1-f(x)}f'(x), \frac{1-x}{1-f(x)}b\right)$, and so on to the nth knot. Our main result follows.

[1] From now on, we drop knot indices, with the exception of Figure 2, where it has been retained in order to avoid a collision of the symbol x with itself.

Fig. 3. The function ϕ (case $\mu_{\max} = \mu^+$)

Theorem 1. *The optimal worst-case error is equal to*

$$\mathcal{E}_n(a,b) = \frac{1}{2(n+1)^2}\frac{(a-1)(1-b)}{(a-b)} \tag{4}$$

and the minimum in (3) is achieved at the point

$$x^* = \frac{1}{(n+1)^2}\left(1 + 2n\,\frac{1-b}{a-b}\right). \tag{5}$$

As proved by Sonnevend [9], this choice is optimal and cannot be improved by adaptive methods, although the latter may prove superior in practice.

Note that, if f is concave increasing and no a priori information is available on the slopes a and b, i.e., $a = +\infty$ and $b = 0$, then we obtain

$$\mathcal{E}_n(a,b) = \frac{1}{2(n+1)^2}.$$

3. Proof of the theorem

As the proof of Theorem 1 is lengthy, due to the many cases and subcases that have to be probed, we only provide an outline. The reader interested in the complete proof is referred to Guérin [4]. The proof proceeds by induction on n. The result clearly holds for $n = 0$. For $n \geq 1$ we evaluate the expression

$$\mathcal{R}_n(x) = \max_v \max_\mu \left\{ xv\mathcal{E}_0\left(\frac{x}{v}a, \frac{x}{v}\mu\right) + (1-x)(1-v)\mathcal{E}_{n-1}\left(\frac{1-x}{1-v}\,\mu, \frac{1-x}{1-v}\,b\right)\right\},$$

working backwards with respect to the two "max" operators. For fixed x and v, let us consider the function

$$\phi(\mu) = 2\left[xv\mathcal{E}_0\left(\frac{x}{v}a, \frac{x}{v}\mu\right) + (1-x)(1-v)\mathcal{E}_{n-1}\left(\frac{1-x}{1-v}\mu, \frac{1-x}{1-v}b\right)\right].$$

Using the induction hypothesis to eliminate the terms $\mathcal{E}_0$ and $\mathcal{E}_{n-1}$, ϕ can be written as

$$\phi(\mu) = A\frac{v-\mu x}{a-\mu} + B\frac{(1-x)\mu - (1-v)}{n^2(\mu-b)},$$

where $A = ax - v$ and $B = 1 - v - (1-x)b$ are nonnegative scalars.

The graph of ϕ is given on Figure 3. This function has two local optima, denoted μ^- and μ^+. Its maximum $\mu_{\max}$ on $[(1-v)/(1-x), v/x]$ is attained either at μ^+ or at one of the endpoints of the interval. This yields three cases, each one corresponding to the location of the point (x, v) within the triangle OPT of Figure 4. The three regions I, II and III are defined, respectively, as quadrilateral $PQRS$, triangle RST and triangle ORQ. The maximum of ϕ occurs at μ^+ if

Fig. 4. Three cases for μ.

(x, v) belongs to region I, at $(1-v)/(1-x)$ if (x, v) belongs to region II and at v/x if (x, v) belongs to region III.

We introduce, for fixed x, the function ψ:

$$\psi(v) = n^2(a-b)\phi(\mu_{\max}).$$

The value $v_{\max}$ at which ψ reaches its maximum defines a piecewise smooth function of x consisting of three linear and one quadratic pieces. There are two cases to be considered, depending on whether n is larger or less than $(a-1)/(1-b)$. The function $v_{\max}$, illustrated on Figure 5, is defined as

$$v_{\max} = \begin{cases} ax & \text{if } x \in (0, W_1] \\ bx + (1-b)\sqrt{x} & \text{if } x \in [W_1, D_1] \\ \dfrac{a+b}{2}x + \dfrac{2(n+1) - a(2n+1)b}{2(n+1)^2} & \text{if } x \in [D_1, E_1] \\ bx + (1-b) & \text{if } x \in [E_1, 1) \end{cases}$$

in the case $n \leq (a-1)/(1-b)$, and by

$$v_{\max} = \begin{cases} ax & \text{if } x \in (0, F_1] \\ \dfrac{a+b}{2}x + \dfrac{2(n+1) - a(2n+1)b}{2(n+1)^2} & \text{if } x \in [F_1, G_1] \\ 1 - a + ax + (a-1)\sqrt{1-x} & \text{if } x \in [G_1, V_1] \\ bx + (1-b) & \text{if } x \in [V_1, 1) \end{cases}$$

if $n \geq (a-1)/(1-b)$. (Subscript "1" refers to the x-coordinate of a point.)

Fig. 5. The function $v_{\max}$

To conclude the proof, we determine the minimum of the function $R_n(x)$ over the interval $(0, 1)$ by computing the minimal value of R_n over each subinterval. Next, we check that the minimum occurs at the point x^*, with minimal value given by the formula of Theorem 1. This corresponds to (x, v) belonging to region I and

$$v_{\max} = \frac{a+b}{2}x + \frac{2(n+1) - a(2n+1)b}{2(n+1)^2}.$$

4. Numerical tests

Algorithm DYN is based on the optimal formula provided by Theorem 1. It computes the optimal location of n points from left to right or right to left, using the formula for x^* and a straightforward scaling procedure. The choice of the direction is determined by a heuristic procedure based on a priori information. The performance of DYN is compared with that of SONN, the optimal passive algorithm proposed by Sonnevend [9]. The computational complexity of both algorithms is $\mathcal{O}(n)$ function and derivative evaluations.

The performance of DYN and SONN was tested on sets of randomly generated concave functions. Three functional forms were considered: smooth, piecewise linear (PL) and piecewise smooth (PS). For each form, two samples were produced: one consisting of concave increasing functions and the other of general concave functions.

For each sample, the average error was computed for values of n ranging from 1 to 20. This is consistent with the range considered in the bicriteria traffic equilibrium problem discussed in the introduction. The results from both algorithms, which were compared by taking the ratio of the average error of SONN over the average error of DYN, are illustrated in Figures 6(a)-6(f).

- For nearly all functions tested, DYN performed at least as well as SONN. In the case were DYN's performance is worse, the difference in accuracy is at most 2%.
- On some specific functions, the gain in accuracy achieved by DYN is as high as 400%. Large gains were observed on functions exhibiting strong curvature near the endpoints.
- On average, DYN performed better that SONN on all six samples. Gains in accuracy were largest for piecewise smooth functions and least for smooth functions, with gains for piecewise linear functions falling in between.

Fig. 6. Ratio SONN over DYN of average errors for (a) smooth concave increasing, (b) smooth concave, (c) PL concave increasing, (d) PL concave, (e) PS concave increasing and (f) PS concave functions.

(a) (b)

(c) (d)

(e) (f)

References

1. R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton (1962).

2. R.E. Burkard, H.W. Hamacher and G. Rote. Sandwich approximation of univariate convex functions with an application to separable convex programming. Naval Research Logistics **38** (1991), 911-924.
3. B Fruhwirth, R.E. Burkard and G. Rote. Approximation of convex curves with application to the bicriterial minimum cost flow problem. European Journal of Operational Research. **42** (1989), 326-338.
4. J. Guérin. *Une méthode adaptative pour l'approximation de fonctions concaves croissantes*. Mémoire de maîtrise de l'École Polytechnique de Montréal (2000).
5. P. Marcotte and D.L. Zhu. Equilibria with infinitely many differentiated classes of customers. Complementary and Variational Problems, State of the Art. Jong-Shi Pang and Michael Ferris eds., SIAM. Philadelphia (1997), 234-258.
6. P. Marcotte, S. Nguyen and K. Tanguay. Implementation of an efficient algorithm for the multiclass traffic assignment problem. Proceedings of the 13th International Symposium on Transportation and Traffic Theory. Jean-Baptiste Lesort ed., Pergamon (1996), 217-236.
7. E. Novak. Quadrature formulas for convex classes of functions. International Series of Numerical Mathematics **112**. Birkhäuser Verlag, Basel (1993), 283-296.
8. G. Rote. The convergence rate of the sandwich algorithm for approximating convex functions. Computing **48** (1992), 337-361.
9. G. Sonnevend. Optimal passive and sequential algorithms for the approximation of convex functions in $L_p[0,1]^s$, $p = 1, \infty$. Constructive function theory '81. Sofia (1983).
10. A. G. Sukharev. *Minimax models in the theory of numerical methods*, Theory and Decision Library Series B Vol. **21**. Kluwer Academic Publishers (1992).
11. X.Q. Yang and C.J. Goh. A method for convex curve approximation. European Journal of Operational Research **97** (1997), 205-212.