

A Riemannian subgradient algorithm for economic dispatch with valve-point effect[☆]

Pierre B. Borckmans^{*}, S. Easter Selvan, Nicolas Boumal, P.-A. Absil^{**}

Department of Mathematical Engineering, ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:

Received 4 September 2012

Received in revised form 23 January 2013

Keywords:

Economic load dispatch

Nonsmooth optimization

Differential geometry

ABSTRACT

The economic load dispatch problem (ELDP) is a classical problem in the power systems community. It consists in the optimal scheduling of the output of power generating units to meet the required load demand subject to unit and system inequality and equality constraints. This optimization problem is challenging on three different levels: the geometry of its feasible set, the non-differentiability of its cost function and the multimodal aspect of its landscape. For this reason, ELDP has received much attention in the past few years and numerous derivative-free techniques have been proposed to tackle its multimodal and nondifferentiable characteristics. In this work we propose a different approach, exploiting the rich geometrical structure of the problem. We show that the (nonlinear) equality constraint can be handled in the framework of Riemannian manifolds and we develop a feasible (all iterates satisfy the constraints) subgradient descent algorithm to provide fast convergence to local minima. To this end, we show that Clarke's calculus can be used to compute a deterministic admissible descent direction by solving a simple, low-dimensional quadratic program. We test our approach on four real data sets. The proposed method provides fast local convergence and scales well with respect to the problem dimension. Finally, we show that the proposed algorithm, being a local optimization method, can be incorporated in existing heuristic techniques to provide a better exploration of the search space.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The economic load dispatch problem (ELDP) is the optimal scheduling of the output of power generating units to meet the required load demand subject to unit and system equality and inequality constraints [1]. In traditional ELDP, the cost function for each generator is modeled by a single quadratic function. Nevertheless, in practice, one has to take into account highly nonlinear input–output characteristics arising due to valve-point loadings or generating unit ramp rate limits. As a consequence, we end up with a nonsmooth, equality- and inequality-constrained optimization problem, (5), which is in general multimodal (it presents several local optima) and for which classical smooth optimization techniques are thus not suitable.

[☆] This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was financially supported by the Belgian FRFC (Fonds de la Recherche Fondamentale Collective).

^{*} Corresponding author.

^{**} Corresponding author. Tel.: +32 10472597.

E-mail addresses: pborckmans@gmail.com (P.B. Borckmans), easterselvans@gmail.com (S. Easter Selvan), nicolas.boumal@uclouvain.be (N. Boumal), absil@inma.ucl.ac.be (P.-A. Absil).

URL: <http://sites.uclouvain.be/absil/> (P.-A. Absil).

For this reason, ELDP has received much attention in the past few years and numerous derivative-free techniques have been proposed to tackle its multimodal and nondifferentiable aspects. Popular techniques include genetic algorithms (GA) [2], evolutionary programming [3], particle swarm optimization (PSO) [4], and differential evolution (DE) [5].

More recently, “global–local” hybrid methods have appeared that combine a “global” method having good global searching abilities and a “local” method with better fine-tuning abilities. Even though there may not be a clear frontier between the two groups of methods, it is quite evident that methods such as GA, PSO and DE belong to the “global” group whereas a method such as sequential quadratic programming (SQP) belongs to the “local” group. Desirable properties of a local method include the following. (i) The method should be *feasible*, i.e., all the iterates should satisfy the equality and inequality constraints. Indeed, a drawback of infeasible methods is that even if the outcome is close to satisfying the constraints, the cost function value at the outcome may be quite different from its value at the nearest point satisfying the constraints. Another downside of infeasibility is that the intermediate results are not directly usable since they do not respect the balance constraint. (ii) The iterates should converge to a local minimizer of the cost function. (iii) There is a well-known trade-off between low numerical cost of the iteration and fast convergence of the iterates; as a simplistic illustration, concatenating a few steps of a given method improves the speed of convergence per iterate but degrades the numerical cost per iterate. If the sought accuracy is sufficiently low – which is arguably the case of the ELDP where the coefficient values and even the model itself are debatable – then it is preferable to run a lot of cheaper iterations, as this makes it possible to check a stopping criterion more frequently and hence to avoid the unnecessary computational effort inherent to overconvergence. (iv) Methods that exploit the very particular structure of the ELDP should be preferred over “out of the box” numerical algorithms.

SQP is a popular choice for the local method in global–local ELDP algorithms; see, e.g., [6–9]. Whereas SQP is a welcome complement to global search methods, it does not satisfy any of the properties mentioned above. Specifically, it is an infeasible method, since the equality constraint is only satisfied in the limit. Convergence, while empirically observed, is not guaranteed because the classical convergence theory of SQP assumes, among other things, that the cost function and the constraint functions have continuous first derivatives [10, Theorem 18.3], which is not the case in the ELDP (5). As a quadratically convergent method [10, Theorem 18.4], it can be seen as favoring fast convergence of the iterates at the expense of a higher numerical cost per iterate. Finally, it is a general-purpose algorithm, not specifically tailored to the ELDP.

Other local methods have been proposed in combination with global methods. The Nelder–Mead (NM) method has been shown to be an effective local method in combination with PSO [11], but it does not exploit the readily accessible first-order information on the cost function. The modified subgradient (MSG) method used in [12] exploits first-order information on the cost function, but feasibility of the iterates is achieved only after a certain number of steps in view of the *sharp* augmented Lagrangian approach. Shor’s *r*-algorithm, combined with an improved differential evolution (IDE) method in [13], is also of the subgradient type, but the iterates are not feasible.

In this paper, we develop a novel method that satisfies properties (i), (iii), and (iv). It is also built to satisfy (ii) and does it empirically, however, as a consequence of (iv), its convergence does not immediately follow from an existing result, and a detailed convergence analysis is beyond the scope of this paper.

Property (i), feasibility of the iterates, is enforced using the framework of Riemannian optimization. The adequacy of this framework stems from the fact that the equality constraint (3) defines an ellipsoid, which admits a natural structure of a Riemannian manifold. Riemannian optimization, also called optimization on manifolds, is a vibrant area of research nowadays, whose foundations can be found, e.g., in [14–16]. Since this framework is new in the ELDP context, we dedicate a significant part of this paper to laying out the necessary background.

Property (iii) comes by preferring a steepest-descent approach over a second-order approach. However, in view of the non-smoothness due to the valve-point effect, the steepest-descent approach does not rely on gradient techniques but rather on subgradient techniques, using the framework of Clarke’s generalized calculus [17].

In view of the above, the proposed method fits in the framework of Riemannian subgradient descent. With respect to the general-purpose Riemannian subgradient descent method of Dirk et al. [18], a contribution of our development is to incorporate bound constraints in order to handle the generator capacity constraints (2) present in the ELDP. Another contribution of this work is that, whereas many heuristic algorithms for the ELDP consist of (a combination of) existing black-box optimization techniques, the proposed method strives to exploit as much as possible the very particular structure of the ELDP. This allows notably for an efficient representation of the generalized gradient, which enables fast computation of a descent direction by solving a low-dimensional quadratic program.

In summary, the proposed technique provides fast convergence to a nearby local minimum of the ELDP (5), while satisfying the power balance (3) and capacity constraints (2) throughout the optimization process. Therefore, the aforementioned heuristics largely explored in the literature and the proposed subgradient descent algorithm present very complementary properties: the multimodal aspect of the ELDP can be addressed using any global feasible exploration tool while the local refinement of a potential solution is efficiently provided by the proposed approach, including a check for the stationarity of the final iterate.

The remainder of the paper is organized as follows. In Section 2, the ELDP with the valve-point effect is briefly presented, followed by a detailed treatment of the underlying geometry of the optimization problem and an introduction to the necessary differential geometry tools in Section 3. The subgradient descent algorithm is presented in Section 4. Subsequently, its formulation on the Riemannian manifold and its specialization for the ELDP are discussed. Section 5 presents the

implementation details. In Section 6, numerical results are presented and we show how our algorithm can be hybridized with a global scheme to deal with the multimodal aspect of the ELDP. Finally, conclusions are drawn in Section 7.

2. ELDP considering the valve-point effect

2.1. Problem statement

Traditionally, generating-unit cost functions are considered to be convex with the heat rate curves exhibiting monotonically increasing characteristics. However, in reality, the steam admission valves in large steam turbines cause discontinuities in the incremental heat rate curves. Therefore, to accurately model the ELDP, the valve-point loadings in the n generating units have to be incorporated, leading to nonconvex input–output characteristics of the generating units [19]; the cost function is then stated as

$$f_T(\mathbf{p}) = \sum_{i=1}^n f^i(p_i), \quad \mathbf{p} = [p_1, p_2, \dots, p_n]^\top$$

$$= \sum_{i=1}^n a_i p_i^2 + b_i p_i + c_i + |d_i \sin[e_i(p_i^{\min} - p_i)]|. \quad (1)$$

Here $f_T(\mathbf{p})$ is the total production cost (\$/h) pertaining to the n -dimensional output power vector \mathbf{p} and $f^i(p_i)$ is the incremental fuel cost function (\$/h) pertaining to the real power output of the i th unit, p_i . For the i th generating unit, the cost coefficients are denoted by a_i , b_i , c_i , and the constants from the valve-point effect by d_i , e_i . All the coefficients are positive. In matrix notation, we have

$$f_T(\mathbf{p}) = \mathbf{p}^\top \text{Diag}(\mathbf{a})\mathbf{p} + \mathbf{b}^\top \mathbf{p} + \mathbf{c}^\top \mathbf{1} + \mathbf{d}^\top |\sin[\mathbf{e} \circ (\mathbf{p}^{\min} - \mathbf{p})]|,$$

where \circ denotes the component-wise product, $\text{Diag}(\cdot)$ denotes the diagonal matrix obtained from the entries of its vector argument, and the sine term and the absolute value are taken component-wise.

The cost function $f_T(\mathbf{p})$ is to be minimized subject to the following inequality and equality constraints:

(a) generator capacity constraints

$$p_i^{\min} \leq p_i \leq p_i^{\max}, \quad i = 1, \dots, n \quad (2)$$

where p_i^{\min} and p_i^{\max} are the lower and upper power generating limits of the i th unit (MW);

(b) real power balance constraint

$$\sum_{i=1}^n p_i = p_D + p_L(\mathbf{p}), \quad (3)$$

where p_D is the power demand (MW) and $p_L(\mathbf{p})$ stands for the power loss (MW) expressed as

$$p_L(\mathbf{p}) = \sum_{i=1}^n \sum_{j=1}^n p_i B_{ij} p_j + \sum_{i=1}^n b_i^0 p_i + b^{00}, \quad (4)$$

or in matrix notation $p_L(\mathbf{p}) = \mathbf{p}^\top \mathbf{B} \mathbf{p} + \mathbf{b}^0 \mathbf{p} + b^{00}$. Coefficients B_{ij} , b_i^0 , b^{00} are the transmission loss coefficients (B -coefficients) given by the elements of the square matrix \mathbf{B} of size $n \times n$, the vector \mathbf{b}^0 of length n , and the constant b^{00} , respectively. The matrix \mathbf{B} is symmetric positive-definite, hence $p_L(\mathbf{p})$ is a convex quadratic function of \mathbf{p} .

To summarize, we consider the following optimization problem, that we will refer to as the ELDP:

$$\min_{\mathbf{p} \in \mathbb{R}^n} f_T(\mathbf{p}) \quad (1)$$

subject to (2) and (3). (5)

2.2. Geometry of the feasible set

In this subsection, we explore the geometrical interpretation and consequences of constraints (2) and (3).

The bound constraints (2) force \mathbf{p} to lie inside a box whose faces are parallel to the reference frame. The equality constraint (3) imposes that \mathbf{p} lies on a quadric surface, specifically on an ellipsoid since \mathbf{B} is positive definite. The feasible set of (5), denoted by Ω , is thus composed of the points on the ellipsoid that are also inside the box:

$$\Omega := \{\mathbf{p} \in \mathbb{R}^n : \mathbf{p} \text{ satisfies (2) and (3)}\}$$

$$= \{\mathbf{p} \in \mathbb{R}^n : \mathbf{p}^{\min} \leq \mathbf{p} \leq \mathbf{p}^{\max}, \mathbf{p}^\top \mathbf{B}^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}} \mathbf{p} + c^{\text{ell}} = 0\}, \quad (6)$$

where $\mathbf{1}$ is the vector of all ones, $\mathbf{B}^{\text{ell}} = \mathbf{B}$, $\mathbf{b}^{\text{ell}} = \mathbf{b}^0 - \mathbf{1}$, and $c^{\text{ell}} = b^{00} + p_d$. In general, Ω may be composed of more than one connected component, but we did not observe this in practical ELDP instances. The feasible set Ω is depicted in Fig. 1 for the cases $n = 2$ and $n = 3$.

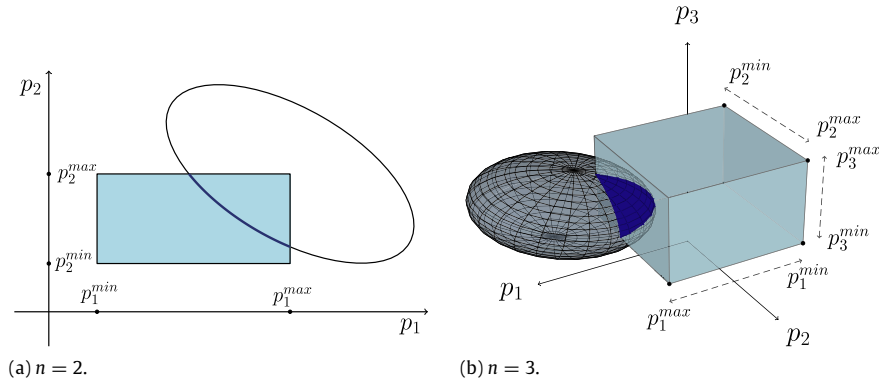


Fig. 1. Representation of the \mathbf{p} -space along with the constraints (inside the box and on the ellipsoid) and the feasible set Ω (darker blue). Notice that Ω has measure zero in \mathbb{R}^n . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

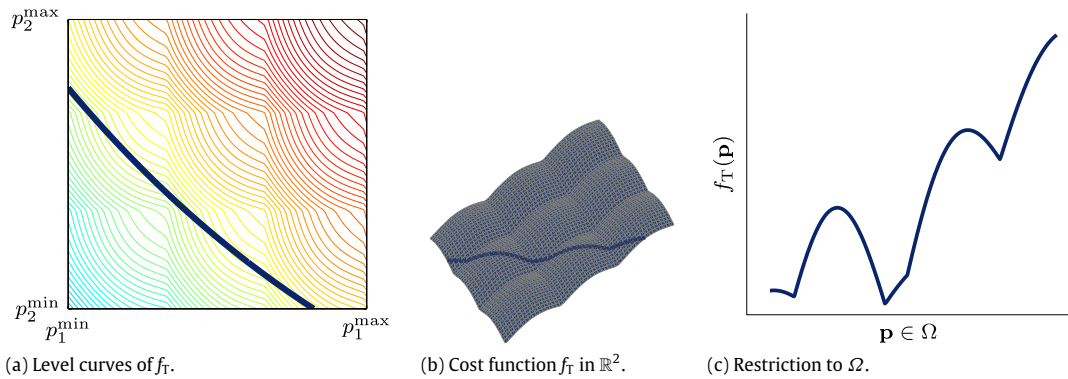


Fig. 2. Landscape of the cost function $f_T(\mathbf{p})$ for an illustrative 2-dimensional case. (a) Level curves of $f_T(\mathbf{p})$ and the ellipsoid constraint. (b) Graph of the cost function $f_T(\mathbf{p})$. (c) Graphs of the restriction of $f_T(\mathbf{p})$ to Ω . The cost function is multimodal and piecewise smooth.

2.3. Structure of the optimization landscape

In this subsection, we provide some insight about the cost function $f_T(\mathbf{p})$ defined in (1). A first observation is that this function is separable, i.e., it consists of a sum of components f^i , each depending only on p_i , the i th component of \mathbf{p} . However, problem (5) does not decompose into independent one-dimensional problems because the components of \mathbf{p} are coupled through the equality constraint (3). The cost function $f_T(\mathbf{p})$ is continuous but not differentiable everywhere. Indeed, at any point $\mathbf{q} \in \mathbb{R}^n$ for which one or more components q_i cancel the corresponding sine term in f^i , the gradient of f_T does not exist because of the absolute value. These points, termed *nondifferentiable points* or *kink points*, are given by the following expression:

$$\mathbf{q} = (q_1, \dots, q_n), \quad \exists i \text{ s.t. } q_i = p_i^{\min} + \frac{k\pi}{e_i}, \quad k \in \mathbb{N}. \quad (7)$$

Nevertheless, at all other points \mathbf{p} in \mathbb{R}^n , the gradient of f_T can be computed:

$$\nabla f_T(\mathbf{p}) = 2\text{Diag}(\mathbf{a})\mathbf{p} + \mathbf{b} - \mathbf{d} \circ \mathbf{e} \circ \cos \boldsymbol{\theta} \circ \text{sign}(\sin \boldsymbol{\theta}), \quad (8)$$

where $\boldsymbol{\theta}$ stands for $[\mathbf{e} \circ (\mathbf{p}^{\min} - \mathbf{p})]$. The function $f_T(\mathbf{p})$ is thus piecewise smooth (as defined in [20]) and one can use the canvas of Clarke's generalized calculus [17] to compute its generalized gradient at the nondifferentiable points. This analysis will be carried out in Section 4.

Now, recall that problem (5) consists of minimizing $f_T(\mathbf{p})$ over the feasible set Ω . As depicted in Fig. 2 for an illustrative 2-dimensional case, the restriction of $f_T(\mathbf{p})$ to Ω presents a partially smooth and multimodal landscape. If the portion Ω of the ellipsoid is not too large, a similar representation can be obtained for a 3-dimensional case, by projecting the feasible set Ω onto the nearest 2-dimensional plane; see Fig. 3, where the observations made for the 2-dimensional case can also be noted.

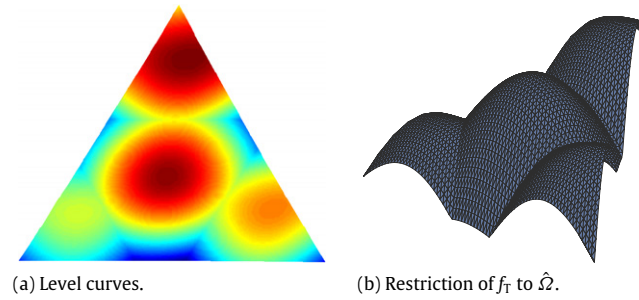


Fig. 3. Landscape of the cost function $f_T(\mathbf{p})$ for an illustrative 3-dimensional case. This representation is made possible by projecting the 3 dimensional feasible set Ω onto the nearest plane. (a) Level curves of the restriction of $f_T(\mathbf{p})$ to the projection $\hat{\Omega}$ of Ω . (b) Graph of the restriction of $f_T(\mathbf{p})$ to $\hat{\Omega}$. The cost function is multimodal and piecewise smooth.

2.4. Summary of the optimization challenges

From the previous sections, it appears that the ELDP is a challenging optimization problem on three different levels: the geometry of its feasible set, the non-differentiability of its cost function, and the multimodal aspect of its landscape. The rest of this paper presents our approach to deal with these difficulties.

3. Optimization exploiting the geometry of Ω

The proposed local method can be viewed as an adaptation of the classical line-search scheme

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \alpha^k \mathbf{d}^k, \quad (9)$$

where \mathbf{p}^k and \mathbf{p}^{k+1} denote the current and next iterates, \mathbf{d}^k is the search direction, and α^k is the step length. In this section, with property (i) – feasibility – in mind, we generalize the “+” operation by means of the concept of retraction and we introduce “admissibility” conditions on \mathbf{d}^k in order to guarantee that \mathbf{p}^{k+1} remains in Ω for all α^k sufficiently small. Then, in Section 4, we will show how to choose \mathbf{d}^k as the *steepest* admissible direction. Finally, the selection of α^k , as well as other implementation details, will be addressed in Section 5.

3.1. The ellipsoid manifold

As explained in Section 2.2, the feasible set Ω of the ELDP (5) has a rich geometrical structure that we would like to exploit. The set of points that satisfy the equality constraint (3) is an ellipsoid centered at

$$\mathbf{a}^{\text{ell}} = -\frac{1}{2}(\mathbf{B}^{\text{ell}})^{-1}\mathbf{b}^{\text{ell}}. \quad (10)$$

This smooth surface has a natural structure of a manifold, specifically of an $(n - 1)$ -dimensional submanifold of \mathbb{R}^n . We will call it the ellipsoid manifold:

$$\begin{aligned} \mathcal{E}^{n-1} &:= \{\mathbf{p} \in \mathbb{R}^n : \mathbf{p} \text{ satisfies (3)}\} \\ &= \{\mathbf{p} \in \mathbb{R}^n : \mathbf{p}^\top \mathbf{B}^{\text{ell}} \mathbf{p} + \mathbf{p}^\top \mathbf{b}^{\text{ell}} + c^{\text{ell}} = 0\}. \end{aligned} \quad (11)$$

The gist of the proposed Riemannian optimization approach is to restrict the optimization domain \mathbb{R}^n to the constraint manifold, in this case with the resulting advantage that every iterate belongs to \mathcal{E}^{n-1} , i.e., satisfies the equality constraint (3). The ELDP (5) then becomes:

$$\min_{\mathbf{p} \in \mathcal{E}^{n-1}} f_T(\mathbf{p}) \quad \text{subject to } \mathbf{p}^{\min} \leq \mathbf{p} \leq \mathbf{p}^{\max}.$$

We mention that an alternative way, used by several ELDP heuristics, of respecting the equality constraint (3) is to resort to a slack variable, which amounts to performing the optimization on $(n - 1)$ variables while computing the last variable explicitly. A difficulty with this approach is that the description of the cost function f_T , of the feasible set Ω , and of the nondifferentiable points in terms of the $(n - 1)$ remaining variables becomes more intricate. This difficulty is avoided in the Riemannian approach.

3.2. Riemannian optimization ingredients

The classical line-search scheme (9) is not meant to produce iterates that remain on a submanifold such as \mathcal{E}^{n-1} . In this subsection, we provide the needed ingredients of differential geometry (the branch of mathematics that studies manifolds) in order to produce iterates on an abstract submanifold \mathcal{M} of \mathbb{R}^n . Then, in Section 3.3, we will specialize the ingredients to the manifold \mathcal{E}^{n-1} , and finally, in Section 3.5, we will reintroduce the bound constraints (2).

Instrumental to the definition of an admissible direction \mathbf{d}^k is the notion of *tangent space* to \mathcal{M} at a point $\mathbf{p} \in \mathcal{M}$, denoted by $T_{\mathbf{p}}\mathcal{M}$, and defined as the following vector subspace of \mathbb{R}^n :

$$T_{\mathbf{p}}\mathcal{M} = \{\xi \in \mathbb{R}^n : \exists c : \mathbb{R} \rightarrow \mathcal{M} \text{ with } c(0) = \mathbf{p}, c'(0) = \xi\},$$

where $c'(0)$ is the usual derivative at 0 of the curve c (assumed to exist). From this definition, we see that $T_{\mathbf{p}}\mathcal{M}$ is the set of vectors that are tangent to the manifold at \mathbf{p} . Geometrically, this notion coincides with the concept of a tangent plane to a smooth surface, as depicted in Fig. 4. The tangent bundle $T\mathcal{M}$ is the collection of the tangent spaces at all $\mathbf{p} \in \mathcal{M}$.

The notion of *steepest* admissible direction will require a norm on $T_{\mathbf{p}}\mathcal{M}$. Each tangent space $T_{\mathbf{p}}\mathcal{M}$ is a vector space and as such can be endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathbf{p}}$. The natural way of doing this is by restricting the canonical inner product of \mathbb{R}^n to $T_{\mathbf{p}}\mathcal{M}$, i.e.,

$$\langle \cdot, \cdot \rangle_{\mathbf{p}} : T_{\mathbf{p}}\mathcal{M} \times T_{\mathbf{p}}\mathcal{M} \rightarrow \mathbb{R} : (\xi, \zeta) \mapsto \langle \xi, \zeta \rangle_{\mathbf{p}} = \langle \xi, \zeta \rangle = \xi^{\top} \zeta.$$

We then say that \mathcal{M} is a *Riemannian submanifold* of \mathbb{R}^n . The inner product induces a notion of norm: $\|\xi\|_{\mathbf{p}} = \langle \xi, \xi \rangle_{\mathbf{p}}^{1/2}$.

The *normal space* to \mathcal{M} at a point $\mathbf{p} \in \mathcal{M}$, denoted as $N_{\mathbf{p}}\mathcal{M}$, is the orthogonal complement of $T_{\mathbf{p}}\mathcal{M}$ in \mathbb{R}^n :

$$N_{\mathbf{p}}\mathcal{M} = (T_{\mathbf{p}}\mathcal{M})^{\perp}.$$

One can then compute the *projection* $P_{\mathbf{p}}(\mathbf{v})$ of a vector $\mathbf{v} \in \mathbb{R}^n$ onto $T_{\mathbf{p}}\mathcal{M}$ by removing the normal component of \mathbf{v} .

We now turn to the generalization of the “+” operation of (9). A *retraction* on \mathcal{M} [21,16] is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto \mathcal{M} that satisfies $R(0_{\mathbf{p}}) = \mathbf{p}$ for all \mathbf{p} (where $0_{\mathbf{p}}$ denotes the origin of $T_{\mathbf{p}}\mathcal{M}$) and $\frac{d}{dt}R(t\xi_{\mathbf{p}})|_{t=0} = \xi_{\mathbf{p}}$ for all $\xi_{\mathbf{p}} \in T_{\mathbf{p}}\mathcal{M}$. The restriction of R to $T_{\mathbf{p}}\mathcal{M}$ is denoted by $R_{\mathbf{p}}$. Observe that R given by $R_{\mathbf{p}}(\alpha\mathbf{d}) = \mathbf{p} + \alpha\mathbf{d}$ is a valid retraction in \mathbb{R}^n , hence the generalization.

Using these tools, one can adapt the iterative process (9) as follows: assuming that $\mathbf{p}^k \in \mathcal{M}$, construct $\mathbf{p}^{k+1} \in \mathcal{M}$ according to

$$\mathbf{p}^{k+1} = R_{\mathbf{p}^k}(\alpha^k P_{\mathbf{p}^k}(\mathbf{d}^k)), \quad \alpha^k \in \mathbb{R}, \mathbf{d}^k \in \mathbb{R}^n.$$

3.3. Optimization ingredients on \mathcal{E}^{n-1}

We now specifically provide the aforementioned tools for the ellipsoid manifold \mathcal{E}^{n-1} . A schematic diagram is given in Fig. 4.

Let $c(t)$ be a curve on \mathcal{E}^{n-1} parametrized by $t \in \mathbb{R}$:

$$c : \mathbb{R} \rightarrow \mathcal{E}^{n-1} : t \rightarrow c(t), \quad \text{s.t. } c(0) = \mathbf{p}, \dot{c}(0) = \xi.$$

For all $t \in \mathbb{R}$, the ellipsoid equation gives $c(t)^{\top} B^{\text{ell}} c(t) + c(t)^{\top} \mathbf{b}^{\text{ell}} + c^{\text{ell}} = 0$. Taking the derivative with respect to t and evaluating at $t = 0$ (substituting for \mathbf{p} and ξ), one obtains:

$$\xi^{\top} B^{\text{ell}} \mathbf{p} + \mathbf{p}^{\top} B^{\text{ell}} \xi + \xi^{\top} \mathbf{b}^{\text{ell}} = 0.$$

Since B^{ell} is a symmetric matrix, one can rewrite:

$$\xi^{\top} (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}) = 0.$$

The *tangent space* is thus defined as follows:

$$T_{\mathbf{p}}\mathcal{E}^{n-1} = \{\xi \in \mathbb{R}^n : \xi^{\top} (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}) = 0\}. \quad (12)$$

The *normal space* is then obtained by:

$$\begin{aligned} N_{\mathbf{p}}\mathcal{E}^{n-1} &= \{\mathbf{v} \in \mathbb{R}^n : \xi^{\top} \mathbf{v} = 0, \forall \xi \in T_{\mathbf{p}}\mathcal{E}^{n-1}\} \\ &= \{\tau (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}), \tau \in \mathbb{R}\}. \end{aligned} \quad (13)$$

The *projection* $P_{\mathbf{p}}(\mathbf{v})$ of a vector $\mathbf{v} \in \mathbb{R}^n$ onto $T_{\mathbf{p}}\mathcal{E}^{n-1}$ can then be constructed so as to remove the normal component of \mathbf{v} :

$$P_{\mathbf{p}}(\mathbf{v}) = \mathbf{v} - \tau (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}),$$

where the value of $\tau \in \mathbb{R}$ must be determined to ensure that $P_{\mathbf{p}}(\mathbf{v})$ belongs to $T_{\mathbf{p}}\mathcal{E}^{n-1}$:

$$(\mathbf{v} - \tau (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}))^{\top} (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}}) = 0,$$

which yields

$$\tau = \frac{\mathbf{v}^{\top} (2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}})}{\|(2B^{\text{ell}} \mathbf{p} + \mathbf{b}^{\text{ell}})\|^2}.$$

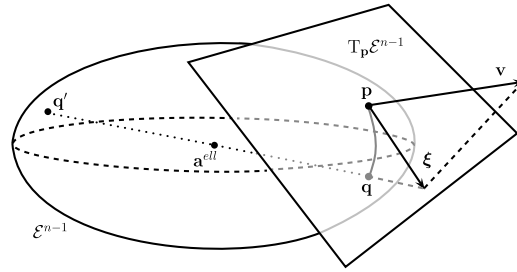


Fig. 4. Illustration of the optimization tools on the ellipsoid manifold \mathcal{E}^{n-1} . The vector $\mathbf{v} \in \mathbb{R}^n$ is projected onto the tangent space $T_p \mathcal{E}^{n-1}$: $\xi = P_p(\mathbf{v})$. The retraction of the tangent vector ξ is then computed: $\mathbf{q} = R_p(\xi)$.

Defining $\mathbf{n}_p = \frac{(2\mathbf{B}^{\text{ell}}\mathbf{p} + \mathbf{b}^{\text{ell}})}{\|(2\mathbf{B}^{\text{ell}}\mathbf{p} + \mathbf{b}^{\text{ell}})\|}$ as the unit normal vector to the ellipsoid at \mathbf{p} , the projection becomes:

$$P_p(\mathbf{v}) = \mathbf{v} - \mathbf{n}_p \mathbf{n}_p^\top \mathbf{v} = (I - \mathbf{n}_p \mathbf{n}_p^\top) \mathbf{v}. \quad (14)$$

It remains to choose a retraction R on \mathcal{E}^{n-1} . On a Riemannian manifold, the exponential map, based on geodesics, is a possible choice for R [16, Section 5.4], but computing geodesics on the ellipsoid manifold \mathcal{E}^{n-1} is a research area on its own; see, e.g., [22,23] and the references therein. Instead, we use a simpler choice that consists, for $\mathbf{p} \in \mathcal{E}^{n-1}$ and $\xi \in T_p \mathcal{E}^{n-1}$, in defining $R_p(\xi)$ to be the intersection with \mathcal{E}^{n-1} of the line segment between $\mathbf{p} + \xi$ and the center \mathbf{a}^{ell} of the ellipsoid. Specifically,

$$\begin{aligned} R_p(\xi) &= \mathbf{a}^{\text{ell}} + \beta (\mathbf{p} + \xi - \mathbf{a}^{\text{ell}}) \\ &= \mathbf{a}^{\text{ell}} + \beta \mathbf{w} = \mathbf{q}, \end{aligned} \quad (15)$$

where \mathbf{a}^{ell} is given by (10) and $\beta \in \mathbb{R}$ must be chosen so as to satisfy $\mathbf{q} \in \mathcal{E}^{n-1}$:

$$\mathbf{q}^\top \mathbf{B}^{\text{ell}} \mathbf{q} + \mathbf{q}^\top \mathbf{b}^{\text{ell}} + c^{\text{ell}} = 0, \quad (16)$$

which yields:

$$\beta^2 (\mathbf{w}^\top \mathbf{B}^{\text{ell}} \mathbf{w}) + \beta (\mathbf{w}^\top (2\mathbf{B}^{\text{ell}} \mathbf{a}^{\text{ell}} + \mathbf{b}^{\text{ell}})) + (\mathbf{a}^{\text{ell}\top} (\mathbf{B}^{\text{ell}} \mathbf{a}^{\text{ell}} + \mathbf{b}^{\text{ell}}) + c^{\text{ell}}) = 0. \quad (17)$$

Eq. (17) defines a parabola in β . The roots of this parabola correspond to the two intersections with the ellipsoid of the line going through \mathbf{a}^{ell} and $(\mathbf{p} + \xi)$ (see \mathbf{q} and \mathbf{q}' in Fig. 4). Among these two points, the nearest to $(\mathbf{p} + \xi)$ corresponds to the closest root to 1, which is therefore chosen for β . The fact that R_p is a retraction follows from [24, Theorem 15].

3.4. Restriction to sub-ellipsoids

As will be discussed in Section 4.3, it will prove useful at times to consider the restriction of the ellipsoid \mathcal{E}^{n-1} obtained by fixing some coordinates in $\mathbf{p} = (p_1, \dots, p_n)^\top$. Let $\mathcal{C} \subset \{1, \dots, n\}$ be the indices of these constant coordinates with $|\mathcal{C}| = n_c < n$, and let $\mathbf{P} = (P_1, \dots, P_{n_c})^\top$ be the corresponding constants. The following notation is introduced: given a matrix \mathbf{M} , a vector \mathbf{v} and two sets of indices \mathcal{I}_1 and \mathcal{I}_2 , the sub-matrix $[M_{i,j}]_{i \in \mathcal{I}_1, j \in \mathcal{I}_2}$ is denoted as $\mathbf{M}_{\mathcal{I}_1, \mathcal{I}_2}$ and the sub-vector $[v_i]_{i \in \mathcal{I}_1}$ is denoted as $\mathbf{v}_{\mathcal{I}_1}$.

We are interested in the following set:

$$\mathcal{E}_{\mathcal{C}, \mathbf{P}}^{n-1} := \{\mathbf{p} \in \mathcal{E}^{n-1} : \mathbf{p}_{\mathcal{C}} = \mathbf{P}\}.$$

This set describes a Riemannian manifold which is the intersection of the ellipsoid \mathcal{E}^{n-1} with the intersection of the n_c axis-aligned hyperplanes defined by $\mathbf{p}_{\mathcal{C}} = \mathbf{P}$. This manifold is in fact again an ellipsoid of dimension $(n - n_c - 1)$, as depicted in Fig. 5.

Letting $\mathcal{F} = \{1, \dots, n\} \setminus \mathcal{C}$ be the set of free coordinates, this sub-ellipsoid is given by the following equation:

$$\mathcal{E}_{\mathcal{C}, \mathbf{P}}^{n-1} := \left\{ \mathbf{p} \in \mathbb{R}^n : \mathbf{p}_{\mathcal{F}}^\top \widehat{\mathbf{B}}^{\text{ell}} \mathbf{p}_{\mathcal{F}} + \mathbf{p}_{\mathcal{F}}^\top \widehat{\mathbf{b}}^{\text{ell}} + \widehat{c}^{\text{ell}} = 0, \mathbf{p}_{\mathcal{C}} = \mathbf{P} \right\}, \quad (18)$$

where

$$\begin{aligned} \widehat{\mathbf{B}}^{\text{ell}} &= \mathbf{B}_{\mathcal{F}, \mathcal{F}}^{\text{ell}}, \\ \widehat{\mathbf{b}}^{\text{ell}} &= (\mathbf{b}_{\mathcal{F}}^{\text{ell}} + 2\mathbf{B}_{\mathcal{F}, \mathcal{C}}^{\text{ell}} \mathbf{p}_{\mathcal{C}}), \\ \widehat{c}^{\text{ell}} &= \mathbf{p}_{\mathcal{C}}^\top (\mathbf{B}_{\mathcal{C}, \mathcal{C}}^{\text{ell}} \mathbf{p}_{\mathcal{C}} + \mathbf{b}_{\mathcal{C}}^{\text{ell}}) + c^{\text{ell}}. \end{aligned}$$

The center of this new ellipsoid is then given by $\widehat{\mathbf{a}}_{\mathcal{F}}^{\text{ell}} = -\frac{1}{2}(\widehat{\mathbf{B}}^{\text{ell}})^{-1} \widehat{\mathbf{b}}^{\text{ell}}$ and $\widehat{\mathbf{a}}_{\mathcal{C}} = \mathbf{P}$.

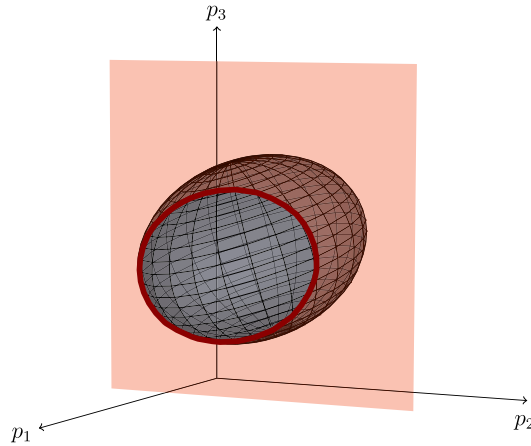


Fig. 5. Illustration of a sub-ellipsoid for $n = 3$, with $\mathcal{C} = \{1\}$ and $\mathcal{F} = \{2, 3\}$.

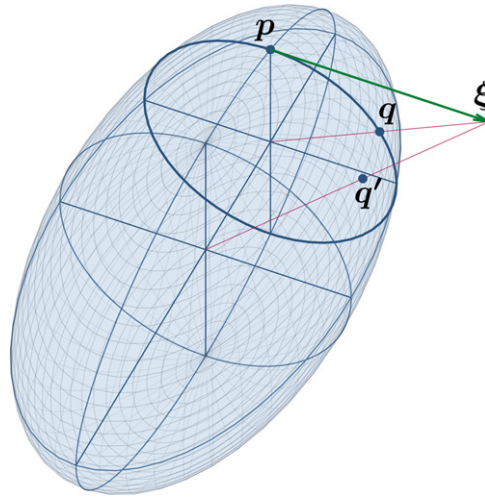


Fig. 6. Retraction of the tangent vector ξ at the point p . When using the retraction R given by (15), the point q' is obtained, outside of the subellipsoid going through p . When using the retraction \hat{R} (19) relative to the center of this subellipsoid, the point q is obtained, belonging to the subellipsoid.

The ingredients presented in the previous section can now be adapted to the sub-ellipsoid when needed. In particular, when computing the retraction $q = R_p(\xi)$ of a vector $\xi \in T_p \mathcal{E}_{\mathcal{C}, p_c}^{n-1}$, one expects q to remain on $\mathcal{E}_{\mathcal{C}, p_c}^{n-1}$. With the definition (15) of R , this is not the case since the scaling factor β affects all of the coordinates of q . However, the following retraction \hat{R} (where \mathcal{C} is omitted in the notation as it can be deduced from the argument ξ , see (44) for the robustified version) can be used to ensure the desired property:

$$\hat{R}_p(\xi) = q, \quad \text{with} \quad \begin{cases} q_{\mathcal{C}} = p_{\mathcal{C}}, \\ q_{\mathcal{F}} = \hat{a} + \beta (p_{\mathcal{F}} + \xi_{\mathcal{F}} - \hat{a}), \end{cases} \quad (19)$$

where β is computed using (17) with the corresponding \hat{B}^{ell} , \hat{b}^{ell} , \hat{c}^{ell} and \hat{a}^{ell} . The difference between the two retractions R and \hat{R} is illustrated in Fig. 6.

Noting that $\hat{R}_p(\xi) = R_p(\xi)$ when $\mathcal{C} = \emptyset$, Eq. (19) will be used whenever a retraction is needed in the remainder of this paper.

3.5. Respecting the bound constraints

So far in this section, we have considered performing optimization on \mathcal{E}^{n-1} . However, as presented in Section 2.2, the feasible set Ω is only a portion of \mathcal{E}^{n-1} , delimited by the bound constraints (2). In view of the sought property (i) mentioned in the introduction, we want to produce iterates that remain within these bounds at all times. To this end, a possible approach

would be to penalize the cost function f_T with a log barrier term:

$$\widehat{f}_T(\mathbf{p}) = \begin{cases} f_T(\mathbf{p}) + \mu_k \phi(\mathbf{p}) & \text{if } \mathbf{p}^{\min} \leq \mathbf{p} \leq \mathbf{p}^{\max}, \\ +\infty & \text{otherwise,} \end{cases}$$

with $\phi(\mathbf{p}) = \ln(\mathbf{p} - \mathbf{p}^{\min}) + \ln(\mathbf{p}^{\max} - \mathbf{p})$,

where $\mu_k \downarrow 0$ when $k \rightarrow \infty$. This technique aims at repelling iterates from the boundaries of Ω to avoid premature convergence to nonstationary points. Other techniques to handle the bound constraints include penalty terms and augmented Lagrangian (see, e.g., Chapter 17 in [10]), but they do not ensure feasibility of the iterates.

We choose to adopt an alternative to these approaches. The principle consists of computing the steepest admissible direction \mathbf{d}^k at the current iterate \mathbf{p}^k , and then setting $\mathbf{p}^{k+1} = \widehat{R}_{\mathbf{p}}(\alpha^k \mathbf{d}^k)$ where α^k is selected by means of a line-search technique. In this context, by admissible direction, we mean the following.

Definition 3.1. An *admissible direction* at $\mathbf{p} \in \Omega$ is a vector $\mathbf{d} \in T_{\mathbf{p}}\mathcal{E}^{n-1}$ for which there exists $\epsilon > 0$ such that $\widehat{R}_{\mathbf{p}}(\alpha \mathbf{d})$ belongs to Ω for all $\alpha \in [0, \epsilon]$.

The computation of the steepest admissible direction will be presented in Section 4 and the line-search technique will be discussed in Section 5.3

We already mention that, in order to be functional in combination with the chosen basic Armijo-type line-search technique, the proposed approach requires a robustification mechanism, described in Section 5.2, both for the capacity constraints (2) and for the nondifferentiable points (7). Otherwise, since the nondifferentiable points and boundary points form a zero-measure set, all the iterates would be likely to be considered differentiable points that strictly satisfy the bound constraints (2), with the consequence that the computed steepest admissible direction would merely be the projected gradient (20) at all times, yielding unsatisfactory convergence as illustrated by the squares in Fig. 9.

4. Subgradient descent for the ELDP

This section chiefly concerns the computation of the steepest admissible direction for the ELDP (5). We will proceed gradually. The concept of subgradient is first recalled in an unconstrained setting in Section 4.1, then it is generalized to the Riemannian setting in Section 4.2 with a view towards considering the ELDP cost function on the ellipsoid manifold \mathcal{E}^{n-1} , which is done in Section 4.3, before reintroducing the bound constraints in Section 4.4.

As stated in Section 2.3, the cost function $f_T(\mathbf{p})$ is piecewise smooth, i.e., differentiable almost everywhere. Let S , resp. D , denote the set of nondifferentiable points, resp. differentiable points, of f_T in Ω :

$$S = \left\{ \mathbf{p} \in \Omega : \exists i \text{ s.t. } p_i = p_i^{\min} + \frac{k\pi}{e_i}, k \in \mathbb{N} \right\}, \quad D = \Omega \setminus S.$$

We will also consider the set S^0 , resp. D^0 , of points of S , resp. D , that are not on the boundary of the box (2). Note that in ELDP instances, one must expect that both S^0 and $S \setminus S^0$ are nonempty, and likewise for D^0 and $D \setminus D^0$.

For $\mathbf{p} \in D^0$, the steepest admissible direction $\text{grad}f_T(\mathbf{p})$ is simply the projection of the gradient of $f_T(\mathbf{p})$ onto $T_{\mathbf{p}}\mathcal{E}^{n-1}$:

$$\text{grad}f_T(\mathbf{p}) = P_{\mathbf{p}}(\nabla f_T(\mathbf{p})), \quad (20)$$

where $P_{\mathbf{p}}$ is given by (14) and $\nabla f_T(\mathbf{p})$ by (8). For all other points $\mathbf{q} \in \Omega \setminus D^0$, describing the steepest admissible direction requires more sophisticated developments that we present in the rest of this section.

4.1. Subgradient descent

Given a nonsmooth, nonconvex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, Clarke's generalized directional derivative is given by:

$$f^\circ(\mathbf{x}; \mathbf{d}) = \limsup_{\mathbf{y} \rightarrow \mathbf{x}, t \downarrow 0} \frac{f(\mathbf{y} + t\mathbf{d}) - f(\mathbf{y})}{t}.$$

The generalized gradient of f , noted $\partial f(\mathbf{x})$, is then defined as the set of the subgradients \mathbf{s} of $f^\circ(\mathbf{x}; \cdot)$:

$$\partial f(\mathbf{x}) = \left\{ \mathbf{s} \in \mathbb{R}^n : f^\circ(\mathbf{x}; \mathbf{v}) \geq \mathbf{v}^\top \mathbf{s}, \forall \mathbf{v} \in \mathbb{R}^n \right\}.$$

For a general nonsmooth function, this set can be difficult to describe in practice. However, if the function is locally Lipschitz, which is the case of the ELDP cost function (1), then the generalized gradient at \mathbf{x} is the convex hull of all points \mathbf{s} of the form

$$\mathbf{s} = \lim_{i \rightarrow \infty} \nabla f(\mathbf{x}_i),$$

where $\{\mathbf{x}_i\}$ is a sequence converging to \mathbf{x} such that f is differentiable at each \mathbf{x}_i (see [25, Proposition 5]). If moreover, as is the case of the ELDP cost function (1), $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be expressed as the pointwise maximum of m smooth functions

$f_j : \mathbb{R}^n \rightarrow \mathbb{R}$, i.e.,

$$f(\mathbf{x}) = \max_{j=1,\dots,m} f_j(\mathbf{x}), \quad (21)$$

then the generalized gradient can be simply described as

$$\partial f(\mathbf{x}) = \text{co} \{ \nabla f_j(\mathbf{x}) \mid j \in \mathcal{I}_f(\mathbf{x}) \}, \quad (22)$$

where $\text{co}\{\cdot\}$ denotes the convex hull and $\mathcal{I}_f(\mathbf{x})$ denotes the set of indices for which the maximum in (21) is attained, i.e.,

$$\mathcal{I}_f(\mathbf{x}) = \{j \in \{1, \dots, m\} \mid f(\mathbf{x}) = f_j(\mathbf{x})\}. \quad (23)$$

Using this framework, and without considering any constraint for the time being, the steepest descent direction \mathbf{d}^k is given by the opposite of the shortest vector in $\partial f(\mathbf{x}^k)$ [26, Lemma 2.1], which can be obtained by solving the following quadratic programming problem:

$$\min_{\substack{\lambda_j \geq 0 \\ \sum \lambda_j = 1}} \left\| \sum_{j \in \mathcal{I}_f(\mathbf{x}^k)} \lambda_j \nabla f_j(\mathbf{x}^k) \right\|^2. \quad (24)$$

In practice, when the set $\partial f(\mathbf{x}^k)$ is composed of a single element $\nabla f_j(\mathbf{x}^k)$, the subproblem becomes trivial as the descent direction is simply given by $\mathbf{d}^k = -\nabla f_j(\mathbf{x}^k)$. When $\partial f(\mathbf{x}^k)$ contains at least two elements, solving the subproblem allows us to compute a descent direction even though the function is not differentiable.

4.2. Riemannian subgradient descent

In order to extend the subgradient descent to a Riemannian manifold setting, one has to redefine the needed ingredients. Let \mathcal{M} be a submanifold of \mathbb{R}^n and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a piecewise smooth function, defined as the maximum of m smooth functions f_j . Using the tools presented in Section 3.2, the projected gradient of each function f_j at a point $\mathbf{x} \in \mathcal{M}$ is defined as follows:

$$\text{grad} f_j(\mathbf{x}) = P_{\mathbf{x}}(\nabla f_j(\mathbf{x})) \in T_{\mathbf{x}}\mathcal{M}.$$

The generalized projected gradient is then given by:

$$\begin{aligned} \text{grad} f(\mathbf{x}) &= \text{co} \{ \text{grad} f_j(\mathbf{x}) \mid j \in \mathcal{I}_f(\mathbf{x}) \} \\ &= \text{co} \{ P_{\mathbf{x}}(\nabla f_j(\mathbf{x})) \mid j \in \mathcal{I}_f(\mathbf{x}) \}. \end{aligned} \quad (25)$$

The steepest admissible direction \mathbf{d}^k is obtained by computing the shortest vector in $\text{grad} f(\mathbf{x}^k)$, solving the following quadratic program:

$$\begin{aligned} \min_{\substack{\lambda_j \geq 0 \\ \sum \lambda_j = 1}} \left\| \sum_{j \in \mathcal{I}_f(\mathbf{x}^k)} \lambda_j \text{grad} f_j(\mathbf{x}^k) \right\|^2 &= \min_{\substack{\lambda_j \geq 0 \\ \sum \lambda_j = 1}} \left\| \sum_{j \in \mathcal{I}_f(\mathbf{x}^k)} \lambda_j P_{\mathbf{x}^k}(\nabla f_j(\mathbf{x}^k)) \right\|^2 \\ &= \min_{\substack{\lambda_j \geq 0 \\ \sum \lambda_j = 1}} \left\| P_{\mathbf{x}^k} \left(\sum_{j \in \mathcal{I}_f(\mathbf{x}^k)} \lambda_j \nabla f_j(\mathbf{x}^k) \right) \right\|^2. \end{aligned} \quad (26)$$

The steepest-descent admissible direction is computed as follows:

$$\mathbf{d}^k = -P_{\mathbf{x}^k} \left(\sum_{j \in \mathcal{I}_f(\mathbf{x}^k)} \lambda_j^* \nabla f_j(\mathbf{x}^k) \right),$$

where λ^* is the solution of (26).

4.3. Application to the ELDP cost function on \mathcal{E}^{n-1}

In this subsection, we adapt the subgradient descent scheme to the ELDP, temporarily ignoring the bound constraints (2) until Section 4.4. We first show that the ELDP cost function f_T (1) can be expressed as the pointwise maximum of smooth functions, then we compute the associated generalized gradient and we formulate the search for a descent direction as a quadratic programming problem.

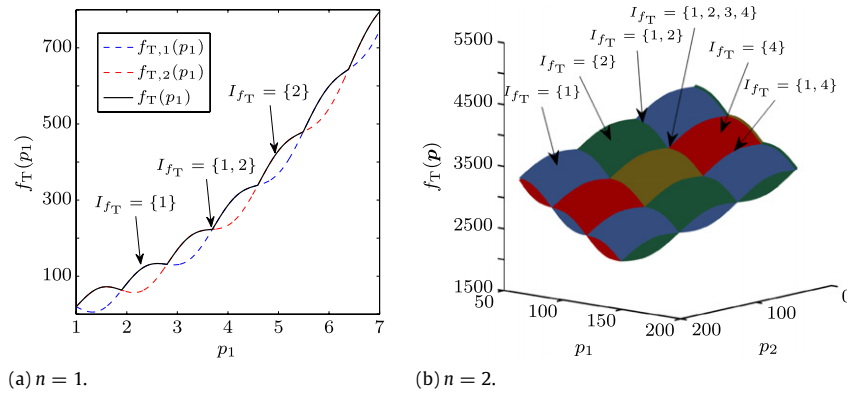


Fig. 7. The ELDP cost function $f_T(\mathbf{p})$ as the pointwise maximum of 2^n functions $f_{T,j}(\mathbf{p})$.

Recalling that the ELDP cost function $f_T(\mathbf{p})$ is the sum of n functions $f^i(p_i)$ ($i = 1, \dots, n$), let $\hat{f}^i(\mathbf{p}) \doteq f^i(p_i)$. Each of these functions can be decomposed into two components, $\hat{f}_q^i(\mathbf{p}) = a_i p_i^2 + b_i p_i + c_i$ (the quadratic term) and $\hat{f}_s^i(\mathbf{p}) = d_i \sin[e_i(p_i^{\min} - p_i)]$ (the sinusoidal term). The ELDP cost function can then be written as follows:

$$\begin{aligned} f_T(\mathbf{p}) &= \sum_{i=1}^n \hat{f}_q^i(\mathbf{p}) + \left| \hat{f}_s(\mathbf{p}) \right| \\ &= \sum_{i=1}^n \max \left(\hat{f}_q^i(\mathbf{p}) - \hat{f}_s^i(\mathbf{p}), \hat{f}_q^i(\mathbf{p}) + \hat{f}_s^i(\mathbf{p}) \right). \end{aligned} \quad (27)$$

Clearly, f_T can be expressed as the maximum of 2^n functions $f_{T,j}$, taking all the possible combinations of the maximum arguments in (27):

$$\begin{aligned} f_T(\mathbf{p}) &= \max_{j=1, \dots, 2^n} f_{T,j}(\mathbf{p}) \\ &= \max_{j=1, \dots, 2^n} \left(\sum_{i=1}^n \hat{f}_q^i(\mathbf{p}) + \sum_{i=1}^n (-1)^{\lfloor \frac{j-2^i}{2^{i-1}} \rfloor} \hat{f}_s^i(\mathbf{p}) \right) \\ &= f_Q(\mathbf{p}) + \max_{j=1, \dots, 2^n} f_{S,j}(\mathbf{p}), \end{aligned}$$

where $a \% b$ denotes the remainder of the division a/b .

The set of indices $\mathcal{I}_{f_T}(\mathbf{p}) = \{j \in \{1, \dots, 2^n\} \mid f_T(\mathbf{p}) = f_{T,j}(\mathbf{p})\}$ can now be described. Let $\mathcal{S}(\mathbf{p})$ be the set of indices of the entries p_i in \mathbf{p} that cancel the sine term, and let $\mathcal{F}(\mathbf{p})$ be the other indices of \mathbf{p} :

$$\begin{aligned} \mathcal{S}(\mathbf{p}) &= \{s_1, \dots, s_{n_s}\} \\ &= \left\{ i \in \{1, \dots, n\} \mid p_i = p_i^{\min} + \frac{k\pi}{e_i}, k \in \mathbb{N} \right\}, \end{aligned} \quad (28)$$

$$\begin{aligned} \mathcal{F}(\mathbf{p}) &= \{f_1, \dots, f_{n_f}\} \\ &= \{1, \dots, n\} \setminus \mathcal{S}(\mathbf{p}). \end{aligned} \quad (29)$$

As depicted in Fig. 7, the cardinality of $\mathcal{I}_{f_T}(\mathbf{p})$ is then given by $|\mathcal{I}_{f_T}(\mathbf{p})| = 2^{n_s}$. If $\mathcal{S}(\mathbf{p})$ is empty, only one function $f_{T,j}$ has to be considered, thus $|\mathcal{I}_{f_T}(\mathbf{p})| = 2^0 = 1$.

As per (25), the generalized projected gradient of f_T at \mathbf{p} is now defined as follows:

$$\begin{aligned} \text{grad } f_T(\mathbf{p}) &= \text{co} \{ \text{grad } f_{T,j}(\mathbf{p}) \mid j \in \mathcal{I}_{f_T}(\mathbf{p}) \} \\ &= P_{\mathbf{p}}(\nabla f_Q(\mathbf{p})) + \text{co} \{ P_{\mathbf{p}}(\nabla f_{S,j}(\mathbf{p})) \mid j \in \mathcal{I}_{f_T}(\mathbf{p}) \}. \end{aligned} \quad (30)$$

We now consider the quadratic subproblem (26) for the ELDP to determine the descent direction \mathbf{d} at a point $\mathbf{p} \in \mathcal{E}^{n-1}$:

$$\min_{\substack{\lambda_j \geq 0 \\ \sum \lambda_j = 1}} \left\| P_{\mathbf{p}} \left(\sum_{j \in \mathcal{I}_{f_T}(\mathbf{p})} \lambda_j \nabla f_{T,j}(\mathbf{p}) \right) \right\|^2. \quad (31)$$

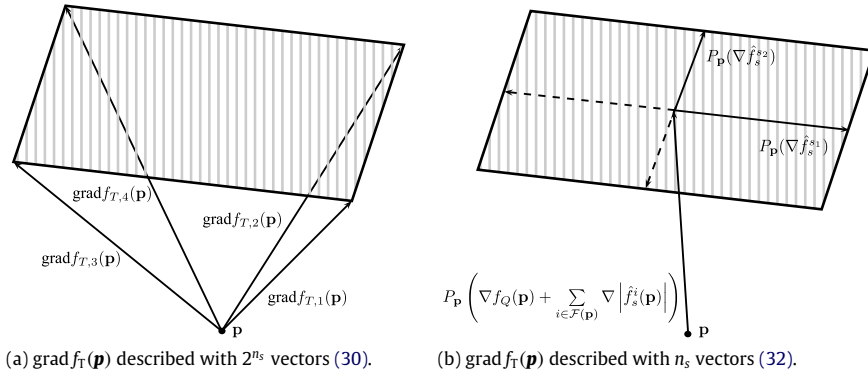


Fig. 8. Illustration of the generalized gradient (dashed area). The description of $\text{grad} f_T(\mathbf{p})$ (30) using 2^{n_s} vectors and the description (32) using n_s vectors are equivalent.

This is a quadratic program with 2^{n_s} variables λ_j . However, as illustrated in Fig. 8, the number of variables can be drastically reduced since the generalized projected gradient $\text{grad} f_T(\mathbf{p})$ can actually be described with only n_s vectors. Indeed, noting that $|\hat{f}_s^i(\mathbf{p})|$ is differentiable for $i \in \mathcal{F}(\mathbf{p})$ and recalling that f_T is a separable function, $\text{grad} f_T(\mathbf{p})$ can be advantageously rewritten in the following way:

$$\text{grad} f_T(\mathbf{p}) = P_{\mathbf{p}}(\nabla f_Q(\mathbf{p})) + \sum_{i \in \mathcal{F}(\mathbf{p})} P_{\mathbf{p}}(\nabla |\hat{f}_s^i(\mathbf{p})|) + \left\{ \sum_{i \in \mathcal{S}(\mathbf{p})} \lambda_i P_{\mathbf{p}}(\nabla \hat{f}_s^i(\mathbf{p})) \mid \lambda_i \in [-1, 1] \right\}. \quad (32)$$

The quadratic subproblem (31) hence becomes:

$$\min_{\lambda_i \in [-1, 1]} \left\| P_{\mathbf{p}} \left(\nabla f_Q(\mathbf{p}) + \sum_{i \in \mathcal{F}(\mathbf{p})} \nabla |\hat{f}_s^i(\mathbf{p})| + \sum_{i \in \mathcal{S}(\mathbf{p})} \lambda_i \nabla \hat{f}_s^i(\mathbf{p}) \right) \right\|^2. \quad (33)$$

Introducing the following notation:

$$\mathbf{S} = [P_{\mathbf{p}}(\nabla \hat{f}_s^{s_1}(\mathbf{p})) \cdots P_{\mathbf{p}}(\nabla \hat{f}_s^{s_{n_s}}(\mathbf{p}))], \quad (34)$$

$$\mathbf{g} = P_{\mathbf{p}} \left(\nabla f_Q(\mathbf{p}) + \sum_{i \in \mathcal{F}(\mathbf{p})} \nabla |\hat{f}_s^i(\mathbf{p})| \right), \quad (35)$$

the quadratic subproblem can be rewritten as a general bound-constrained convex quadratic program of the form:

$$\begin{aligned} \lambda^* &= \arg \min_{\lambda} \|\mathbf{g} + \mathbf{S}\lambda\|^2 \\ &= \arg \min_{\lambda} \lambda^{\top} \mathbf{S}^{\top} \mathbf{S} \lambda + 2\lambda^{\top} \mathbf{S}^{\top} \mathbf{g} \\ &\text{subject to } -1 \leq \lambda \leq 1, \end{aligned} \quad (36)$$

where $\mathbf{S}^{\top} \mathbf{S}$ is positive semidefinite (positive definite if \mathbf{S} has full rank). The descent direction \mathbf{d} is then computed as $\mathbf{d} = -(\mathbf{g} + \mathbf{S}\lambda^*)$.

4.4. Including the bound constraints

Solving the subproblem (36) does not always provide an admissible descent direction for points on the boundary of the feasible set Ω . Indeed, if the resulting direction \mathbf{d} points outside Ω , no progress can be made by following it. However, the bound constraints can be incorporated into the quadratic problem in order to ensure an admissible descent direction for all points in Ω . Let $l_i(\mathbf{p})$ and $u_i(\mathbf{p})$, $i = 1, \dots, n$, denote the lower and upper bound constraints:

$$\begin{aligned} l_i(\mathbf{p}) &= -p_i + p_i^{\min} \leq 0, \\ u_i(\mathbf{p}) &= p_i - p_i^{\max} \leq 0. \end{aligned}$$

Let $\mathcal{L}(\mathbf{p})$ and $\mathcal{U}(\mathbf{p})$ be the indices of the active constraints, and let $\mathcal{B}(\mathbf{p})$ be their union:

$$\begin{aligned}\mathcal{L}(\mathbf{p}) &= \{l_1, \dots, l_{n_l}\} \\ &= \{i \in \{1, \dots, n\} \mid l_i(\mathbf{p}) = 0\}, \\ \mathcal{U}(\mathbf{p}) &= \{u_1, \dots, u_{n_u}\} \\ &= \{i \in \{1, \dots, n\} \mid u_i(\mathbf{p}) = 0\}, \\ \mathcal{B}(\mathbf{p}) &= \mathcal{L}(\mathbf{p}) \cup \mathcal{U}(\mathbf{p}).\end{aligned}\tag{37}$$

Noting that $\nabla l_i(\mathbf{p}) = -\mathbf{e}_i$ and $\nabla u_i(\mathbf{p}) = \mathbf{e}_i$, we introduce the following notation:

$$\mathbf{B} = \left[P_{\mathbf{p}}(-\mathbf{e}_{l_1}) \cdots P_{\mathbf{p}}(-\mathbf{e}_{l_{n_l}}) \mid P_{\mathbf{p}}(\mathbf{e}_{u_1}) \cdots P_{\mathbf{p}}(\mathbf{e}_{u_{n_u}}) \right].\tag{38}$$

The search for an admissible descent direction can now include the projected gradient of the active constraints and can thus be expressed as the following quadratic program:

$$\begin{aligned}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= \arg \min_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \|\mathbf{g} + \mathbf{S}\boldsymbol{\lambda} + \mathbf{B}\boldsymbol{\mu}\|^2 \\ \text{subject to } &\begin{cases} -1 \leq \lambda_i \leq 1, \\ \mu_i \geq 0. \end{cases}\end{aligned}\tag{39}$$

This is again a convex quadratic problem of dimension $|\mathcal{L}(\mathbf{p}) \cup \mathcal{B}(\mathbf{p})| \leq n$. The steepest-descent admissible direction is finally computed as follows:

$$\mathbf{d} = -(\mathbf{g} + \mathbf{S}\boldsymbol{\lambda}^* + \mathbf{B}\boldsymbol{\mu}^*).\tag{40}$$

4.5. First-order stationarity condition

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a nonsmooth (Lipschitz) function to be minimized subject to a set of constraints $c_i(\mathbf{x}) \leq 0$ ($i = 1, \dots, m$). The first order necessary stationarity conditions for this constrained nonsmooth problem can be stated as follows (see e.g. [25, Section 3]):

Theorem 4.1. *If \mathbf{x}^* minimizes $f(\mathbf{x})$ locally, then there exist numbers μ_i ($i = 1, \dots, m$) such that:*

- (a) $\mu_i \geq 0$,
- (b) $\mu_i c_i(\mathbf{x}^*) = 0$,
- (c) $\mathbf{0} \in \partial f(\mathbf{x}^*) + \sum_{i=1}^m \mu_i \nabla c_i(\mathbf{x}^*)$.

Applying this theorem to the ELDP, we obtain the following characterization of its local minimizers:

Theorem 4.2. *If \mathbf{p}^* minimizes $f_{\Gamma}(\mathbf{p})$ locally in Ω , then there exist numbers μ_i^-, μ_i^+ ($i = 1, \dots, n$) such that:*

- (a) $\mu_i^- \geq 0, \mu_i^+ \geq 0$,
- (b) $\mu_i^- l_i(\mathbf{p}^*) = 0, \mu_i^+ u_i(\mathbf{p}^*) = 0$,
- (c) $\mathbf{0} \in \text{grad } f_{\Gamma}(\mathbf{p}^*) + \sum_{i=1}^n (\mu_i^- \text{grad}(l_i(\mathbf{p}^*)) + \mu_i^+ \text{grad}(u_i(\mathbf{p}^*)))$.

The technique proposed in the previous section to compute admissible descent directions offers an important advantage over the classical alternatives (barrier term, augmented Lagrangian, etc.). Computing the direction $\mathbf{d} = -(\mathbf{g} + \mathbf{S}\boldsymbol{\lambda}^* + \mathbf{B}\boldsymbol{\mu}^*)$ at a local minimizer point \mathbf{p}^* yields the zero vector. Therefore, the norm of the direction \mathbf{d} can be used as a stopping criterion for the descent algorithm, ensuring that it will stop at a (numerically) stationary point.

5. Implementation

In the previous sections, all the ingredients needed to perform local optimization of $f_{\Gamma}(\mathbf{p})$ in Ω were introduced. In this section, we present how, given a feasible initial iterate $\mathbf{p}^0 \in \Omega$, the Riemannian subgradient descent method is applied to produce a sequence of iterates $\mathbf{p}^k \in \Omega$. The implementation details are now given.

5.1. Generating a feasible iterate

Recalling the geometrical considerations presented in Section 2.2, the feasible set Ω is the intersection of an axis-aligned box and an ellipsoid surface. From there, a simple approach to generate a feasible point $\mathbf{p} \in \Omega$ is given by Algorithm 1. The loop is needed because the scaling performed by R may output a point outside the bound constraints. Note that Algorithm 1 becomes superfluous if the proposed local method is combined with a feasible global method (i.e, with iterates on Ω).

5.2. Computing the descent direction

The steepest-descent admissible direction at a point \mathbf{p}^k is described by (40). It can be computed, provided that the sets $\mathcal{L}(\mathbf{p}^k)$ and $\mathcal{B}(\mathbf{p}^k)$ are available. However, in practice, it is numerically unlikely for an iterate \mathbf{p}^k to ever cancel exactly any of

Algorithm 1 Feasible point generation**Output:** $\mathbf{p} \in \Omega$, a feasible point**repeat**Draw \mathbf{x} from the uniform distribution on
 $[p_1^{\min}, p_1^{\max}] \times \cdots \times [p_n^{\min}, p_n^{\max}]$ $\mathbf{p} \leftarrow R_{\mathbf{x}}(\mathbf{0})$, where R is given by formula (15)**until** $\mathbf{p} \in \Omega$ **return** \mathbf{p}

the sine terms in $f_T(\mathbf{p}^k)$. Therefore, $\mathcal{S}(\mathbf{p}^k)$ and its complement $\mathcal{F}(\mathbf{p}^k)$ are approximated in the following way:

$$\mathcal{S}_{\epsilon}(\mathbf{p}) = \left\{ i \in \{1, \dots, n\} : \left| p_i - \left(p_i^{\min} + \frac{k\pi}{e_i} \right) \right| \leq \epsilon \right\}, \quad (41)$$

$$\mathcal{F}_{\epsilon}(\mathbf{p}) = \{1, \dots, n\} \setminus \mathcal{S}_{\epsilon}(\mathbf{p}). \quad (42)$$

Similarly, the bound constraints indices are approximated as follows:

$$\mathcal{B}_{\epsilon}(\mathbf{p}) = \{i \in \{1, \dots, n\} \mid l_i(\mathbf{p}) \geq -\epsilon \text{ or } u_i(\mathbf{p}) \geq -\epsilon\}. \quad (43)$$

An “ ϵ -steepest-descent ϵ -admissible direction” is then computed at each iteration, using Algorithm 2. In its nondifferentiable-related aspect, this approach is reminiscent of Goldstein’s ϵ -generalized gradient [27]. In its bound-related aspect, it has an active-set flavor.

Algorithm 2 DESCENTDIRECTION(\mathbf{p} , $\mathcal{S}_{\epsilon}(\mathbf{p})$, $\mathcal{F}_{\epsilon}(\mathbf{p})$, $\mathcal{B}_{\epsilon}(\mathbf{p})$) - Descent direction**Input:** $\mathbf{p} \in \Omega$, $\mathcal{S}_{\epsilon}(\mathbf{p})$, $\mathcal{F}_{\epsilon}(\mathbf{p})$, $\mathcal{B}_{\epsilon}(\mathbf{p})$ **Output:** \mathbf{d} , an admissible descent directionCompute \mathbf{S} , \mathbf{g} and \mathbf{B} at \mathbf{p} according to (34), (35) and (38), in which $\mathcal{S}(\mathbf{p})$, $\mathcal{F}(\mathbf{p})$, $\mathcal{B}(\mathbf{p})$ are replaced by $\mathcal{S}_{\epsilon}(\mathbf{p})$, $\mathcal{F}_{\epsilon}(\mathbf{p})$, $\mathcal{B}_{\epsilon}(\mathbf{p})$ Compute (λ^*, μ^*) solving the quadratic programming sub-problem (39) $\mathbf{d} \leftarrow -(\mathbf{g} + \mathbf{S}\lambda^* + \mathbf{B}\mu^*)$ **return** \mathbf{d}

5.3. Computing the step size

Once a descent admissible direction has been computed, the step size must be determined. We propose to adopt Armijo’s rule with two minor modifications. First, the initial step size is retained from the previous step size computation. Second, before performing classical backtracking, a few trials are performed by expanding the previous step size. These two modifications provide possible faster convergence of the descent algorithm by allowing bigger step sizes. The line search uses the retraction \hat{R} defined by (19). In practice, given a vector ξ , the corresponding set \mathcal{C} involved in (19) is approximated as follows:

$$\mathcal{C}_{\epsilon} = \{i \in \{1, \dots, n\} \mid |\xi_i| \leq \epsilon\}. \quad (44)$$

The computation of the step size is described in Algorithm 3.

5.4. Subgradient descent for the ELDP

The subgradient descent algorithm can now be completely described and is given by Algorithm 4. In the taxonomy of optimization methods [10], the proposed approach can be considered an active-set-like feasible ϵ -subgradient-descent method for nonsmooth nonconvex problems with both equality and bound constraints.

6. Numerical experiments

The Riemannian subgradient descent algorithm presented in the previous section was implemented in MATLAB. The data from four instances of ELDP with valve-point effect were collected, for dimensions $n = 3, 5, 6$ and 15 (sources: [28–31]). (We are not aware of publicly-available ELDP instances with valve-point effect and a quadratic loss function for which n is greater than 15.) These data sets contain the following information for each instance:

- the coefficients of f_T for the quadratic term:

$$a_i \text{ [\$/(h} \cdot \text{MW}^2)], \quad b_i \text{ [\$/(h} \cdot \text{MW)]}, \quad c_i \text{ [\$/h]},$$

- the coefficients of f_T for the sine term:

$$d_i \text{ [\$/h]}, \quad e_i \text{ [1/MW]},$$

- the bound constraints: p_i^{\min}, p_i^{\max} [MW],

Algorithm 3 STEPSIZE(\mathbf{p} , \mathbf{g} , α_0 , β , γ , n_f , n_b) - Line search using Armijo's rule

Input: $\mathbf{p} \in \Omega$, \mathbf{g} search direction, α_0 initial step size, β sufficient decrease constant, γ step size shrinking factor, n_f number of forward steps, n_b number of backtracking steps

Output: α , a step size ensuring sufficient decrease

Consider $\tilde{f}_T(\mathbf{p}) = \begin{cases} f_T(\mathbf{p}) & \text{if (2),} \\ +\infty & \text{otherwise.} \end{cases}$

Step 1 (Forward exploration):

```

for  $s = n_f \rightarrow 1$  do
  if  $\tilde{f}_T(\hat{R}_p(\gamma^{-s} \frac{\mathbf{g}}{\|\mathbf{g}\|})) < f_T(\mathbf{p}) - \beta \alpha_0 \gamma^{-s} \|\mathbf{g}\|$  then
    return  $\alpha_0 \gamma^{-s}$ 
  end if
end for

```

Step 2 (Backtracking):

```

for  $s = 0 \rightarrow n_b$  do
  if  $\tilde{f}_T(\hat{R}_p(\gamma^s \frac{\mathbf{g}}{\|\mathbf{g}\|})) < f_T(\mathbf{p}) - \beta \alpha_0 \gamma^s \|\mathbf{g}\|$  then
    return  $\alpha_0 \gamma^s$ 
  end if
end for
return 0

```

Algorithm 4 Riemannian Subgradient Descent for the ELDP

Input: $\mathbf{p}^0 \in \Omega$, ϵ singular point detection tolerance, δ descent direction norm tolerance, β sufficient decrease constant, γ step size shrinking factor, n_f number of forward steps, n_b number of backtracking steps

Output: $\mathbf{p}^* \in \Omega$, a generalized stationary point

Set $k = 0$, $\alpha^0 = 1$

Step 1 (Compute the admissible descent direction)

```

Compute  $\mathcal{J}_\epsilon(\mathbf{p}^k)$ ,  $\mathcal{F}_\epsilon(\mathbf{p}^k)$  and  $\mathcal{B}_\epsilon(\mathbf{p}^k)$  (41)–(43)
 $\mathbf{d}^k \leftarrow \text{DESCENTDIRECTION}(\mathbf{p}^k, \mathcal{J}_\epsilon(\mathbf{p}^k), \mathcal{F}_\epsilon(\mathbf{p}^k), \mathcal{B}_\epsilon(\mathbf{p}^k))$ 
if  $\|\mathbf{d}^k\| < \delta$  then
  return  $\mathbf{p}^k$ 
end if
 $\mathbf{g}^k \leftarrow \mathbf{d}^k / \|\mathbf{d}^k\|$ 

```

Step 2 (Compute the step size)

$\alpha^k \leftarrow \text{STEPSIZE}(\mathbf{p}^k, \mathbf{g}^k, \alpha^{k-1}, \beta, \gamma, n_f, n_b)$

Step 3 (Compute the new iterate)

```

if  $\alpha^k > 0$  then
   $\mathbf{p}^{k+1} \leftarrow \hat{R}_{\mathbf{p}^k}(\alpha^k \mathbf{g}^k)$ 
   $k \leftarrow k + 1$ 
  Go to Step 1
else
  return  $\mathbf{p}^k$ 
end if

```

- the power demand: p_d [MW],
- the transmission loss coefficients:

$$\mathbf{B} \text{ [1/MW]}, \quad \mathbf{B}^0 \text{ [-]}, \quad B^{00} \text{ [MW]}.$$

An important remark about these data sets is that for the cases $n = 5$ and $n = 6$, only the first two dimensions present the valve-point effect ($d_i = e_i = 0$, $i \geq 2$). The consequence is that the multimodal aspect for these two data sets is much less pronounced compared to the two other data sets, for which the valve-point effect is present in all dimensions. The MATLAB code and data sets are available online at <http://www.inma.ucl.ac.be/~borckmans/ELDP>. In the following experiments, the default parameters were chosen as follows: $\epsilon = 10^{-8}$, $\beta = 10^{-4}$, $\gamma = 0.5$, $n_f = 3$, $n_b = 50$, $\delta = 10^{-12}$.

6.1. Local convergence

The trajectories generated by Algorithm 4 for different starting points are illustrated in Fig. 9 for the case $n = 3$. Whenever the sequence of iterates approaches a nondifferentiable point, the projected subgradients are computed, and the

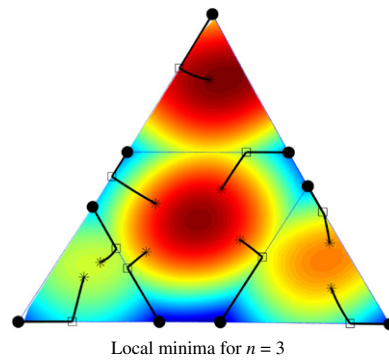


Fig. 9. Local convergence illustration for $n = 3$. The trajectories from 9 starting points (stars) are depicted. The final iterates obtained using a naive steepest-descent algorithm (squares) and the proposed Riemannian subgradient algorithm (dots) are represented. The dashed blue lines correspond to the nondifferentiable sub-ellipsoids.

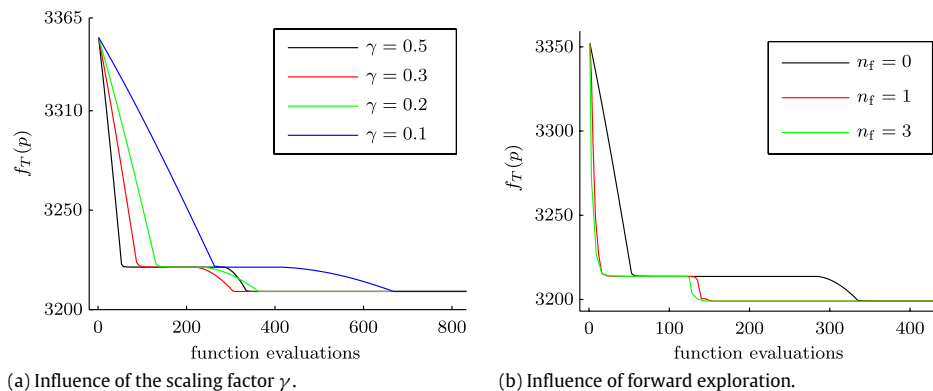


Fig. 10. Local convergence properties for $n = 3$.

descent direction resulting from the quadratic subproblem follows the corresponding sub-ellipsoid. These sub-ellipsoids are represented as dashed lines in Fig. 9. For comparison purposes, applying a simple Riemannian gradient descent (ignoring the nondifferentiable aspect of f_T) from the same starting points yields convergence to nonstationary points.

Algorithm 4 was tested by performing a large number of trials for each ELDP instance, starting from random points. The final iterates of the trials were tested for the necessary stationarity condition; the norm of the final descent direction, as computed by Algorithm 2 was smaller than 10^{-12} for all the trials. Considering two points to be different if their corresponding costs are at least 10^{-8} apart from each other, the proposed algorithm was able to consistently converge to a small set of potential local minima for the cases $n = 3, 5$ and 6 (9 minima for $n = 3$, 7 minima for $n = 5$, and 7 minima for $n = 6$). As mentioned earlier, the small number of local minima for the cases $n = 5$ and $n = 6$ is due to the limited valve-point effect. On the other hand, for the case $n = 15$, the number of local minima is much higher since the valve-point effect affects all the dimensions. After 10^6 trials, the proposed approach consistently identified solutions in a set of around 2×10^5 points.

6.2. Parameter influence

As depicted in Fig. 10(a) and (b), the convergence speed is affected by the line search scaling factor γ and is enhanced when reusing the previous successful step size and allowing forward exploration (n_f steps).

Another important parameter of the proposed algorithm is ϵ , the tolerance for detecting the singular entries in (41). Smaller values for ϵ lead to better precision but at the cost of taking more function evaluations to detect singularities. This can be observed in Fig. 11, where the plateaus correspond to iterates approaching singular sub-ellipsoids of Ω .

6.3. Global exploration

The Riemannian subgradient descent presented in this paper is inherently a local optimizer. It is very efficient at computing a local minimizer in the vicinity of a given starting point in Ω , but it is not meant to roam around the cost function landscape in search of the global minimum. On the other hand, the global heuristics that are largely explored in the

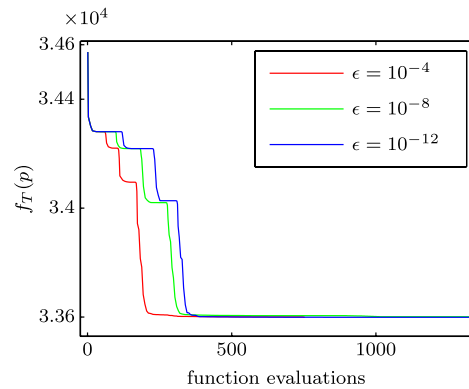


Fig. 11. Influence of ϵ ($n = 15$).

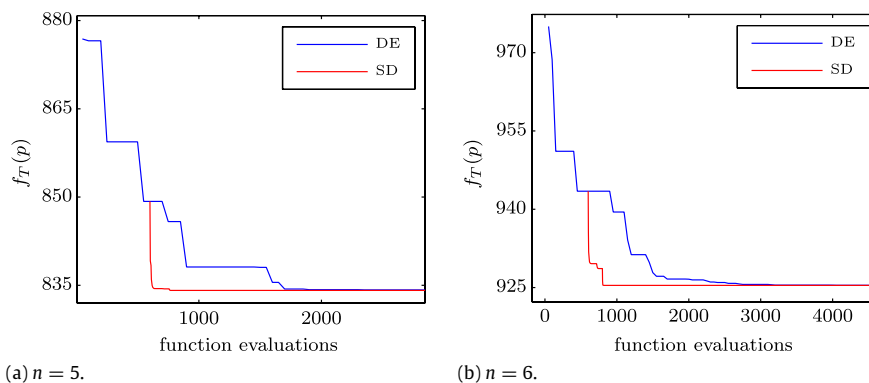


Fig. 12. Using the proposed subgradient descent (SD) algorithm as a post processing tool. Stopping the DE algorithm progression at some point and refining the output using the proposed SD algorithm (red curve) provides faster convergence than DE alone (blue curve). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

literature have very complementary properties: they often provide good exploration of the search space but they tend to struggle to refine the final iterate and they do not attempt to check stationarity (and even feasibility in some cases) of their output. This calls for hybrid algorithms that combine the best of both approaches.

In order to illustrate this idea, we implemented the Differential Evolution (DE) algorithm [32] as proposed in [33]. This population-based heuristic is known to provide reasonably good exploration of the search space, relying quite heavily on random updates, combining selected individuals. As in many heuristics proposed to solve the ELDP, the authors of [33] use the concept of a slack variable to respect the equality constraint (3).

A first approach to hybridize DE with our algorithm, known as the sequential approach [34], is to stop the global exploration at some point before applying the subgradient descent (Algorithm 4). This post-processing step avoids the slow refinement of the solution and produces in practice a final iterate that satisfies the stationary conditions from Theorem 4.2. The typical behavior of such an approach is depicted in Fig. 12.

Another option is to integrate the subgradient descent within the global technique, following the cyclical approach mentioned in [34]. The idea is to refine the best individual produced by the DE algorithm at each iteration using the projected subgradient descent. The benefit of this approach is illustrated in Fig. 13.

7. Conclusion

In this paper, we showed how the geometrical structure of the ELDP (5) can be exploited to perform local optimization of its associated cost function. The canvas of optimization on Riemannian manifolds was used to maintain feasible iterates at all times. The nondifferentiable nature of the cost function was handled using Clarke's generalized calculus. A simple procedure was presented to compute an admissible descent direction and to perform a curvilinear search along that direction while satisfying the constraints. The resulting local method is unique in that it possesses the combination of desirable properties mentioned in the introduction. Finally, we showed that the multimodal aspect of the ELDP can be addressed by hybridizing the proposed local method with a global optimization technique.

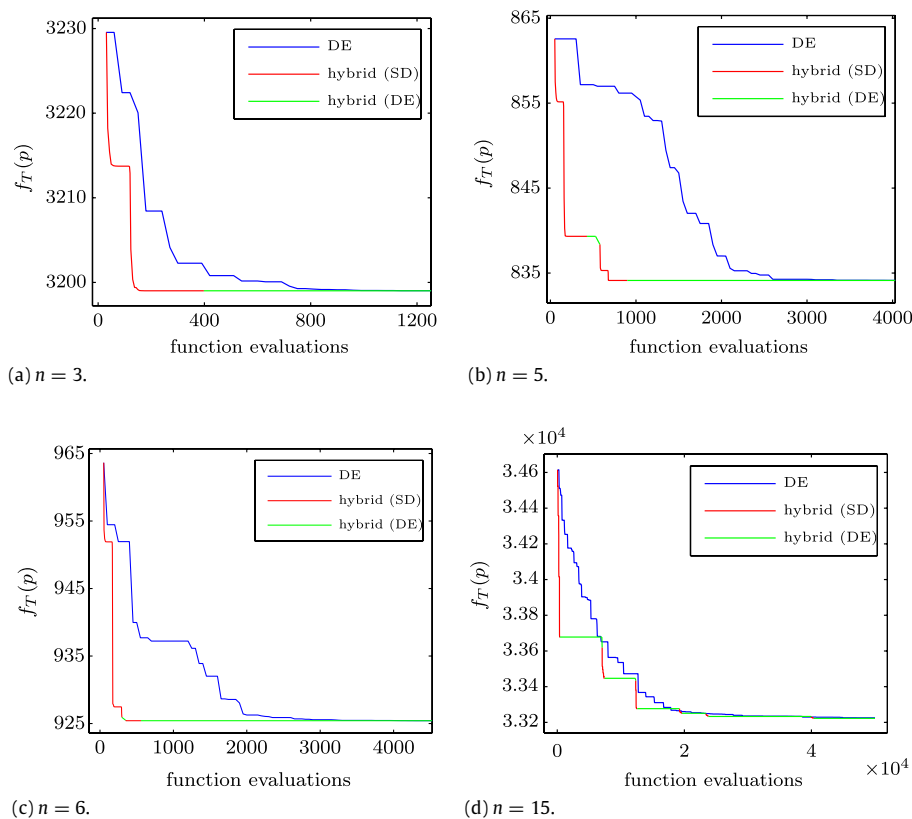


Fig. 13. Benefit of hybridizing DE with the subgradient descent (SD) algorithm. In this cyclical hybrid approach, DE is used to escape local minima (green curve), then the proposed SD algorithm is applied to refine the best individual (red curve). The outcome is a faster convergence than when the DE technique is applied alone (blue curve). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Acknowledgments

The authors would like to thank Arnaud Browet and Samuel Melchior for their valuable comments and suggestions that came out of many insightful conversations.

The first author was supported by a FRIA (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture) fellowship. The third author was supported by a FNRS (Fonds National de la Recherche Scientifique) fellowship.

References

- [1] N. Sinha, R. Chakrabarti, R. Chattopadhyay, Evolutionary programming techniques for economic load dispatch, *IEEE Transactions on Evolutionary Computation* 7 (2003) 83–94.
- [2] D. Walters, G. Sheble, Genetic algorithm solution of economic dispatch with valve point loading, *IEEE Transactions on Power Systems* 8 (1993) 1325–1332.
- [3] H.-T. Yang, P.-C. Yang, C.-L. Huang, Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions, *IEEE Transactions on Power Systems* 11 (1996) 112–118.
- [4] J.-B. Park, K.-S. Lee, J.-R. Shin, K. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, *IEEE Transactions on Power Systems* 20 (2005) 34–42.
- [5] N. Noman, H. Iba, Differential evolution for economic load dispatch problems, *Electric Power Systems Research* 78 (2008) 1322–1331.
- [6] T. Victoire, A. Jeyakumar, Hybrid PSO-SQP for economic dispatch with valve-point effect, *Electric Power Systems Research* 71 (2004) 51–59.
- [7] P. Attaviriyanupap, H. Kita, E. Tanaka, J. Hasegawa, A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function, *IEEE Power Engineering Review* 22 (2002) 77.
- [8] J. Cai, Q. Li, L. Li, H. Peng, Y. Yang, A hybrid CPSO-SQP method for economic dispatch considering the valve-point effects, *Energy Conversion and Management* 53 (2012) 175–181.
- [9] J. Alsumait, J. Sykulski, A. Al-Othman, A hybrid GA-PS-SQP method to solve power system valve-point economic dispatch problems, *Applied Energy* 87 (2010) 1773–1781.
- [10] J. Nocedal, S.J. Wright, *Numerical Optimization*, second ed., in: Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.
- [11] T. Niknam, A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem, *Applied Energy* 87 (2010) 327–339.
- [12] C. Yaşar, S. Özyön, A new hybrid approach for nonconvex economic dispatch problem with valve-point effect, *Energy* 36 (2011) 5838–5845.
- [13] X. Yuan, L. Wang, Y. Zhang, Y. Yuan, A hybrid differential evolution method for dynamic economic dispatch with valve-point effects, *Expert Systems with Applications* 36 (2009) 4042–4048.
- [14] U. Helmke, J.B. Moore, *Optimization and Dynamical Systems*, in: Communications and Control Engineering Series, Springer-Verlag London Ltd., London, 1994. With a foreword by R. Brockett.

- [15] A. Edelman, T.A. Arias, S.T. Smith, The geometry of algorithms with orthogonality constraints, *SIAM Journal on Matrix Analysis and Applications* 20 (1998) 303–353.
- [16] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [17] F.H. Clarke, Generalized gradients and applications, *Transactions of the American Mathematical Society* 205 (1975) 247–262.
- [18] G. Dirr, U. Helmke, C. Lageman, Nonsmooth Riemannian optimization with applications to sphere packing and grasping, in: F. Allgwer, P. Fleming, P. Kokotovic, A. Kurzhanski, H. Kwakernaak, A. Rantzer, J. Tsitsiklis, F. Bullo, K. Fujimoto (Eds.), *Lagrangian and Hamiltonian Methods for Nonlinear Control 2006*, in: *Lecture Notes in Control and Information Sciences*, vol. 366, Springer Berlin/Heidelberg, 2007, pp. 29–45.
- [19] T. Victorio, A. Jeyakumar, Deterministically guided PSO for dynamic dispatch considering valve-point effect, *Electric Power Systems Research* 73 (2005) 313–322.
- [20] R.T. Rockafellar, A property of piecewise smooth functions, *Computational Optimization and Applications* 25 (2003) 247–250.
- [21] R.L. Adler, J.-P. Dedieu, J.Y. Margulies, M. Martens, M. Shub, Newton's method on Riemannian manifolds and a geometric model for the human spine, *IMA Journal of Numerical Analysis* 22 (2002) 359–390.
- [22] A.M. Perelomov, A note on geodesics on ellipsoid, 2002, arXiv:math-ph/0203032.
- [23] C. Karney, Algorithms for geodesics, *Journal of Geodesy* 87 (2013) 43–55.
- [24] P.-A. Absil, J. Malick, Projection-like retractions on matrix manifolds, *SIAM Journal on Optimization* 22 (2012) 135–158.
- [25] F.H. Clarke, A new approach to Lagrange multipliers, *Mathematics of Operations Research* 2 (1976) 165–174.
- [26] J.V. Burke, A.S. Lewis, M.L. Overton, A robust gradient sampling algorithm for nonsmooth, nonconvex optimization, *SIAM Journal on Optimization* 15 (2005) 751–779.
- [27] A.A. Goldstein, Optimization of Lipschitz continuous functions, *Mathematical Programming* 13 (1977) 14–22.
- [28] Z. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, *IEEE Transactions on Power Systems* 18 (2003) 1187–1195.
- [29] T.N. Malik, A. Ul Asar, M.F. Wyne, S. Akhtar, A new hybrid approach for the solution of nonconvex economic dispatch problem with valve-point effects, *Electric Power Systems Research* 80 (2010) 1128–1136.
- [30] Z. Gaing, Closure to discussion of particle swarm optimization to solving the economic dispatch considering the generator constraints, *IEEE Transactions on Power Systems* 19 (2004) 2122–2123.
- [31] N. Sinha, K.M. Hazarika, S. Paul, H. Shekhar, A. Karmakar, Improved real quantum evolutionary algorithm for optimum economic load dispatch with non-convex loads, in: B. Panigrahi, S. Das, P. Suganthan, S. Dash (Eds.), *Swarm, Evolutionary, and Memetic Computing*, in: *Lecture Notes in Computer Science*, vol. 6466, Springer Berlin/Heidelberg, 2010, pp. 689–700.
- [32] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (1997) 341–359.
- [33] A. Bhattacharya, P. Chattopadhyay, Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch, *IEEE Transactions on Power Systems* 25 (2010) 1955–1964.
- [34] A.I. Selvakumar, Enhanced cross-entropy method for dynamic economic dispatch with valve-point effects, *International Journal of Electrical Power & Energy Systems* 33 (2011) 783–790.