Noname manuscript No.
(will be inserted by the editor)

Tight Bounds on Non-Convex Dispatches

Do you have a subtitle? If so, write it here

Loïc Van Hoorebeeck · Anthony Papavasiliou · P.-A. Absil

Received: date / Accepted: date

Abstract The economic dispatch problem is a major component of power systems and a lot of work has been made to provide fast and robust algorithms to solve the various economic dispatches. In order to capture physical effects such as the power losses of the network or the valve point loading effects in large combined cycle gas turbines, non-convex models are also considered. However, such additions to the model make the convergence analysis more tedious and few methods actually provide insights on the global optimality of the returned solution. In this work, we show how to efficiently provide lower and upper bounds to a non-convex economic dispatch problem. The method developed here is applied on intensively studied test cases and gives information on how close to the global optimal solution the available literature is.

Keywords Non-convex dispatch · Riemannian Descent · Non-smooth optimization

1 Introduction

With the recent massive integration of renewable energy sources in the energy mix, there is an increasing need for flexible units to counteract the inevitable uncertainties on the production. Hence, large gas units such as combined cycle gas turbines (CCGT) become also a major component of modern power systems, due to their ability to quickly responds to variations. The European Commission foresees a slight increase in gas-based electricity production for the 2030 european power mix and a stabilization around 20 % for 2050 [9]. For this reason, accurate representation of the model and the complex cost function of such units is of interest for the system operators, see, e.g., [14].

L. Van Hoorebeeck
Av. George Lemaitre 4-6, bte L4.05.01, 1348 Louvain-la-Neuve, Belgium.
Tel.: +32 10 47 80 32
E-mail: loic.vanhoorebeeck@uclouvain.be

The economic dispatch (ED) problem consists in the optimal scheduling of committed units to serve a given load at minimal cost. Two sets of constraints are considered. On one hand, the *operational constraints* ensure the feasibility of the dispatch and include limited power ranges, ramping rate, prohibited operation zones and so on. On the other hand, *balance constraints* force the dispatch to meet the load and guarantee that enough reserves are available. Usually, the economic dispatch problem does not take the network into account, the latter being tackled by the optimal power flow (OPF) problem. This omission rises two main problems on the practicality of the solution of an ED problem. First, an (optimal) solution of the ED will not be in general feasible for the network, *i.e.*, will not satisfy the *Kirchhoff's* laws. Second, the solution may no longer be optimal due to the power losses. To remedy the latter point, Kron introduced a quadratic model for the power losses which has been popularized by Kirchmayer [10]. Using this model, the losses, and indirectly the network, are taken into account.

In the economic dispatch problem, only the variable part of the cost function is considered because the units are already committed, and the fixed cost paid. Therefore for gas units, the cost is mostly linked to the fuel burnt. In usual thermal unit, this input-output function is modeled as a smooth quadratic function. However, such a function fails to efficiently model large CCGT units due to the valve-point effect (VPE) [7]. The valve-point effect refers to the increase of throlling losses when operating a turbine *off* a valve-point, that is just *after* the opening of the valve. Similarly, the unit operates the most efficiently when loaded *at* a valve-point, that is just *before* the next valve is open. A non-smooth and non-convex function, see (2), is commonly used to model this effect. This makes the optimization much challenging as the non-convexity implies the existence of several local minima and the non-smoothness prevents the use of conventional derivative-based techniques.

In order to solve this problem, the literature mostly follows two approaches: i) stochastics heuristics which aim at efficiently spanning the search space to quickly converge to a good solution and ii) slower deterministic methods based on an approximation of the objective. Examples of i) include imperialist algorithms [15,27], genetic algorithms [16], evolutionary algorithms [22,18,17, 3], simulated annealing algorithms [25,20]. Examples of ii) are [19,26] which use approximations of the objective without providing lower bound and our previous works [23,24] where the power losses are neglected.

Surprisingly, few studies try to derive guarantees such as convergence results or lower bounds to the global solution. As such methods rely on meta-parameters difficult to estimate *a priori*, it is delicate to estimate the quality of a solution obtained via these methods on a whole new test case. Besides, as the developed methods get more and more complex and the computational power increases, it is possible to reach low solution objectives, but it is worth asking *when* to stop and whether the current method already solves the problem to a satisfying extent. This motivates the present study which follows the work made in [23,24]. These studies focused on tackling the non-convex and

non-smooth objective. Here, we also include non-convexities in the feasible set due to the consideration of power losses.

The contributions of this work are the following. First, a full study of the geometric properties of the feasible set is made. Second, we show how to obtain an accurate approximation of the convex hull of this set. Third, we extend the local method of [1] to accommodate the multi-period dispatch problem. This local method is a Riemannian subgradient method which takes advantage of the inherent characterization of the feasible set as a Riemannian manifold. Lastly, we show how to combine our previous work in [24] with the extended local method to provide a competitive objective along with a lower bound to the global solution.

The organization of the paper is as follows. Section 2 introduces all objectives and feasible sets considered in the paper. We then show how to combine those ingredients to form several optimization problem which will be used to tackle the main problem, namely the economic dispatch problem. A study of the feasible set as well as the proposed relaxation is also made. The methods are detailed in section 3: the adaptive piecewise linearization method from [23] is shortly described as well as the extended Riemannian subgradient method. Ingredients of differential geometry needed for the later are given in this section. Section 4 gathers the results of the methods for several test cases and compare the objectives with state-of-the-arts methods. The lower bounds derived from our method allows to assess how close the solution of the proposed method and the other methods from the literature are to the optimal solution. Finally, conclusions are drawn in section 5.

2 Problem Formulation

In this section, all the objective functions and constraints considered in this work are introduced, then the main optimization problem is given as well as the other optimization problems which will be used through this paper to attempt to find lower bounds and feasible solutions, i.e., upper bound, to the main problem.

2.1 Notations

All vectors are written in bold, e.g. $\boldsymbol{x}$. An implicit partition is used for dealing with double indices: if $\boldsymbol{x}$ depends on both indices $g = 1, \dots |G|$ and

$t = 1, \dots |T|$, then $\boldsymbol{x}$ is partitioned as follows,

$$\boldsymbol{x} = \begin{pmatrix} x_{11} \\ \vdots \\ x_{n1} \\ x_{12} \\ \vdots \\ x_{n|T|} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{|T|} \end{pmatrix}.$$

The index g stands for the generator unit g listed in the set G and the index t stands for the timestep listed in the set T.

2.2 Objective Functions

2.2.1 Main objective

The objective f is defined as the sum of the production cost of every generator unit f_g at each time step,

$$f(\boldsymbol{p}) = \sum_{t \in T} \sum_{g \in G} f_g(p_{gt}) \,, \tag{1}$$

where the production of unit g at time step t is denoted as p_{gt}. A quadratic function is usually used for modeling the fuel cost of a given unit, but this fails to model the inherent non-convex characteristic of the problem when the VPE is taken into account. Hence, and similarly as what is commonly done in the literature, the cost function is modeled as the sum of a smooth quadratic part, f_g^{Q}, with a non-smooth rectified sine aimed at capturing the VPE, f_g^{VPE},

$$f_g(p_{gt}) = \underbrace{A_g p_{gt}^2 + B_g p_{gt} + C_g}_{:= f_g^{\mathrm{Q}}(p_{gt})} + \underbrace{D_g \left| \sin E_g (p_{gt} - P_g^-) \right|}_{:= f_g^{\mathrm{VPE}}(p_{gt})} , \tag{2}$$

where P_g^- is the minimum power production of generator g and A_g, B_g, C_g, D_g, E_g are parameters.

2.2.2 Surrogate objectives

A straightforward way of dealing with the non-linearities of the objective is to piecewise-linearize it. Two cases should be considered, either the objective is totally linearized, or only the non-quadratic part. The advantage of the former is its simplicity and the fact that the small over-approximation error on the quadratic part slightly compensates the large under-appromation error on the sine part, see [19], but the disadvantage is that the approximation is no-longer an under-approximation because the sine vanishes around the kink-points; this feature is discussed in [23].

MILP Objective The piecewise-linearization of a given function is entirely defined by the set of knots, which refers to the points where the pieces meet. Let $\boldsymbol{X}_{gt} := \left(X_{gt1}, \ldots, X_{gtn_{gt}^{\text{knot}}}\right)$, be the set of knots of unit g at time t, the approximation reads

$$h_{gt}^{\text{MILP}}(p_{gt}) := \begin{cases} \mathit{\Pi}[f_g, \boldsymbol{X}_{gt}](p_g) & \text{if } f_g^{\text{VPE}}(p_{gt}) \neq 0\,, \\ f_g^{\text{Q}}(p_g) & \text{else.} \end{cases} \tag{3}$$

In this expression, $\mathit{\Pi}[f, \boldsymbol{X}]$ stands for the piecewise-linear interpolation of a function f given the knots $\boldsymbol{X}$. These approximations of the fuel costs are then aggregated as in (1) to form the total surrogate objective h^{MILP},

$$h^{\text{MILP}}(\boldsymbol{p}) = \sum_{t \in T} \sum_{g \in G} h_{gt}^{\text{MILP}}(p_{gt})\,. \tag{4}$$

Note that even if the fuel cost f_g does not depend on the timestep, the approximation h_{gt}^{MILP} is time-dependent because so is the set of knots which define the approximation.

MIQP Objective The quadratic approximation is computed similarly as the previous case, except for the fact that only the non-quadratic part is interpolated:

$$h_{gt}^{\text{MIQP}}(p_{gt}) := f_g^{\text{Q}}(p_{gt}) + \mathit{\Pi}[f_g^{\text{VPE}}, \boldsymbol{X}_{gt}](p_g)\,. \tag{5}$$

The full objective is then obtained by summing over each timestep and unit, *e.g.*, for the MIQP objective,

$$h^{\text{MIQP}}(\boldsymbol{p}) = \sum_{t \in T} \sum_{g \in G} h_{gt}^{\text{MIQP}}(p_{gt})\,. \tag{6}$$

2.2.3 Objectives for the feasibility problem

Any objective may serve for finding a feasible solution and, in general, a constant objective is often a good choice. However, a feasible solution close to a given point $\boldsymbol{p}_0$ will be required in §2.4.3. We define the feasibility objective f^{feas} as

$$f^{\text{feas}}(\boldsymbol{p}\,; \boldsymbol{p}_0, \lambda_{\text{N}}, \lambda_{\text{Q}}) = \lambda_{\text{N}}\, ||\boldsymbol{p} - \boldsymbol{p}_0||_2^2 + \lambda_{\text{Q}} \sum_{t \in T} \sum_{g \in G} f_g^{\text{Q}}(\boldsymbol{p})\,, \tag{7}$$

for given parameters λ_{N}, $\lambda_{\text{Q}} \in \mathbb{R}_{\geqslant 0}$. Discussion over these parameters is made in Section 2.4.3.

2.3 Constraints

Constraints of an economic dispatch problem can be classified into two categories. First, the operational constraints ensure that the solution is physically feasible, this includes power range constraints and ramp limits. Secondly, the balance constraints impose that the demand is met and spinning reserves are available in case of contingencies. The constraints considered in this work are the following:

2.3.1 Power range limits

$$P_g^- \leqslant p_{gt} \leqslant P_g^+ \,, \tag{8}$$

where P_g^+ is the maximum power output of unit g.

2.3.2 Ramp rate restrictions

$$R_g^- \leqslant p_{gt} - p_{g(t-1)} \leqslant R_g^+ \,, \tag{9}$$

where R_g^- and R_g^+ are the ramp-down and ramp-up rates of unit g, respectively.

2.3.3 Power balance

$$\sum_{g \in G} p_{gt} = P_t^{\mathrm{D}} + p_t^{\mathrm{loss}} \quad \forall\, t = 1 \ldots |T| \,, \tag{10}$$

where P_t^{D} is the demand and p_t^{loss} the transmission losses in period t. The losses can be computed using the *Kron*'s formula [21]

$$p_t^{\mathrm{loss}} = \boldsymbol{p}_t{}^{\intercal} \boldsymbol{B} \boldsymbol{p}_t + \boldsymbol{B}_0 \boldsymbol{p}_t + B_{00} \,, \tag{11}$$

for a given matrix $\boldsymbol{B}$, vector $\boldsymbol{B}_0$ and parameter B_{00}. The matrix $\boldsymbol{B}$ which contains the *loss-coefficient* is symmetric, as it is obtained as the real part of a hermitian matrix, but not necessarily positive definite [21]. The latter case makes Lagrangian relaxation challenging even without VPE as the Lagrangian function would be non-convex. Discussion about these coefficients is made in Section 2.3.5.

2.3.4 Spinning reserve constraints

The spinning reserve is the ready-to-serve power obtained by increasing the output of generators that are already synchronized [6]. This differs from non-spinning reserves which include generators that can be brought online in a short-time (fast-start generators) but also the power available by increasing the import or decreasing the export to other systems. Here, the reserve requirements are modeled as [17]:

$$\left(\Delta_t^{(1)} = \sum_{g\in G} P_g^+ - (P_t^{\mathrm{D}} + p_t^{\mathrm{loss}} + P_t^{\mathrm{S}})\right) \geqslant 0 \tag{12}$$

$$\left(\Delta_t^{(2)} = \sum_{g\in G} \min\left(P_g^+ - p_{gt}, R_g^+\right) - P_t^{\mathrm{S}}\right) \geqslant 0 \tag{13}$$

$$\left(\Delta_t^{(3)} = \sum_{g\in G} \min\left(P_g^+ - p_{gt}, \frac{R_g^+}{6}\right) - \frac{P_t^{\mathrm{S}}}{6}\right) \geqslant 0 \tag{14}$$

Constraints (12) and (13) model the one hour spinning reserve and constraint (14) the 10 minutes spinning reserve requirement. Note that if the losses are neglected, $p_t^{\mathrm{loss}} = 0$, then constraint (12) does not depend on decision variables and is therefore simply a test on the feasibility of the problem.

2.3.5 Topology of the feasible set

Let us study the feasible set defined by (8) - (14). First, we show that the matrix associated with the B-coefficients is symmetric but not necessarily positive definite. Then, the definition of quadratic surface, or quadric is given, and (10) is expressed as a cartesian product of quadrics. Finally, a characterization of the quadric in this specific case is made.

Construction of the B-coefficients: Following [21], the losses are computed as

$$p_t^{\mathrm{loss}} = \boldsymbol{p}_t{}^{\intercal} \Re[H] \boldsymbol{p}_t$$

where

$$H = \Psi^{\intercal} C^{\intercal} R_{\mathrm{bus}} C^* \Psi^* \tag{15}$$

is hermitian and its real part is partitioned as

$$\Re[H] = \begin{pmatrix} \boldsymbol{B} & \frac{\boldsymbol{B}_0}{2} \\ \frac{\boldsymbol{B}_0{}^{\intercal}}{2} & B_{00} \end{pmatrix} . \tag{16}$$

The transformation matrices Ψ and C are invertible and full column rank, respectively. The matrix R_{bus} is the real part of the symmetric bus impedance matrix and is therefore also symmetric. In real networks, $\Re[H]$ and B can be expected to be invertible. (Not sure of it ?)

Coefficients of the matrix $\boldsymbol{B}$ can also be directly computed as

$$B_{ij} = \frac{R_{ij}}{|V_i|\,|V_j|} \frac{cos(\theta_i - \theta_j)}{\cos\phi_i \cos\phi_j} , \tag{17}$$

where R_{ij} is the real part of entry (i, j) of the impedance matrix, $\theta_i = \delta_i - \phi_i$ is the difference between the bus voltage angle δ_i and $\phi_i = \tan^{-1}\frac{Q_i}{P_i}$ which depends on the ratio between the reactive and real power at node i [11].

Characterization of the quadric hypersurfaces [2]

Definition 1 (Quadratic function) Let V be a vector space on the field $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. A relation

$$V \to \mathbb{K} : \boldsymbol{x} \mapsto \Psi(\boldsymbol{x}) = \rho(\boldsymbol{x}) + 2\phi(\boldsymbol{x}) + a$$

with a quadratic form ρ, a linear form ϕ and a constant $a \in \mathbb{R}$ is called a *quadratic function*.

Definition 2 (Quadric hypersurface) Let $\Psi : \mathbb{R}^n \to \mathbb{R}$ be a nonzero quadratic function, then its zero set

$$\mathcal{Q}(\Psi) = \{\boldsymbol{x} \,|\Psi(\boldsymbol{x}) = 0\} \subset \mathbb{R}$$

is a *quadric* of $\mathbb{R}^n$.

For some timestep t, equation (10) can be written as a quadric by choosing the relation

$$\mathbb{R}^n \to \mathbb{R} : \boldsymbol{p}_t \mapsto \Psi_t(\boldsymbol{p}_t) = \boldsymbol{p}_t{}^\intercal \boldsymbol{A} \boldsymbol{p}_t + 2\boldsymbol{b}^\intercal \boldsymbol{p}_t + c_t \tag{18}$$

with $\boldsymbol{A} = \boldsymbol{B}$, $\boldsymbol{b} = \frac{\boldsymbol{B}_0 - \mathbb{1}}{2}$ and $c_t = B_{00} - P_t^{\mathrm{D}}$. Since this constraint holds for every timestep, this yields the following set

$$\mathcal{Q}^{\mathrm{tot}} := \mathcal{Q}(\Psi_1) \times \mathcal{Q}(\Psi_2) \times \ldots \times \mathcal{Q}(\Psi_{|T|}) \tag{19}$$

Characterisation of the hypersurface Let r be the rank of $\boldsymbol{A} = \boldsymbol{B}$ and q the number of positive eigenvalue, by construction $\boldsymbol{A}$ is invertible and therefore $r = n$. Following the classification of [2], the quadric hypersurfaces is said to be of type 2 (Mittelpunktsquadrik or quadric with middle point). Indeed let us compute the rank of

$$\bar{A}_t = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^\intercal & c_t \end{pmatrix}, \tag{20}$$

since A is invertible, the *Guttman rank additivity formula* yields [28]

$$\operatorname{rank} \bar{A}_t = \operatorname{rank} A + \operatorname{rank}(c_t - \boldsymbol{b}^\intercal \boldsymbol{A}^{-1} \boldsymbol{b}) \,. \tag{21}$$

In general, $c_t \neq \boldsymbol{b}_t \boldsymbol{A}^{-1} \boldsymbol{b}$, it follows that $\operatorname{rank} \bar{A}_t \geqslant r = \operatorname{rank} \boldsymbol{A}$ and henceforth $\mathcal{Q}(\Psi_t)$ is a type-2 quadric. When all the eigenvalue of the quadratic form are positive, the non-degenerated type-2 quadric is an ellipsoid, illustrated in Fig. 3, otherwise it is an elliptic hyperboloid, see Fig 4.[1] A feature of the type-2 quadric is the existence of a center, $\boldsymbol{c}$, computed as

$$\boldsymbol{c} = -\boldsymbol{A}^{-1} \boldsymbol{b}. \tag{22}$$

[1] Here the implicit assumption on the problem feasibility is made, hence the (infeasible) case where all eigenvalues are negative is not considered.

2.4 Optimization problems

In this section, the different considered optimisation problems are described using the previously given constraints and objectives. First, the main problem, namely the economic dispatch with valve-point effect and transmission losses, is introduced. Then, the formulations of the surrogate problems, which provide lower bounds to the former problem are given. The feasibility problems used for finding a feasible solution or prove that non exists are also presented. Finally, the optimization problems used to obtain a relaxation of the feasible set are given.

2.4.1 Main problem: Economic Dispatch with VPE and transmission losses

Using the previously defined mathematical expressions, the main problem denoted as (P) reads

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & (1) \\ \text{s.t.} \quad & (8)-(14)\,. \end{aligned} \tag{P}$$

This is a non-smooth and non-convex continuous problem. The feasible set is the intersection of a polytope and a quadric hypersurface further described in Section 2.4.4.

2.4.2 Surrogate problem: Relaxation of the main problem

The surrogate problem, (S), aims at i) finding good infeasible solution guesses and ii) providing a lower bound to the solution of (P). This is achieved through an underapproximation of the objective similarly as [23] and a relaxation of the quadratic constraint 10. This relaxation, i.e., the approximation of the convex hull, is explained in Section 2.4.4. The surrogate problem, (S), reads

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & (6) \\ \text{s.t.} \quad & (8)-(9), (10)_R, (12)-(14)\,, \end{aligned} \tag{S}$$

where $(10)_R$ stands for the relaxation of constraint (10). Two relaxations are considered, a linear relaxation and a (convex) quadratic one, depending on whether $\boldsymbol{B}$ is positive definite.

2.4.3 Feasibility problem

The feasibility problem, F, focuses on finding feasible solution closed to the starting point $\boldsymbol{p}_0$.

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & (7) \\ \text{s.t.} \quad & (8)-(14)\,. \end{aligned} \tag{F}$$

This problems depends on the parameters λ_{N} and λ_{Q}. When $\lambda_{\mathrm{Q}} = 0$, (F) becomes a projection on the feasible set. If no initial guess $\boldsymbol{p}_0$ is available, λ_{N} is set to zero and the problem is a quadratically constrained quadratic program (QCQP). Finally, if both parameters are set to zero, (F), becomes a usual feasibility problem without any objective.

The fixed-time feasibility problem is also considered. It is similar as (F), except that the problem is decoupled with respect to a given timestep t. It reads

$$\begin{aligned} \min_{\boldsymbol{p}_t} \quad & \lambda_{\mathrm{Q}} f_g^{\mathrm{Q}}(\boldsymbol{p}_t) \\ \text{s.t.} \quad & (8)_t, (10)_t - (14)_t \,, \end{aligned} \tag{F$_t$}$$

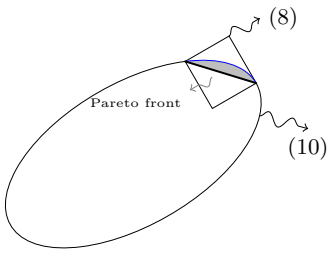
where $(\cdot)_t$ indicates that the constraint must only hold for the given timestep t.

2.4.4 Relaxation problem

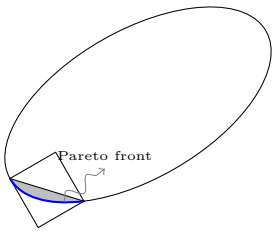
The relaxation problem intents to compute a convex relaxation of the constraint (10), written $(10)_R$. Two different cases are considered, either the B-coefficients matrix is positive definite or there is at least a negative eigenvalue, and the matrix is indefinite. This relaxation problem is decoupled with respect to the time t. Indeed, if $(10)_{t,R}$ is the relaxation of $(10)_t$, then taking the cartesian product of every time step yields a relaxation of (10).

Case I: $\boldsymbol{B}$ is positive definite In this case, the feasible set generated by constraint (10) is the surface of an ellipsoid. However, the power ranges of (8) restricts the feasible set in a box which is very small with respect to the ellipsoid. Hence, a linearization is often performed . In order to get a relaxation, the set induced by $(10)_R$ should include the set induced by (10). This condition is fulfilled if $(10)_R$ is the intersection of the ellipsoid interior with any chord which has no intersection with the feasible set of (P). Ideally, the chord should be chosen to minimize the relaxed set. However, the optimal chord is complicated to compute in practice. For example in the specific case where there are exacly n intersections between the ellipsoid and the box, the optimal chord is the one defined by the n intersections. Figure 1 illustrates such a plane and put in emphasis the optimal or non-optimal characteristic of the relaxation according to the position of the box, this type of behaviour has been studied in [13] for non-convex network constraints. Unfortunately, finding these n intersections is computationely intensive to find such an ideal plane: the simple enumeration of the hyper-cube vertices becomes intractable for a small number of generator $|G|$.

The procedure to obtain a relaxed plane at a given timestep t is the following: first a feasible point for timestep t, $\tilde{\boldsymbol{p}}_t^0$, is computed by solving (F$_t$), then the slope of the plane is obtained as the tangent plane of the ellipsoid



(a) Non-optimal relaxation: the Pareto front is the plane which is not feasible for the initial not relaxed problem.



(b) Optimal relaxation: the Pareto front is the ellipsoid surface which is feasible for the initial not relaxed problem.

Fig. 1: Optimal versus non-optimal relaxation.

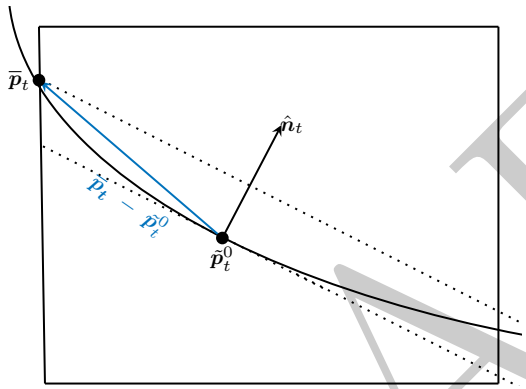


Fig. 2: Schetch of the geometry.

in $\tilde{\boldsymbol{p}}_t^0$ and finally, the plane is shifted toward the interior of the ellipsoid. The value of the shift is given with the following optimization problem

$$S_t^* := \max_{\bar{\boldsymbol{p}}_t \in \mathcal{X}_t} \hat{\boldsymbol{n}}_t \cdot (\bar{\boldsymbol{p}}_t - \tilde{\boldsymbol{p}}_t^0), \tag{Shift$_t$}$$

where $\mathcal{X}_t$ is the feasible set of (F_t). This procedure is illustrated in Figure 2 which is a magnification of Figure 1 around the available power ranges. The explicit procedure is given in Algorithm 1.

For a given time t, the relaxed balance constraint $(10)_{t,R}$ reads

$$\begin{aligned} \sum_{g \in G} p_{gt} &\geqslant P_t^{\mathrm{D}} + p_t^{\mathrm{loss}}, \\ 0 &\geqslant \boldsymbol{p}_t \cdot (\boldsymbol{A}\tilde{\boldsymbol{p}}_t^0 + \boldsymbol{b}_t) + S_t^*. \end{aligned} \tag{23}$$

Algorithm 1 Procedure to obtain relaxation at given timestep t

Require: $\boldsymbol{A}$ positive definite, $\tilde{\boldsymbol{p}}_t^0$ feasible for (F_t)
$\tilde{\boldsymbol{p}}_t^0 \leftarrow$ solution of (F_t)
$\boldsymbol{s}_t \leftarrow \boldsymbol{A}\tilde{\boldsymbol{p}}_t^0 + \boldsymbol{b}_t$
$S_t^* \leftarrow \max_{\bar{\boldsymbol{p}}_t \in \mathcal{X}_t} \hat{\boldsymbol{n}}_t \cdot (\bar{\boldsymbol{p}}_t - \tilde{\boldsymbol{p}}_t^0)$
$(10)_{t,R} \leftarrow \left(0 \geqslant \boldsymbol{p}_t \cdot \boldsymbol{s}_t + S_t^*\right) \bigcup \left(\sum_{g \in G} p_{gt} \geqslant P_t^{\mathrm{D}} + p_t^{\mathrm{loss}}\right)$
return $(10)_{t,R}$

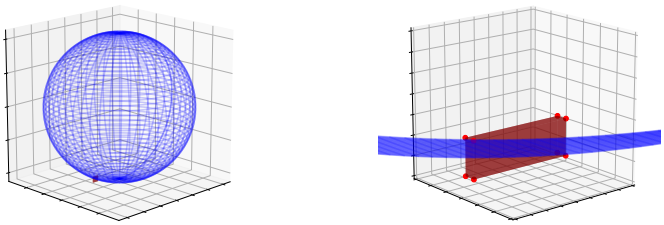


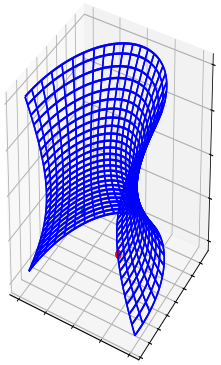
Fig. 3: Illustration of the relative size of the power ranges with respect to the quadric (ellipsoid) for a 3-unit problem at a given timestep t. The admissible power range, (8), is the interior of the red cube and the power balance, (10), the surface of the blue ellipsoid. The right figure is a magnification around the admissible power ranges. The red dots, in the right Figure, are the vertices of the box.

This linear relaxation is motivated by the fact that the relative size of the quadric is much larger than the admissible ranges of the unit. This phenomenon is illustrated in Figure 3 for a 3-unit problem at a given timestep t.

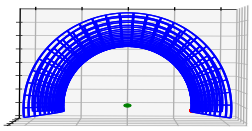
Case II: $\boldsymbol{B}$ is indefinite In this case, there are at least two eigenvalues of $\boldsymbol{A}$ of opposite sign, and therefore the closure of the quadric defined by $\boldsymbol{B}$ is no longer a convex set. Figure 4 illustrates an example of this case. Figures 4b and 4c show that a single plane will not be enough to construct the relaxation: Figure 4b prompts the use of an interior relaxation plane, in a similar way as the case I, but Figure 4c demonstrates that en exterior plane should also be used. To tackle this issue, we solve (Shift_t) for both directions $\hat{\boldsymbol{n}}_t$ and $-\hat{\boldsymbol{n}}_t$. The whole procedure is explicitly given in Algorithm 2 and depicted in Figure 5.

2.4.5 Comparison of the optimization problem

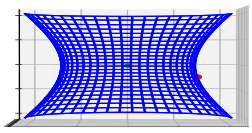
The characterization of each optimization problem is specified in Table 1. The two last problems are decoupled with respect to the time which reduces



(a) Full hyperboloid.



(b) OXZ view.



(c) OXY view.

Fig. 4: Illustration of the relative size of the power ranges with respect to the quadric for a 3-unit problem at a given timestep t. In this example the Kron's matrix is *not* positive definite: two eigenvalues are positive and the last one is negative. The quadric is a onesheet hyperboloid. The admissible power range, (8), is the interior of the red cube, too small to be distinguishable, and the power balance, (10), the surface of the blue hyperboloid. Figures 4b and 4c show different views. The green point is the center of the quadric.

Algorithm 2 Procedure to obtain relaxation at given timestep t for a non-convex quadric.

$\tilde{\boldsymbol{p}}_t^0 \leftarrow$ solution of (F_t)
$\boldsymbol{s}_t \leftarrow \boldsymbol{A}\tilde{\boldsymbol{p}}_t^0 + \boldsymbol{b}_t$
$S_t^{*,\mathrm{int}} \leftarrow \max_{\bar{\boldsymbol{x}}_t \in \mathcal{X}_t} \hat{\boldsymbol{n}}_t \cdot (\bar{\boldsymbol{x}}_t - \tilde{\boldsymbol{p}}_t^0)$
$S_t^{*,\mathrm{ext}} \leftarrow \max_{\bar{\boldsymbol{y}}_t \in \mathcal{X}_t} -\hat{\boldsymbol{n}}_t \cdot (\bar{\boldsymbol{y}}_t - \tilde{\boldsymbol{p}}_t^0)$
$(10)_{t,R} \leftarrow \left(0 \geqslant \boldsymbol{p}_t \cdot \boldsymbol{s}_t + S_t^{*,\mathrm{int}}\right) \bigcup \left(0 \leqslant \boldsymbol{p}_t \cdot \boldsymbol{s}_t + S_t^{*,\mathrm{ext}}\right)$
return $(10)_{t,R}$

significantly the size of the problem. They are considered as *easy* problem in comparison with the three first ones and, in the test cases studied here, can be solved to optimality in less than a second. Among the three larger problem, (P) is unquestionably the more difficult which makes sense as it is the main problem of interest. (F) is arguably easier than (S): the reason is that any solution of (F) is acceptable as the goal is to find a feasible solution. Of course, the closer the solution is to the input, the better but this is not

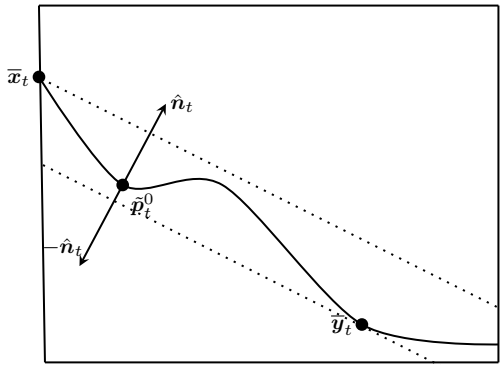


Fig. 5: Example of the 2-plane relaxation for a surface whose interior is non-convex.

an issue if the finale optimality gap is large. On the other hand, (S) is a true optimization problem in the sense that we are interested in the lowest objective and especially a high lower bound.

Table 1: Comparison of the optimization problems

	(P)	(S)	(F)	(F_t)	($Shift_t$)
Classification	NLP	MIQP	QCQP	QCQP	QCLP
Convexity	Non-convex	Non-convex	Non-convex	Non-convex	Non-convex
Objective	Non-convex, non-smooth	Piecewise-linear	Quadratic	Quadratic	Linear
Feasible set	Non-convex	Convex[2]	Non-convex	Non-convex	Non-convex
Problem size	$\|T\|\|G\|$	$\|T\|\|G\|$	$\|T\|\|G\|$	$\|G\|$	$\|G\|$

3 Methods

In this section, we explain how to combine all the elements developed in Section 2 to solve (P). First, we show how to get a lower bound of the problem and then, we show how to obtain an upper bound, i.e., a feasible solution, and improve it using a Riemannian gradient descent. Finally, both steps are combined in a single algorithm.

3.1 Getting a lower bound

In a similar fashion as [23,24], the lower bound is obtained through an underapproximation of the objective via piecewise-linearization. However, this is

[2] The initial feasible set is non-convex but the modelization of the piecewise-linear objective is made through integer variables which makes the feasible set inherently non-convex.

not sufficient here as the feasible set is the subset of a quadric. Hence, this non-convex set is relaxed using the solution of (Shift_t) which requires for each timestep t a point $\tilde{\boldsymbol{p}}_t^0$ feasible for (F_t).

The goal here is to obtain a lower bound but also a candidate which is globally efficient, meaning that its objective is close to the global optimum. In general, this candidate will not be feasible due to the relaxation of the feasible set, but we expect it to be sufficiently close to the feasible set such that when we project it back to the feasible set, it does not lose its optimality feature.

Feasible set relaxation $\tilde{\boldsymbol{p}}_t^0$ is readily obtained for each t with Algorithm 3. Notice that the point is not globally feasible and hence $f(\tilde{\boldsymbol{p}}^0)$ is *not* an upper bound to the global solution. This point is simply a starting point for Algorithms 1 and 2 depending on whether $\boldsymbol{B}$ is positive definite.

Algorithm 3 Find $\tilde{\boldsymbol{p}}_T^0$ feasible for each timestep t

for $t \in T$ **do**
  $\tilde{\boldsymbol{p}}_t^0 \leftarrow \arg\min\ (\text{F}_t)$
end for
return $\tilde{\boldsymbol{p}}_t^0$

Solution of the surrogate problem The lower bound can be obtained via the adaptive piecewise-linearization algorithm (APLA) described in [23] but this method suffers from long execution time. In practice, we are not interested in spending too much time in getting the lower bound, therefore in the numerical experiments section 4, we rather use the heuristic based on APLA given in [24]. To simplify the discussion, the description is made here with APLA.

The method APLA can be summarized as follows. First a set of knots which define the piecewise-linear approximation is defined. These knots are chosen such as the surrogate function h^{MIQP} in (6) is an under-approximation. Then, the (first) surrogate problem $(S)^1$ defined with the (first) set of knots $\boldsymbol{X}^1$ is solved with a MIP solver. If we neglect the fact that the feasible set is relaxed, the solution returned by the solver is non-optimal because i) the optimality is guaranteed up to a given tolerance and ii) the surrogate objective under-approximates the real objective. To remedy the latter point, the approximation is refined around the returned solution; this adaptive refinement exhibits lower number of integer variables than a global refinement consisting in doubling the number of linear pieces. It is proven in [23] that this method converges up to the solver tolerance, i.e., the second cause of non-optimality goes to zero as the number of APLA iterations goes to infinity. The method is outlined in the dotted frame of Figure 6 .

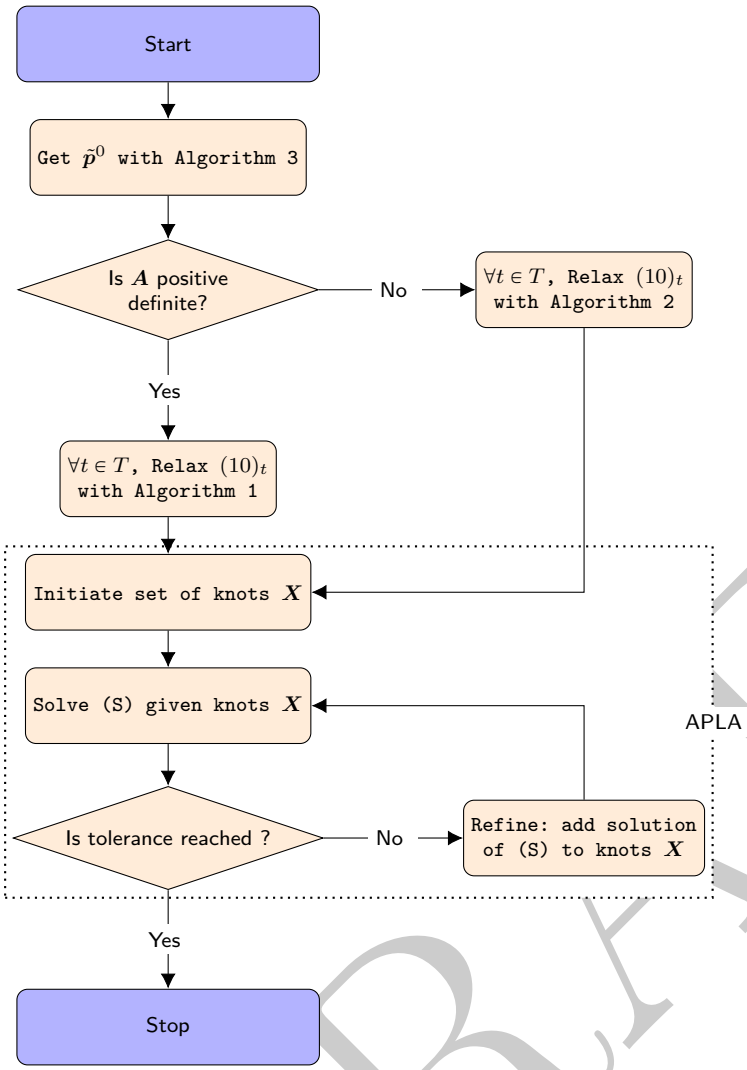


Fig. 6: Flowchart of the method to obtain a lower bound along with a first (infeasible) candidate with low objective.

3.2 Getting an upper bound

A simple and direct method to obtain a feasible point, i.e., a first upper bound to the global solution, is to project the candidate obtained at the end of the procedure depicted in Figure 6. However, the projection may destroy the global optimality and it is worthy to try to improve the obtained feasible solution through a local search. In [19], the authors use a barrier method as a local solver. However, this method mildly improves the solution and it relies on

barrier parameters difficult to estimate *a priori*. In this section, we propose to adapt the Riemannian gradient descent given in [4] to accounts for reserves and multiple timesteps.

Riemannian subgradient scheme The Riemannian subgradient descent can be seen as a classical line-search scheme,

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \alpha^k \boldsymbol{d}^k \tag{24}$$

where α^k is the stepsize and $\boldsymbol{d}^k$ the (descent) direction at iteration k. Usually, the question remains on how to choose the stepsize and the descent direction to fully determine the scheme. However here, an extra work is needed to redefine the "+" operation; as $\boldsymbol{p}$ is not defined on a vector field, it is not true in general that (24) yields a feasible point. A simple idea would be to project the resulting point on the feasible set, this defines a projected line-search scheme. This is not a good idea for this problem for two reasons. Firstly the projection onto the feasible set of (P) is a costly operation (see classification of (F) in Table 1) and a usual line-search scheme requires at least a few dozen iterations. Secondly, the geometry shows a rich structure of a manifold which could be taken as an advantage.

In the next paragraphs, the general Riemannian geometry is introduced, then the retraction – the extension of the "+" operator – and the descent direction are described. Finally, the stepsize rule is given.

Riemannian geometry In a similar way as [4], we define the quadric manifold and extended quadric manifold.

Proposition 1 (Quadric manifold) *Let $\Psi : \mathbb{R}^n \to \mathbb{R}$ be a nonzero quadratic function, the quadric hypersurface $\mathcal{Q}(\Psi)$ is a $n-1$ dimensional smooth manifold of $\mathbb{R}^n$.*

Definition 3 (Extended quadric manifold) The cartesian product of the quadrics defined for each timestep t as defined in (18) is called *extended quadric manifold* and computed as

$$\mathcal{Q}^{\text{tot}} := \mathcal{Q}(\Psi_1) \times \mathcal{Q}(\Psi_2) \times \ldots \times \mathcal{Q}(\Psi_{|T|}).$$

Proposition 1 follows from the $\mathcal{C}^\infty$ smoothness of the function Ψ, see [12] for more details. The extended quadric manifold is effectively a manifold because the cartesian product of smooth manifolds is also a manifold. In this case, the dimension of the manifold is $|T|\,(|G|-1)$.

A first important object to be described when dealing with manifolds is the tangent space, intuitively, it refers to the first order approximation of the manifold at a given point $\boldsymbol{p}$. The motivation of using the tangent space comes from the definition of the manifold as a topological space that is locally homeomorphic to the Euclidean space of the same dimension. This mathematical object is used in a lot of algorithms on manifolds in the following way: the tangent plane is defined at a given point $\boldsymbol{p}$, then any other point $\boldsymbol{p}'$ of the

manifold is mapped to the tangent space through the *logarithmic map*, in this tangent space the usual vectorial operations can be used and the resulting vector is mapped back to the manifold via the *exponential map*.

The tangent space of the quadric manifold $\mathcal{Q}(\Psi_t)$ at a given point $\boldsymbol{p}_t$ is defined in general as

Definition 4 Let $\mathcal{Q}(\Psi_t)$ be a smooth real manifold, the tangent space reads

$$\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) = \left\{\boldsymbol{\xi} \in \mathbb{R}^{|G|} \mid \exists c : \mathbb{R} \mapsto \mathcal{Q}(\Psi_t) \text{ with } c(0) = 0,\, c'(0) = \boldsymbol{\xi}\right\}.$$

Using the specific structure of the quadric manifolds, this tangent space is computed as [4]

$$\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) = \left\{\boldsymbol{\xi} \in \mathbb{R}^{|G|} \mid \boldsymbol{\xi}^\intercal(2\boldsymbol{A}\boldsymbol{p}_t + \boldsymbol{b}_t) = 0\right\}, \tag{25}$$

and $\dim\left(\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)\right) = |G| - 1$.

Definition 5 (Tangent bundle) The tangent bundle $\mathrm{T}\mathcal{Q}(\Psi_t)$ of a manifold $\mathcal{Q}(\Psi_t)$ is defined as the union of every tangent space at every point of the manifold,

$$\mathrm{T}\mathcal{Q}(\Psi_t) = \bigcup_{\boldsymbol{p}_t \in \mathcal{Q}(\Psi_t)} \mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t).$$

Since every tangent space is a vector space, they can be endowed with an inner product $\langle\cdot,\cdot\rangle_{\boldsymbol{p}_t}$ defined as the restriction of the canonical euclidean product on the tangent space $\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)$,

$$\langle\cdot,\cdot\rangle_{\boldsymbol{p}_t} : \mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) \times \mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) \to \mathbb{R} : (\boldsymbol{\xi}, \boldsymbol{\zeta}) \mapsto \langle\boldsymbol{\xi}, \boldsymbol{\zeta}\rangle_{\boldsymbol{p}_t} = \boldsymbol{\xi}^\intercal\boldsymbol{\zeta}. \tag{26}$$

Similarly, we define an inner product $\langle\cdot,\cdot\rangle_{\boldsymbol{p}}$ as the restriction of the canonical inner product on the tangent space,

$$\langle\cdot,\cdot\rangle_{\boldsymbol{p}} : \mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}} \times \mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}} \to \mathbb{R} : (\boldsymbol{\xi}, \boldsymbol{\zeta}) \mapsto \langle\boldsymbol{\xi}, \boldsymbol{\zeta}\rangle_{\boldsymbol{p}} = \boldsymbol{\xi}^\intercal\boldsymbol{\zeta} = \sum_{t\in T}\langle\boldsymbol{\xi}_t, \boldsymbol{\zeta}_t\rangle_{\boldsymbol{p}_t}. \tag{27}$$

This inner product induces the canonical norm: $||\boldsymbol{\xi}||_{\boldsymbol{p}} = \langle\boldsymbol{\xi}, \boldsymbol{\zeta}\rangle_{\boldsymbol{p}}^{1/2}$. The smooth manifold equipped with an inner product on the tangent space at every point is called a *Riemannian* manifold.

Equipped with these inner products, the *normal space* can be computed as the orthogonal complement of the tangent space.

Definition 6 Let $\mathcal{Q}(\Psi_t)$ be a $|G| - 1$ smooth manifold embedded in $\mathbb{R}^n$ and $\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)$ its tangent space, the normal space is defined as

$$\mathrm{N}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) := \mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)^\perp \tag{28}$$

It follows from (25) that

$$\mathrm{N}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t) = \{\tau(2\boldsymbol{A}\boldsymbol{p}_t + \boldsymbol{b}_t) \mid \tau \in \mathbb{R}\}, \tag{29}$$

and $\dim\left(\mathrm{N}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)\right) = 1$.

Now that an expression for the tangent and normal space of each individual manifold have been obtained, both can be computed for the extended quadric manifold.

Proposition 2 $\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{tot} = \mathrm{T}_{\boldsymbol{p}_1}\mathcal{Q}(\Psi_1) \times \mathrm{T}_{\boldsymbol{p}_2}\mathcal{Q}(\Psi_2) \times \ldots \times \mathrm{T}_{\boldsymbol{p}_T}\mathcal{Q}(\Psi_T)$

Proof See Guillemin & Pollack 1.2

Proposition 3 $\mathrm{N}_{\boldsymbol{p}}\mathcal{Q}^{tot} = \mathrm{N}_{\boldsymbol{p}_1}\mathcal{Q}(\Psi_1) \times \mathrm{N}_{\boldsymbol{p}_2}\mathcal{Q}(\Psi_2) \times \ldots \mathrm{N}_{\boldsymbol{p}_{|T|}}\mathcal{Q}(\Psi_{|T|})$

Proof We first show that $\mathrm{N}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}} = (\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}})^{\perp} \supseteq \mathrm{N}_{\boldsymbol{p}_1}\mathcal{Q}(\Psi_1) \times \mathrm{N}_{\boldsymbol{p}_2}\mathcal{Q}(\Psi_2) \times \ldots \times \mathrm{N}_{\boldsymbol{p}_{|T|}}\mathcal{Q}(\Psi_{|T|})$ and then we conclude with an argument on the dimensions.

i) Let $\boldsymbol{p} \in \mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}}$ and $\boldsymbol{p}' \in \mathrm{N}_{\boldsymbol{p}_1}\mathcal{Q}(\Psi_1) \times \mathrm{N}_{\boldsymbol{p}_2}\mathcal{Q}(\Psi_2) \times \ldots \times \mathrm{N}_{\boldsymbol{p}_{|T|}}\mathcal{Q}(\Psi_{|T|})$, both are partitioned as follows: $\boldsymbol{p} = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_{|T|} \end{pmatrix}$ and $\boldsymbol{p}' = \begin{pmatrix} \boldsymbol{p}'_1 \\ \boldsymbol{p}'_2 \\ \vdots \\ \boldsymbol{p}'_{|T|} \end{pmatrix}$. It follows from Proposition 2 and Definition 6 that $\boldsymbol{p}^{\intercal}\boldsymbol{p}' = 0$ and therefore that $\boldsymbol{p}' \in (\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}})^{\perp}$.

ii) Since $\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}}$ is a linear subspace of $\mathbb{R}^{|G||T|}$, we have $|G|\,|T| - \dim\left(\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}}\right) = \dim\left((\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}})^{\perp}\right) = |G|\,|T| - |T|\,(|G| - 1) = |T|$. This concludes the proof as

$$\dim\left(\mathrm{N}_{\boldsymbol{p}_1}\mathcal{Q}(\Psi_1) \times \mathrm{N}_{\boldsymbol{p}_2}\mathcal{Q}(\Psi_2) \times \ldots \times \mathrm{N}_{\boldsymbol{p}_{|T|}}\mathcal{Q}(\Psi_{|T|})\right) = |T| \ .$$

Since we have shown that both tangent and normal spaces of the extended quadric are the cartesian products of the tangent and normal space of the individual manifold $\mathcal{Q}(\Psi_t)$, we can easily extend the projection operator from [4] by working composantwisely.

The projection $P_{\boldsymbol{p}}(v)$ of a vector $\boldsymbol{v} \in \mathbb{R}^{|G||T|}$ partitioned as $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{|T|})^{\intercal}$ onto $\mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}}$ can be constructed by removing the normal component of $\boldsymbol{v}$:

$$P_{\boldsymbol{p}}(v) = ({\boldsymbol{v}_1}^{\perp}, \ldots, \boldsymbol{v}_{|T|}^{\perp}), \tag{30}$$

where $\boldsymbol{v}_t^{\perp} = \boldsymbol{v}_t - \tau_t(2\boldsymbol{A}\boldsymbol{p}_t + \boldsymbol{b}_t)$ and τ_t is chosen to ensure that $P_{\boldsymbol{p}_t}(\boldsymbol{v}_t)$ belong to $\mathrm{T}_{\boldsymbol{p}_t}\mathcal{Q}(\Psi_t)$, i.e.,

$$\tau_t = \frac{{\boldsymbol{v}_t}^{\intercal}(2\boldsymbol{A}\boldsymbol{p}_t + \boldsymbol{b}_t)}{||(2\boldsymbol{A}\boldsymbol{p}_t + \boldsymbol{b}_t)||^2}. \tag{31}$$

Retraction

Definition 7 (Retraction) A retraction R is a smooth mapping from the tangent bundle of a manifold to the manifold itself,

$$\mathrm{R}_t : \mathrm{T}\mathcal{Q}(\Psi_t) \to \mathcal{Q}(\Psi_t) : (\boldsymbol{p}_t, \boldsymbol{\xi}_t) \mapsto \boldsymbol{q}_t := \mathrm{R}_t(\boldsymbol{p}_t, \boldsymbol{\xi}_t).$$

It is clear that the retraction is not unique, and in fact, the retraction can be seen as an approximation of the exponential map. Indeed, in this specific case the exponential map cannot be easily computed, see the discusion in [4], but some retractions can be easily computed.

The retraction considered in this work and introduced in [4] is illustrated in Figure 7: the retraction $\mathrm{R}_t(\boldsymbol{p}_t, \boldsymbol{\xi}_t)$ is obtained by looking at the intersection $\boldsymbol{q}_t$ between the quadric and the line between $\boldsymbol{\xi}_t$ and the quadric center $\boldsymbol{c}_t$. Note that this intersection is not supposed to be unique (see $\boldsymbol{q}'_t$); to remedy this, the closest point to $\boldsymbol{\xi}_t$ is chosen. In practice, the starting point, $\boldsymbol{v}_t$, may not lie in the tangent space of $\boldsymbol{p}_t$. An extra step of projection is then needed, $\boldsymbol{\xi}_t = P_{\boldsymbol{p}}(\boldsymbol{v}_t)$.

This retraction can be readily extended to the multistep case: it suffices to work with each composant independently:

$$\mathrm{R} : \mathrm{T}\mathcal{Q}^{\mathrm{tot}} \to \mathcal{Q}^{\mathrm{tot}} : (\boldsymbol{p}, \boldsymbol{\xi}) \mapsto \boldsymbol{q} := \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_{|T|} \end{pmatrix}, \tag{32}$$

with $\boldsymbol{q}_t = \mathrm{R}_t(\boldsymbol{p}_t, \boldsymbol{\xi}_t)$. Notice that, if the retraction is illustrated with an ellipse in Fig.7, it is not reduced to this specific quadric. Any type-2 quadric or quadric with a middle point (see § 2.3.5), can be considered.

Going back to equation (24), if the direction $\boldsymbol{d}^k$ does not belong the to the tangent space of the current iterate $\boldsymbol{p}^k$, it reads

$$\boldsymbol{p}^{k+1} = \mathrm{R}(\boldsymbol{p}^k, P_{\boldsymbol{p}}\left(\alpha^k \boldsymbol{d}^k\right)). \tag{33}$$

Descent direction on a manifold Before the discussion on the *descent* direction, we define the concept of admissible direction which accommodates the constrained feasible set defined by (10).

Definition 8 An admissible direction defined at point $\boldsymbol{p} \in \mathcal{Q}^{\mathrm{tot}}$ is a vector $\boldsymbol{d} \in \mathrm{T}_{\boldsymbol{p}}\mathcal{Q}^{\mathrm{tot}}$ for which there exists $\epsilon > 0$ such that $\mathrm{R}(\boldsymbol{p}, \alpha\boldsymbol{d})$ belongs to $\mathcal{Q}^{\mathrm{tot}}$ for all $\alpha \in [0, \epsilon]$.

The gradient is inherent in the concept of steepest descent, but the function (2) is only smooth almost-everywhere, the zero-measure set where it is

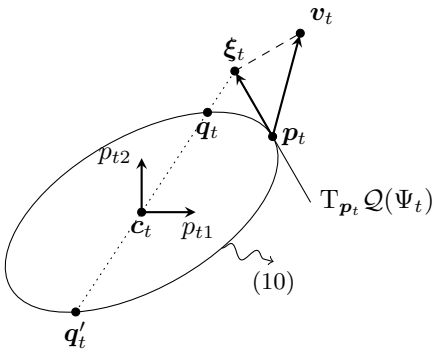


Fig. 7: Illustration of the retraction, $\mathrm{R}_t(\boldsymbol{p}_t, \boldsymbol{\xi}_t)$.

non-smooth is located at positions where the argument of the absolute value changes of sign. This finite set is computed as

$$S = \underbrace{\bigcup_{g\in G} S_g \times \ldots \times \bigcup_{g\in G} S_g}_{|T| \text{ times}}, \tag{34}$$

where

$$S_g := \left\{ P_g^- + \frac{(j-1)\pi}{2E_g} \mid j = 1, \ldots, 1 + \left\lceil (P_g^+ - P_g^-)\frac{2E_g}{\pi} \right\rceil \right\}. \tag{35}$$

For a convex function, the gradient is often remplaced by the subgradient, however this mathematical object cannot be used for the nonconvex functions (2). Here, we consider the closely connected concept of *generalized gradient* introduced in [5]. First, let us define the generalized directional derivative $f^\circ(\boldsymbol{p}; \boldsymbol{v})$ of the locally Lipschitz function $f : X \to R$, for a Banach space X, in the direction $\boldsymbol{v}$ as

$$f^\circ(\boldsymbol{p}; \boldsymbol{v}) = \limsup_{\substack{\boldsymbol{h}\to 0 \\ \lambda \downarrow 0}} \frac{f(\boldsymbol{p} + \boldsymbol{h} + \lambda \boldsymbol{v}) - f(\boldsymbol{p} + \boldsymbol{h})}{\lambda}.$$

This function is convex, independently on the convexity of f. The *generalized gradient* of f at $\boldsymbol{p}$, written $\partial f(\boldsymbol{p})$, is defined as the subdifferential of the convex function $f^\circ(\boldsymbol{p}, \cdot)$ at $\mathbf{0}$. In particular we have,

$$\partial f(\boldsymbol{p}) = \left\{ \boldsymbol{\zeta} \in X^* \mid f^\circ(\boldsymbol{p}; \boldsymbol{v}) \geqslant \langle \boldsymbol{v}, \boldsymbol{\zeta} \rangle \quad \forall \boldsymbol{v} \in X \right\}, \tag{36}$$

with X^* the dual space of X. The generalized gradient shares some important properties with the subdifferential of a convex function, namely the fact that it is a nonempty convex and compact set and that if a point $\boldsymbol{p}$ is a local

minimizer of f, then $\mathbf{0} \in \partial f(\boldsymbol{p})$. Furthermore, if f is convex, then the generalized gradient coincides with the subdifferential, and for a point $\boldsymbol{p}$ locally differentiable, we have $\partial f(\boldsymbol{p}) = \{\nabla f(\boldsymbol{p})\}$.

The function (1), is a (locally) Lischitz function that can be computed as the pointwise maximum of $m := 2^{|G||T|}$ smooth functions[3], i.e.,

$$f(\boldsymbol{p}) = \max_{j=1,\dots,m} f_j(\boldsymbol{p}). \tag{37}$$

Hence, in this specific case, the generalized gradient can be described as

$$\partial f(\boldsymbol{p}) = \operatorname{co}\left\{\nabla f_j(\boldsymbol{p}) \mid j \in \mathcal{I}_f(\boldsymbol{p})\right\}, \tag{38}$$

where $\operatorname{co}\{\cdot\}$ denotes the convex hull and $\mathcal{I}_f$ the set of indices for which the maximum in (37) is attained.

This framework is valid for the unconstrained problem (P). Let us now integrate the manifold-like constraint (10) and then the other linear constraints.

Given a smooth function f_j from the pointwise maximum in (37), the *projected gradient* is defined as follows

$$\operatorname{grad} f_j(\boldsymbol{p}) = P_{\boldsymbol{p}}\left(\nabla f_j(\boldsymbol{p})\right), \tag{39}$$

and the projected generalized gradient is given by

$$\operatorname{grad} f(\boldsymbol{p}) = \operatorname{co}\left\{\operatorname{grad} f_j(\boldsymbol{p}) \mid j \in \mathcal{I}_f(\boldsymbol{p})\right\}. \tag{40}$$

The steepest admissible direction $\boldsymbol{d}^k$ from iterate $\boldsymbol{p}^k$ is obtained by computing the shortest vector in $\operatorname{grad} f(\boldsymbol{p}^k)$, see [4] for more details. This can be computed by minimizing the norm of the convex combination of the projected gradients. If the coefficients of the convex coefficients are given by

$$\begin{aligned}\boldsymbol{\lambda}^k &= \arg \min_{\substack{\boldsymbol{\lambda}\geqslant 0 \\ \sum\lambda_j=1}} \left\| \sum_{j\in\mathcal{I}_f(\boldsymbol{p}^k)} \lambda_j \operatorname{grad} f_j(\boldsymbol{p}^k) \right\|^2 \\ &= \arg \min_{\substack{\boldsymbol{\lambda}\geqslant 0 \\ \sum\lambda_j=1}} \left\| P_{\boldsymbol{p}^k}\left(\sum_{j\in\mathcal{I}_f(\boldsymbol{p}^k)} \lambda_j \nabla f_j(\boldsymbol{p}^k) \right) \right\|^2 \end{aligned} \tag{41}$$

then the steepest-descent admissible direction is computed as

$$\boldsymbol{d}^k = -P_{\boldsymbol{p}^k}\left(\sum_{j\in\mathcal{I}_f(\boldsymbol{p}^k)} \lambda_j^k \nabla f_j(\boldsymbol{p}^k) \right). \tag{42}$$

This optimization problem is defined on a high dimensional ($2^{|G||T|}$) simplex and should be resolved at each iteration. To remedy the expected high solving time, [4] also introduces a reformulation which takes advantages of the

[3] Since $|x| = \max\{x, -x\}$ and there are $|G|\,|T|$ absolute values in (1).

specific form of the function (1) and considerently reduces the dimension of the problem.

Let $\mathcal{S}(\boldsymbol{p}^k)$ be the set of indices of $\boldsymbol{p}^k$ that cancel the sine term and $\mathcal{F}(\boldsymbol{p}^k)$ the remaining indices, i.e.,

$$\begin{aligned}\mathcal{S}(\boldsymbol{p}^k) &= \left\{ \underbrace{(g_1^s, t_1^s)}_{:=s_1}, \underbrace{(g_2^s, t_2^s)}_{:=s_2}, \dots, \underbrace{(g_{n_s^k}^s, t_{n_s^k}^s)}_{:=s_{n_s^k}} \right\} \\ &= \bigcup_{t\in T} \mathcal{S}_t(\boldsymbol{p}_t^k) = \bigcup_{t\in T} \left\{(g,t) \mid g \in G, f_g^{\mathrm{VPE}}(p_{gt}) = 0\right\} \end{aligned} \tag{43}$$

$$\begin{aligned}\mathcal{F}(\boldsymbol{p}^k) &= \left\{ \underbrace{(g_1^f, t_1^f)}_{:=f_1}, \underbrace{(g_2^f, t_2^f)}_{:=f_2}, \dots, \underbrace{(g_{n_f^k}^f, t_{n_f^k}^f)}_{:=f_{n_f^k}} \right\} \\ &= \bigcup_{t\in T} \mathcal{F}_t(\boldsymbol{p}_t^k) = \bigcup_{t\in T} \left\{(g,t) \mid g \in G, (g,t) \notin \mathcal{S}(\boldsymbol{p}_t^k)\right\} \end{aligned} \tag{44}$$

and we have naturally $T \times G = \mathcal{S}(\boldsymbol{p}^k) \bigcup \mathcal{F}(\boldsymbol{p}^k)$ for all $\boldsymbol{p}^k \in \mathcal{Q}^{\mathrm{tot}}$, $\left|\mathcal{S}(\boldsymbol{p}^k)\right| = n_s^k$ and $\left|\mathcal{F}(\boldsymbol{p}^k)\right| = n_f^k = |T|\,|G| - n_s^k$. Using these sets, the projected generalized gradient can be efficiently split between a smooth and a nonsmooth part. Let $\boldsymbol{g}^k$ be the smooth part and $\boldsymbol{S}^k$ the matrix containing the nonsmooth parts to be combined,

$$\boldsymbol{S}^k = \left[P_{\boldsymbol{p}^k}\left(\nabla f_{s_1}^{\mathrm{VPE}}(p_{s_1}^k)\right), \dots, P_{\boldsymbol{p}^k}\left(\nabla f_{s_{n_s^k}}^{\mathrm{VPE}}(p_{s_{n_s^k}}^k)\right)\right] \in \mathbb{R}^{|T||G| \times n_s^k} \tag{45}$$

$$\boldsymbol{g}^k = P_{\boldsymbol{p}^k}\left(\nabla f^{\mathrm{Q}}(\boldsymbol{p}^k) + \sum_{(g,t)\in\mathcal{F}(\boldsymbol{p}^k)} \nabla f_g^{\mathrm{VPE}}(p_{gt}^k)\right) \in \mathbb{R}^{|T||G|} \tag{46}$$

The subproblem 41 can be rewritten as

$$\boldsymbol{\lambda}^k = \arg \min_{-1 \leqslant \boldsymbol{\lambda} \leqslant 1} \left|\left|\boldsymbol{g}^k + \boldsymbol{S}^k \boldsymbol{\lambda}\right|\right|_2^2, \tag{47}$$

and the admissible descent direction $\boldsymbol{d}^k$ is computed as $\boldsymbol{d}^k = -(\boldsymbol{g}^k + \boldsymbol{S}^k \boldsymbol{\lambda}^k)$. The subproblem (47) is a convex quadratic programming (QP) problem of dimension $n_s^k \leqslant |T|\,|G|$ much easier to solve than any problem from Table 1.

Notice that until now, the unique considered constraint is (10), hence *admissible* refers to this sole constraint. The generalization of the descent direction on a *constrained manifold*, such as the feasible set of (P) is made hereafter.

Descent direction on constrained manifold If, as in (P), the feasible set is a manifold further constrained by q linear constraints under the form $\boldsymbol{c}_i{}^\intercal \boldsymbol{p} \leqslant 0$ with $i = 1 \dots q$, we define the matrix $\boldsymbol{C}^k$ of the projected active constraints at point $\boldsymbol{p}^k$ whose columns are given by

$$\boldsymbol{C}^k_{*,j} = \boldsymbol{c}_j \quad \text{for all } j \in \{1 \dots q\} \text{ such that } \boldsymbol{c}_j{}^\intercal \boldsymbol{p}^k = 0. \tag{48}$$

We have $\boldsymbol{C}^k \in \mathbb{R}^{|T||G| \times n_c^k}$ with $0 \leqslant n_c^k \leqslant q$.

The subproblem (47) becomes

$$(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) = \arg \min_{\substack{-1 \leqslant \boldsymbol{\lambda} \leqslant 1 \\ \boldsymbol{\mu} \geqslant 0}} \left|\left| \boldsymbol{g}^k + \boldsymbol{S}^k \boldsymbol{\lambda} + \boldsymbol{C}^k \boldsymbol{\mu} \right|\right|_2^2, \tag{Sub}$$

and the descent direction $\boldsymbol{d}^k = -(\boldsymbol{g}^k + \boldsymbol{S}^k \boldsymbol{\lambda}^k + \boldsymbol{C}^k \boldsymbol{\mu}^k)$.

Note that the dimension of (Sub), the number of decision variables, is comprised between 0 and $q\, n_s^k \ll 2^{|G||T|}$. Furthermore, for a point locally smooth located in the interior of the domain, (Sub) is trivial and the direction is given by the gradient: $\boldsymbol{d}^k = -\boldsymbol{g}^k = -\nabla f(\boldsymbol{p}^k)$.

In order to complete the description of the line-search scheme, it remains to choose a stepsize rule and a stopping criterion.

Stopping criterion and step rule It can be shown that the direction $\boldsymbol{d}^k$ at a stationary point $\boldsymbol{p}^k$ yields the zero vector [5]. Hence, a natural stopping criterion is to monitor the direction norm. Unfortunately, as studied in [8], that type of criterion on the norm of the KKT violation is not reliable as this norm varies nonsmoothly around stationary points. Hence, the second criterion used here is the size of the step size α^k, if the stepsize has to be too small for the point to be admissible, the algorithm stops.

A common practice for the stepsize is to use the so-called *Armijo's rule*. This rule ensures that the step α^k at iteration k makes the next iterate $\boldsymbol{p}^{k+1} = \mathrm{R}(\boldsymbol{p}^k, \alpha^k \boldsymbol{d}^k)$ feasible while sufficiently decreasing the objective. An explicit implementation of Armijo's rule is given in [4, Algorithm 3], we slightly modify it such that it returns 0 if $\boldsymbol{d}^k = \boldsymbol{0}$ (up to a given tolerance) and -1 if no step size above a given treshold is found.

Implementation details It appears that for problems which are large enough, the subproblem (Sub) becomes inconsistent, in the sense that the direction obtained is only admissible in a tiny neighbourhood around the previous iterate. This can arise when a variable p_{gt} is twice tighted, e.g., $p_{gt} = P_g^-$ (constraint 8 is tight) and $p_{gt} - p_{g(t-1)} = R_g^+$ (constraint (9) is tight). To remedy this situation, the variable is frozen at its value and is no longer a decision variable. This allows to temporarily free the algorithm which may find an other direction for which an admissible stepsize is available. This procedure as well as the complete method to obtain a feasible solution and improve it is given in Fig. 8.

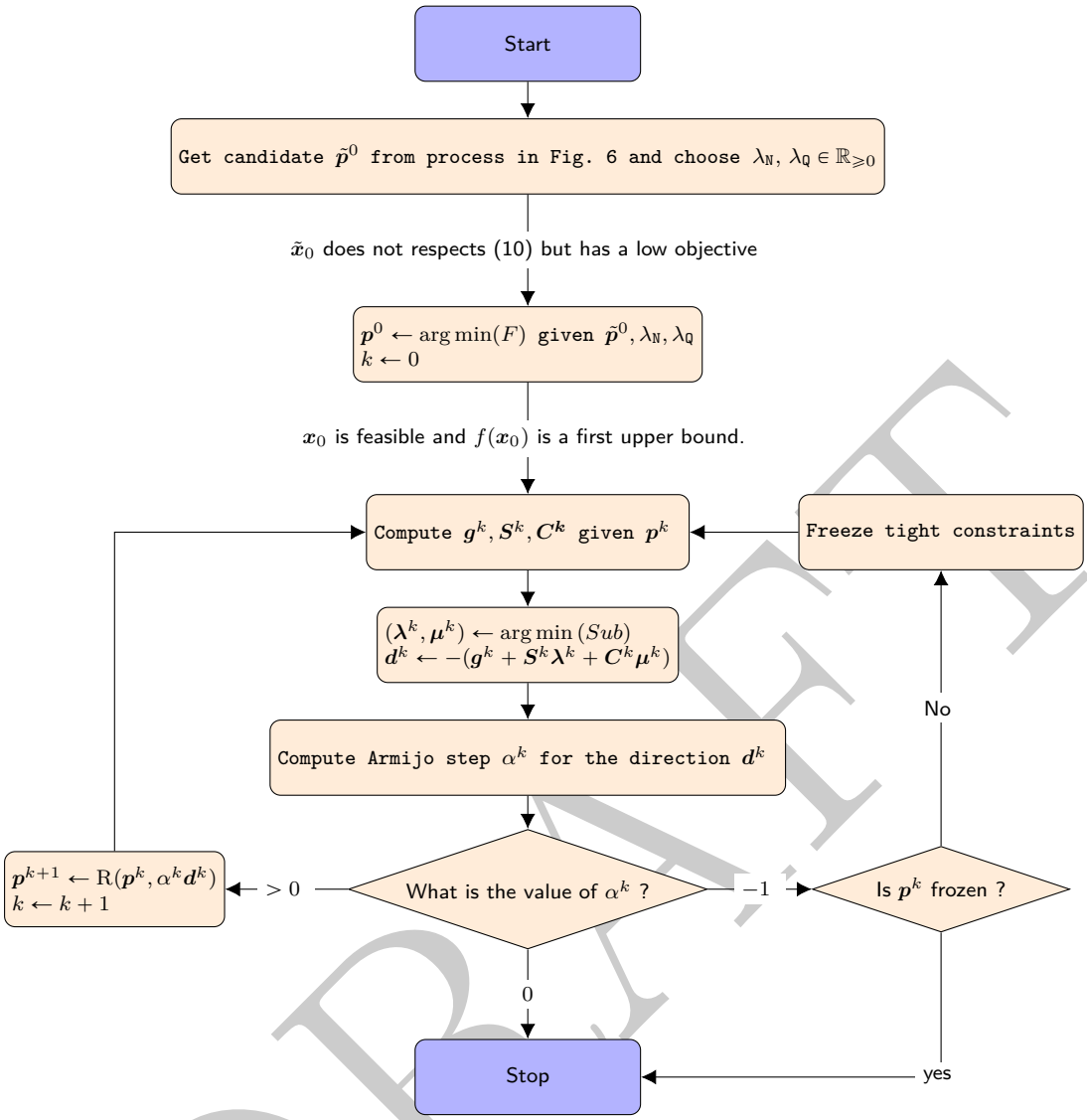


Fig. 8: Flowchart of the method to obtain a lower bound along with a first (infeasible) candidate with low objective.

4 Test cases

In a similar fashion as [19], the method is tested on several data set with different number of units and a time horizon of 24 timesteps. For each data set, the best objective (or upper bound) is reported along with the best lower

Table 2: Summary results: 5-unit case

Method	Cost			S-Time	Loss (MW)	Deviation (MW)	Lower bound
	Min	Avg	Max				
BBOSB[27]	43018	43066	43197	NA	194.65	0.01	
HIGA[15]	43125	43162	43259	1.37	194.79	0.074	
ICA[16]	43117	43144	43210	NA	194.80	0.014	
MILP-IPM[19]	43084			0.58	195.26	0.00095	
APLA	43250			0.38	193.98	1.6e-9	42527.85
APLA-RSG	43092			0.5	194.18	2e-11	42527.85

bound. The optimal gap, defined as the difference between the best known upper and lower bound is also given.

In order to account for the different processor speed from other methods, the scaled CPU time is used [19]

$$\text{Scaled CPU time} = \frac{\text{Given CPU Speed}}{\text{Base CPU speed}} \text{Given CPU time}\,, \tag{49}$$

where the base CPU time used in this paper is 3.6 GHz. The scaled CPU time is denoted as S-time.

4.1 5-unit test case

The data comes from [20], it consists of a 5-unit case where all units obey a valve-point effect. The reserve is set to 5% of the demand.

The comparison of the solution obtained with the proposed method with other methods from the literature is made in Table 2. The three first columns report the minimum, average and maximal solution. Deterministic methods only provide the first column. The best solution given in the literature is plugged into our model to compute the losses and the demand deviation defined as the violation of (10). Only the proposed method provides a lower bound which allows us to bound the finale optimality gap as 1.3 %.

The proposed method, APLA-RSG, obtains good objective with respect to others in the literature. It is however beaten by BBOSB, this should be taken in consideration with the fact that i) APLA-RSG provide a lower bounds and ii) fair comparison should take the running time into account. BBOSB only provides the number of function evaluation ($\sim 2e5$). Function evaluations (FEs) allow a fair comparison between methods run on different computers, however they cannot be computed in our case due to the call to the MIP solver in APLA. For the records, the RSG method needs 24750 FEs to converge. We can therefore estimate the equivalent FEs for the whole APLA-RSG as 100 000 which is half as small as BBOSB.

Table 3: Summary results: 10-unit case

Method	Cost			S-Time	Loss (MW)	Deviation (MW)	Lower bound
	Min	Avg	Max				
BBOSB[27]	1039169[4]	*	*	NA	818.22	83	
HIGA[15]	43125	43162	43259	2.06/1.5	194.79	0.074	
ICA[16]	43117	43144	43210	NA	194.80	0.014	
TSMILP [26]	1037487			1.9	832.32	0.013	
MILP-IPM[19]	1040676			0.75	882.74	0.0019	
APLA	1040475			1.6	882.02	1.9e-9	1032045
APLA-RSG	1038108			2.3	809.05	1.3e-11	1032045

4.2 10-unit test case

The data comes from [20] and consists of a 10-unit case. All units obey a VPE. Similarly as [26], the reserve is set to 3.5% of the demand and not 5%. As a matter of fact, the problem with 5% reserve is not feasible: this can be shown by looking at the static dispatch at the highest demand. Since (12) must hold for all t, we have

$$\sum_{g\in G} P_g^+ - (P_t^{\mathrm{D}} + \min_{\boldsymbol{p}_t\in\mathcal{P}_t} p_t^{\mathrm{loss}} + P_t^{\mathrm{S}}) \geqslant 0, \tag{50}$$

where $\mathcal{P}_t$ consists in the intersection of the power ranges $(8)_t$ with a relaxed version of the power balance $(10)_t$,

$$\sum_{g\in G} p_{gt} \geqslant P_t^{\mathrm{D}}. \tag{51}$$

This is a relaxation because negative losses p_t^{loss} are unphysical. It is clear that if the main problem (P) is feasible then (12) holds for all t which implies that (50) also holds for all t. Conversely, if (50) does not hold for any t then (P) must be unfeasible. In this test case, the highest demand occurs for $t = 12$ with $D_t = 2220$ MW and $P_t^{\mathrm{S}} = 111$ MW. The sum of the maximum power ranges is 2358 MW and the minimal power losses are computed as 49.7MW. Hence we conclude that the 10-unit case with 5% reserve is *not* feasible. This explains why [19], despite developing the method to take reserves into account, did not test the 10-unit test case with reserve and this also explains why [26] chose a 3.5% reserve instead of the usual 5%. It also raises questions about some methods from the literature reported in [26, Table V] which pretended to solve this unfeasible problem.

Table 3

[4] This value differs from the reported value of [27] and has been computed from the given solution of [27]. This may come from a mistake in the solution reported. This mistake could explain the high balance deviation.

5 Conclusion

In this work, a method to tackle a non-convex dispatch is developed. Non-convexities come from the inclusion of the valve-point effect, which is an important effect in operation of large gas units and the consideration of power losses modeled as a non-convex quadratic equation.

We show that the quadratic power losses can be defined in term of quadrics which exhibits a rich structure of a Riemannian manifold. The hypothesis of the positive definitiveness of the quadratic constraint is not made, as it is not always the case in practice, and we demonstrate how to construct tight relaxations whether the matrice is positive definite or not. The structure of Riemannian manifold is exploited and we show how to compute all elements needed for a subgradient Riemannian descent.

The proposed method, refered as APLA-RSG, consists in i) finding a lower bound and first candidate solution through the solution of a relaxation of the problem – the APLA part – and ii) projecting this candidate to the feasible set and locally improving it with a Riemannian subgradient descent – the RSG part.

Numerical experiments illustrate that the method reaches competitive objective in a similar amout of time as other methods from the literature. However, APLA-RSG benefits from other advantages, namely the fact that it provides a lower bound and strictly satisfies the balance constraint. The lower bound makes possible the estimation of the optimality gap, despite the fact that the problem is higly non-convex. These lower bounds could also help any further methods, to assess whether their solutions are good enough. Using a deterministic approach, instead of a stochastic metaheuristic, allows to limit potential mistakes, such as returning an unfeasible solution or returning a solution to an unfeasible problem. Both mistakes are exhibited in one of the case considered here.

Further work may include a further complexification of the model. The prohibited operation zones (POZ) could be easily applied to APLA, but then the local search should be limited to the given POZ. A better way of converting the unfeasible solution of APLA to a feasible one should also be investigated. Currently, it is possible to lose the optimality feature of the solution at the projection step. Finally, the extension of the method to optimal power flow and more specifically convex or non-convex ACOPF could also be contemplated.

References

1. Absil, P.A., Sluysmans, B., Stevens, N.: MIQP-based algorithm for the global solution of economic dispatch problems with valve-point effects. In: 2018 Power Systems Computation Conference (PSCC), pp. 1–7 (2018). DOI 10.23919/PSCC.2018.8450877
2. Arens, T., Hettlich, F., Karpfinger, C., Kockelkorn, U., Lichtenegger, K., Stachel, H.: Mathematik, 1. aufl. 2008 edn. Spektrum Akademischer Verlag (2008)
3. Attaviriyanupap, P., Kita, H., Tanaka, E., Hasegawa, J.: A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. IEEE Transactions on Power Systems **17**(2), 411–416 (2002). DOI 10.1109/TPWRS.2002.1007911

4. Borckmans, P.B., Selvan, S.E., Boumal, N., Absil, P.A.: A Riemannian subgradient algorithm for economic dispatch with valve-point effect. J. Comput. Applied. Math. **255**, 848–866 (2013). DOI 10.1016/j.cam.2013.07.002
5. Clarke, F.H.: A new approach to lagrange multipliers. Mathematics of Operations Research **1**(2), 165–174 (1976). DOI 10.1287/moor.1.2.165
6. Corporation, N.A.E.R.: Disturbance control standard – contingency reserve for recovery from a balancing contingency event (2018). URL `https://www.nerc.com/files/BAL-STD-002-0.pdf`
7. Decker, G.L., Brooks, A.D.: Valve point loading of turbines. Electrical Engineering **77**(6), 501–501 (1958). DOI 10.1109/EE.1958.6445133
8. Dutta, J., Deb, K., Tulshyan, R., Arora, R.: Approximate KKT points and a proximity measure for termination. Journal of Global Optimization **56**(4), 1463–1499 (2013). DOI 10.1007/s10898-012-9920-5. URL `https://doi.org/10.1007/s10898-012-9920-5`
9. European Commission: Energy roadmap 2050 (2011)
10. Kirchmayer, L.: Economic Operation of Power Systems. General Electric series. Wiley (1958). URL `https://books.google.be/books?id=sAUjAAAAMAAJ`
11. Kothari, D., Nagrath, I.: Modern Power System Analysis. Tata McGraw-Hill Publishing Company (2003). URL `https://books.google.be/books?id=zfxQFM3RkkEC`
12. Lee, J.M.: Introduction to smooth manifolds (2000)
13. Low, S.H.: Convex relaxation of optimal power flow—part II: Exactness. IEEE Transactions on Control of Network Systems **1**(2), 177–189 (2014). DOI 10.1109/TCNS.2014.2323634
14. Midcontinent Independent System Operator (MISO): Enhanced modeling of combined cycle generators. URL `https://www.misoenergy.org/stakeholder-engagement/issue-tracking/enhanced-modeling-of-combined-cycle-generators`. [Online; accessed 2020-05-28]
15. Mohammadi-Ivatloo, B., Rabiee, A., Soroudi, A.: Nonconvex Dynamic Economic Power Dispatch Problems Solution Using Hybrid Immune-Genetic Algorithm. IEEE Systems Journal **7**(4), 777–785 (2013). DOI 10.1109/JSYST.2013.2258747. Conference Name: IEEE Systems Journal
16. Mohammadi-ivatloo, B., Rabiee, A., Soroudi, A., Ehsan, M.: Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. Energy **44**(1), 228–240 (2012). DOI 10.1016/j.energy.2012.06.034. URL `http://www.sciencedirect.com/science/article/pii/S0360544212004859`
17. Niknam, T., Azizipanah-Abarghooee, R., Aghaei, J.: A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. IEEE Transactions on Power Systems **28**(2), 749–763 (2013). DOI 10.1109/TPWRS.2012.2208273
18. Niknam, T., Narimani, M.R., Azizipanah-Abarghooee, R.: A new hybrid algorithm for optimal power flow considering prohibited zones and valve point effect. Energy Conversion and Management **58**, 197 – 206 (2012). DOI https://doi.org/10.1016/j.enconman.2012.01.017
19. Pan, S., Jian, J., Yang, L.: A hybrid MILP and IPM approach for dynamic economic dispatch with valve-point effects. International Journal of Electrical Power & Energy Systems **97**, 290 – 298 (2018). DOI https://doi.org/10.1016/j.ijepes.2017.11.004
20. Panigrahi, C.K., Chattopadhyay, P.K., Chakrabarti, R.N., Basu, M.: Simulated annealing technique for dynamic economic dispatch **34**(5), 577–586. DOI 10.1080/15325000500360843. URL `https://www.tandfonline.com/doi/full/10.1080/15325000500360843`
21. Saadat, H.: Power System Analysis. McGraw-Hill, New York (1999)
22. Swamp, K.S., Natarajan, A.: Constrained optimization using evolutionary programming for dynamic economic dispatch. In: Proceedings of International Conference on Intelligent Sensing and Information Processing, pp. 314–319 (2005). DOI 10.1109/ICISIP.2005.1529468
23. Van Hoorebeeck, L., Papavasiliou, A., Absil, P.A.: MILP-based algorithm for the global solution of dynamic economic dispatch problems with valve-point effects. IEEE Power and Energy Society General Meeting (2019)
24. Van Hoorebeeck, L., Papavasiliou, A., Absil, P.A.: Global solution of economic dispatch with valve point effects and transmission constraints. Electric Power Systems Research (2020). Accepted.

25. Wong, K.P., Fung, C.C.: Simulated annealing based economic dispatch algorithm. IEEE Proceedings C - Generation, Transmission and Distribution **140**(6), 509–515 (1993). DOI 10.1049/ip-c.1993.0074
26. Wu, Z., Ding, J., Wu, Q.H., Jing, Z., Zheng, J.: Reserve constrained dynamic economic dispatch with valve-point effect: A two-stage mixed integer linear programming approach. CSEE Journal of Power and Energy Systems **3**(2), 203–211 (2017)
27. Xiong, G., Shi, D.: Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects. Energy **157**, 424–435 (2018). DOI 10.1016/j.energy.2018.05.180. URL `http://www.sciencedirect.com/science/article/pii/S0360544218310247`
28. Zhang, F.: The Schur Complement and its Applications, *Numerical Methods and Algorithms*, vol. 4. Springer, New York (2005). DOI 10.1007/b105056