# Power network design with line activity*

Daniel Bienstock, Martina Cerulli, Mauro Escobar, and Leo Liberti

**Abstract** We discuss the problem of optimally designing a power transportation network with respect to line activity. We model this problem as an alternating current optimal power flow with on/off variables on lines. We formulate this problem as a nonconvex MINLP in complex numbers, then we propose two convex MINLP relaxations. We test our formulations on some small-scale standard instances.

**Key words:** Optimal power flow, MINLP, relaxation.

## 1 Introduction

Every network routing problem naturally yields a design counterpart which optimally decides some part of the network topology. Network routing problems based on multicommodity flows yield design problems where arcs, nodes and/or other features are installed/removed according to flow cost and demand. Such optimization problems often arise in telecommunication networks [1], supply chain [2], logistics, and more. They are usually solved using a mixture of Mathematical Programming (MP) formulations of the mixed-integer sort, decomposition strategies, combinatorial algorithms, and heuristics. On the other hand, the first approach —which we

Daniel Bienstock
IEOR, Columbia University, USA
e-mail: dano@columbia.edu

Martina Cerulli, Mauro Escobar, Leo Liberti
LIX CNRS Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France
e-mail: {mcerulli,escobar,liberti}@lix.polytechnique.fr

* This paper has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement n. 764759.

follow in this paper— is always the deployment of off-the-shelf MP solvers on such problems.

Power networks are used to transport and distribute electricity. The transportation occurs at very high voltage levels (hundreds of thousands of volts), while distribution occurs at lower voltage levels (hundreds of volts). Typically, such networks are reasonably sparse, but have some cycles for redundancy-based protection. Moreover, these networks route Alternating Current (AC) rather than Direct Current (DC) [3]. The typical network routing problem for current is known as Optimal Power Flow (OPF). It is well known that the OPF for DC can be well approximated by a Linear Program (LP) (see [4, §1.2.4], [5, Eq. (5.48)]). On the other hand, the OPF for AC, commonly known as ACOPF, is the object of intense research [4, 5] because of its difficulty and importance.

We shall see in the following that the ACOPF can be naturally formulated in MP in many ways, e.g. Quadratically Constrained Quadratic Programming (QCQP), Polynomial Programming (PP), and general Nonlinear Programming (NLP), all of which involve nonconvexities [4]. Common relaxations are LP, Second-Order Cone Programs (SOCP), Semidefinite Programs (SDP) [5]. The variables (voltage, current, power) are naturally defined on continuous domains. A very interesting feature of the ACOPF is that its variables range in complex numbers. While a separation in real and imaginary parts is always possible, matrix formulations and relaxations generally take up twice the amount of storage w.r.t. working directly in complex numbers [6].

Network design problems defined on the OPF in DC can be readily formulated as Mixed-Integer Linear Programs (MILP) [7, 8]. This is also done for problems arising in grid robustness analysis [9, 10], where binary variables model attacks and vulnerabilities [4, Ch. 3]. Binary variables in the ACOPF have also been used to discretize continuous variables arising in nonconvex constraints, so as to obtain an approximate reformulation turning nonconvexities into a finite set of binary choices [5, §4.3.5-4.3.6], which can be dealt with using standard Mixed-Integer Programming (MIP) solvers.

To the best of our knowledge, the first paper exhibiting computational results for the ACOPF with binary variables used for design (rather than approximation) purposes is [11], where binary variables are used to switch generators and shunts on and off: a local NLP solver is deployed on a well-known continuous NLP reformulation of the corresponding nonconvex MINLP. A perspective cut based relaxation of an ACOPF formulation with binary variables for switching generators on and off was proposed in [12]. Another possible approach for working with ACOPF involving binary variables is to apply network design modeling techniques involving binary variables to an LP or SOCP relaxation of the ACOPF. This was done in [13], which proposed inner and outer mixed-integer Diagonally Dominant Programming (DDP) formulations. DDP [14] is a MP technique to approximate the Semidefinite (PSD) cone using LP. The ACOPF is **NP**-hard [15], and remains hard even when the goal is to minimize the number of active generators [12].

In this paper we move a step towards solving a "network design ACOPF" by integrating binary variables that control whether a line is active or not. Our objective is to decrease the number of active lines while still satisfying demand. While this is similar to the optimal switching problem [16], here we start from the ACOPF formulation rather than its DC counterpart. We shall present a (nonconvex) MINLP formulation of the network design problem derived from the ACOPF, and two convex MINLP relaxations. While there is little hope of solving even the tiniest ACOPF instances with the nonconvex MINLP, we show that some results for small ACOPF instances can be obtained using convex MINLPs.

The rest of this paper is organized as follows. We present the ACOPF formulation and a nonconvex MINLP formulation for the corresponding network design problem in Sect. 2. We then propose some new mixed-integer SOCP (MISOCP) relaxations in Sect. 3. We test our formulations with some standard instances in Sect. 4.

## 2 The ACOPF formulation

Modeling the ACOPF can be daunting. Most of the literature refers to Matlab-style modeling: painstakingly filling the correct components of a huge constraint matrix with the correct values. This is the low-level kind of interface to MP solvers which produces "flat" formulations that can be read directly by solvers: extremely fast in execution, but a debugging nightmare. See [4, Eg. 1.2.1] and [17, 18] for some introductory material.

Today, most MP formulations are presented in "structured" form: index sets first, then parameters, decision variables, objective function(s), and constraints, all parametrized by and quantified over the aforementioned indices. Each MP entity (parameter, variable, objective, constraint) is stored in a multi-dimensional jagged array, possibly not completely defined. Structured formulations convey the problem definition much more clearly than flat ones, at least to MP-versed readers. Detailed formulations can be found in [19]. Modeling tools such as AMPL [20] allow for fast(er) debugging.

We model the electrical network as a loopless multi-digraph $G=(B,L)$ where $B$ is the set of nodes and $L$ the set of arcs. In power engineering terminology a node is called a *bus* and an arc is called a *line* or *branch*. We assume $|B|=n$ and $|L|=m$. Parallel arcs occur whenever parallel cables are deployed on connections that must transport excessive amounts of power for a single cable. The $h$-th line $\ell_{bah}$ joining two buses $b$ and $a$ is represented by a pair of anti-parallel arcs $\ell_{bah} = \{(b,a,h),(a,b,h)\}$. We assume that $L$ is partitioned in two sets $L_0, L_1$ with $|L_0| = |L_1|$: for each pair of antiparallel arcs, one is in $L_0$ and the other in $L_1$, according to the asymmetry of the branch admittance matrix $\mathbf{Y}_{bah}$ matrix below.

Ohm's law expresses the current $I_{bah}$ injected on a line $\ell_{bah}$ in function of the voltages $V_b, V_a$ at the endpoints $b$ and $a$, and of the physical properties of the line. The fundamental difference with Ohm's law in DC is that AC yields an asymmetry. While in DC we have $I_{bah} = -I_{abh}$, in AC we instead have:

$$\forall (b,a,h) \in L_0 \quad I_{bah} = Y^{\mathsf{ff}}_{bah} V_b + Y^{\mathsf{ft}}_{bah} V_a \quad \wedge \quad I_{abh} = Y^{\mathsf{tf}}_{bah} V_b + Y^{\mathsf{tt}}_{bah} V_a. \qquad (1)$$

The $Y$ constants in the above equations are defined as follows [4, 19]:

$$\mathbf{Y}_{bah} = \begin{pmatrix} Y^{\mathsf{ff}}_{bah} & Y^{\mathsf{ft}}_{bah} \\ Y^{\mathsf{tf}}_{bah} & Y^{\mathsf{tt}}_{bah} \end{pmatrix} = \begin{pmatrix} \left(\frac{1}{r_{bah}+ix_{bah}} + i\frac{\mathfrak{b}_{bah}}{2}\right)/\tau^2_{bah} & -\frac{1}{(r_{bah}+ix_{bah})\tau_{bah}e^{-i\nu_{bah}}} \\ -\frac{1}{(r_{bah}+ix_{bah})\tau_{bah}e^{i\nu_{bah}}} & \frac{1}{r_{bah}+ix_{bah}} + i\frac{\mathfrak{b}_{bah}}{2} \end{pmatrix}, \quad (2)$$

where $r, x, \mathfrak{b}, \tau, \nu$ measure some physical properties of the line, and are given as part of the instance. The suffixes ff, ft, tf, tt to $Y$ stand for "from-from", "from-to", "to-from", and "to-to": they are a reminder of the direction of the routed quantities w.r.t. the line $\ell_{bah}$.

We can now introduce sets, parameters and decision variables of the ACOPF.

- *Sets:* $B$, $L$ and a set $\mathscr{G}$ of generators partitioned as $\{\mathscr{G}_b \mid b \in B\}$, where $\mathscr{G}_b$ contains the generators attached to bus $b$.
- *Parameters:* power demand (or *load*) $\tilde{S}_b$, *shunt admittance* $A_b$; voltage magnitude bounds $\underline{V}_b, \overline{V}_b$ at each bus $b \in B$; *admittance matrix* $\mathbf{Y}_{bah}$; upper bound $\bar{S}_{bah}$ to injected power magnitude; lower/upper bounds $\underline{\eta}_{bah}, \overline{\eta}_{bah}$ to phase difference at each line $(b,a,h) \in L$; cost coefficients $C_{g2}, C_{g1}, C_{g0}$; lower/upper bounds $\underline{\mathscr{S}}_g, \overline{\mathscr{S}}_g$ to power generated at $g \in \mathscr{G}$; a *reference bus* $r \in B$.
- *Decision variables*: voltage $V_b$ at bus $b \in B$, injected current $I_{bah}$, injected power $S_{bah}$ at each line $(b,a,h) \in L$, and generated power $\mathscr{S}_g$ at each generator $g \in \mathscr{G}$.

All variables range in $\mathbb{C}$. Among the parameters, the power magnitude, voltage magnitude, phase difference bounds, cost coefficients are in $\mathbb{R}$; $r$ ranges in the bus set; the generated power bounds are in $\mathbb{C}$. Limited to this paper we assume that, for two complex numbers $\alpha = \alpha^{\mathsf{r}} + i\alpha^{\mathsf{c}}$ and $\beta = \beta^{\mathsf{r}} + i\beta^{\mathsf{c}}$, $\alpha \le \beta$ means $\alpha^{\mathsf{r}} \le \beta^{\mathsf{r}}$ and $\alpha^{\mathsf{c}} \le \beta^{\mathsf{c}}$. We also recall that $|\alpha| = \sqrt{(\alpha^{\mathsf{r}})^2 + (\alpha^{\mathsf{c}})^2}$ is the magnitude of $\alpha$, that $\alpha^* = \alpha^{\mathsf{r}} - i\alpha^{\mathsf{c}}$ is the conjugate of $\alpha$, and that $|\alpha|^2 = \alpha\,\alpha^*$.

We present now objective function and constraints of what we call the $(S,I,V)$-formulation of the ACOPF.

- *Objective function:* $\min \sum\limits_{g\in\mathscr{G}} (C_{g2}(\mathscr{S}^{\mathsf{r}}_g)^2 + C_{g1}\mathscr{S}^{\mathsf{r}}_g + C_{g0})$, which is quadratic and separable in generated power.
- *Bound constraints:* on voltage magnitude $\underline{V}_b^2 \le |V_b|^2 \le \overline{V}_b^2$ for each $b \in B$; on power magnitude $|S_{bah}|^2 \le \bar{S}^2_{bah}$ for each $(b,a,h) \in L$; on phase difference $\tan(\underline{\eta}_{bah})(V_b V_a{}^*)^{\mathsf{r}} \le (V_b V_a{}^*)^{\mathsf{c}} \le \tan(\overline{\eta}_{bah})(V_b V_a{}^*)^{\mathsf{r}}$ together with $(V_b V_a{}^*)^{\mathsf{r}} \ge 0$ for $(b,a,h) \in L_0$; on generated power $\underline{\mathscr{S}}_g \le \mathscr{S}_g \le \overline{\mathscr{S}}_g$ for each $g \in \mathscr{G}$. Moreover, we have $V_r^{\mathsf{c}} = 0$ and $V_r^{\mathsf{r}} \ge 0$ on the reference bus.
- *Functional constraints:*
  - Power flow equations:

$$\forall b \in B \quad \sum_{(b,a,h)\in L} S_{bah} + \tilde{S}_b = -A_b{}^*|V_b|^2 + \sum_{g\in\mathscr{G}_b} \mathscr{S}_g. \quad (3)$$

  - The relationship between $S, V, I$:

$$\forall (b,a,h) \in L \quad S_{bah} = V_b I_{bah}{}^*. \quad (4)$$

– Ohm's laws Eq. (1), which we write equivalently as:

$$\forall (b,a,h) \in L_0 \quad I_{bah} = Y^{\mathsf{ff}}_{bah} V_b + Y^{\mathsf{ft}}_{bah} V_a \tag{5}$$

$$\forall (b,a,h) \in L_1 \quad I_{bah} = Y^{\mathsf{tf}}_{abh} V_a + Y^{\mathsf{tt}}_{abh} V_b. \tag{6}$$

### 2.1 The network design ACOPF

We now introduce a binary variable $y_{bah}$ for each $(b,a,h)$ in $L$. We have $y_{bah} = 1$ iff the corresponding line is active, and we must ensure that both antiparallel arcs are active/inactive at the same time by $y_{bah} = y_{abh}$. We control activation/deactivation of a line by vanishing its effect on the power flow equations, and ignoring the power magnitude bound when the line is inactive:

$$\forall b \in B \qquad \sum_{(b,a,h)\in L} S_{bah} y_{bah} + \tilde{S}_b = -A_b{}^* |V_b|^2 + \sum_{g\in\mathscr{G}_b} \mathscr{S}_g, \tag{7}$$

$$\forall (b,a,h) \in L \qquad (|S_{bah}| y_{bah})^2 \le \bar{S}^2_{bah}. \tag{8}$$

Note that Eq. (7)-(8) does not cut the global optima of the ACOPF: it suffices to set $y_{bah} = 1$ for each $(b,a,h) \in L$ to see this. Instead, we add an objective function $\min \sum_{(b,a,h)\in L_0} y_{bah}$. We can tackle this bi-objective MINLP either by scalarization or by adding a constraint $\sum_{(b,a,h)\in L_0} y_{bah} \le \xi$ and letting $\xi$ vary in $\{1,\ldots,m/2\}$. In this paper we consider scalarization approach, so that the objective function becomes:

$$\min \sum_{g\in\mathscr{G}} (C_{g2}(\mathscr{S}^{\mathsf{r}}_g)^2 + C_{g1}\mathscr{S}^{\mathsf{r}}_g + C_{g0}) + \rho \sum_{(b,a,h)\in L_0} y_{bah}, \tag{9}$$

where $\rho > 0$ is a scalar weight which we set to 1 for testing purposes. We denote the network design ACOPF problem with binary variables on lines by $\text{ACOPF}_{\mathsf{L}}$.

A few preliminary attempts to solve this formulation with the tiniest possible instance (`case5`, see Sect. 4 below) showed that it could not be solved in 500s by the state-of-the-art BARON [21] solver.

## 3 Linearizing relaxations

The material in this section is motivated by the solution difficulty posed by the nonconvex MINLP formulation of the $\text{ACOPF}_{\mathsf{L}}$. First of all we propose some valid relaxation for ACOPF problem.

The decision variables $I$ for current can be eliminated from the $(S,I,V)$-formulation by replacing them in Eq. (4) with their expressions in Eq. (5)-(6). This yields the $(S,V)$-formulation, which is still a nonconvex NLP. In turn, using Eq. (7)-(9), this NLP yields a nonconvex MINLP formulation for the $\text{ACOPF}_{\mathsf{L}}$.

### 3.1 $(S,V,X)$-relaxation

The only nonlinear terms appearing in the nonconvex constraints of the ACOPF $(S,V)$-formulation are quadratic in voltage: they are products $V_b V_a{}^*$ for some $b,a \in B$. Every such product term can be linearized, i.e. replaced by a new (complex) variable $X_{ba}$ for $b,a \in B$ (we do not include the corresponding defining constraint $X_{ba} = V_b V_a{}^*$). Let us call this the $(S,V,X)$-relaxation. This turns out to be a convex QCQP (more specifically a SOCP). The quadratic terms are: $\mathscr{S}_g^2$ in the minimizing objective and $|S_{bah}|^2$ in the LHS of the power magnitude bound constraints.

### 3.2 $(S,V,X)$-SDP

Note that the $(S,V,X)$-relaxation is an exact reformulation if we enforce $X = VV^{\mathsf{H}}$, where the apex stands for "hermitian transpose", i.e. the transpose of the componentwise complex conjugate. Accordingly, since $X$ is a PSD rank-one matrix, we get a stronger relaxation w.r.t. the $(S,V,X)$-relaxation presented in Sec. 3.1, if we replace $X = VV^{\mathsf{H}}$ by $X \succeq 0$, which yields a complex SDP relaxation called $(S,V,X)$-SDP.

### 3.3 $(S,V,X)$-$\frac{1}{2}$DDP

Given the scarcity of off-the-shelf mixed-integer SDP solvers, we consider a DDP approximation of the PSD cone [14]: since every Diagonally Dominant (DD) matrix is also PSD [22], the constraint "$X$ is DD" yields an inner approximation (i.e. a restriction) of the complex SDP.

Writing the DDP constraints corresponding to $X \succeq 0$ requires splitting $X$ into real and imaginary parts, which yields $\bar{X} = \begin{pmatrix} X^{\mathsf{rr}} & X^{\mathsf{rc}} \\ X^{\mathsf{cr}} & X^{\mathsf{cc}} \end{pmatrix} \in \mathbb{R}^{2n \times 2n}$, where $X^{\mathsf{rr}} = (X^{\mathsf{r}}_{ba})$, $X^{\mathsf{cc}} = (X^{\mathsf{c}}_{ba})$, $X^{\mathsf{rc}}$ linearizes the matrix $(V^{\mathsf{r}}_b V^{\mathsf{c}}_a)$, and $X^{\mathsf{cr}}$ linearizes the matrix $(V^{\mathsf{c}}_b V^{\mathsf{r}}_a)$. We remark that $X^{\mathsf{rr}}, X^{\mathsf{cc}}$ are symmetric matrices, while $X^{\mathsf{rc}}, X^{\mathsf{cr}}$ are not; on the other hand, $X^{\mathsf{rc}}_{ba} = X^{\mathsf{cr}}_{ab}$ for each $b,a \in B$.

Now the DDP inner approximation of $\bar{X} \succeq 0$ states that any diagonal component of $\bar{X}$ is greater than or equal to the sum of the absolute values of the components in the same row. This corresponds to:

$$\forall b \in B \quad X^{\mathsf{rr}}_{bb} \geq \sum_{\substack{a \in B \\ a \neq b}} T^{\mathsf{rr}}_{ba} + \sum_{a \in B} T^{\mathsf{rc}}_{ba} \tag{10}$$

$$\forall b \in B \quad X^{\mathsf{cc}}_{bb} \geq \sum_{\substack{a \in B \\ a \neq b}} T^{\mathsf{cc}}_{ba} + \sum_{a \in B} T^{\mathsf{cr}}_{ba}, \tag{11}$$

where $\bar{T} = \begin{pmatrix} T^{\mathsf{rr}} & T^{\mathsf{rc}} \\ T^{\mathsf{cr}} & T^{\mathsf{cc}} \end{pmatrix}$ is a real variable matrix such that $-\bar{T} \leq \bar{X} \leq \bar{T}$ [14].

The issue with inner DDP approximations is that they may be infeasible even if the corresponding SDP is feasible. Experimentally, this was verified to be the case in every ACOPF instance we tested. This issue can be addressed algorithmically [14], but this would require solving a sequence of DDPs, which would in turn take excessive time. Instead, we chose to only impose Eq. (10), which yielded feasible "half-DDP" relaxations (which we refer to as $(S,V,X)$-$\frac{1}{2}$DDP relaxation) of the tested ACOPF instances. Note that we do not have a general feasibility proof for $\frac{1}{2}$DDP relaxations. So far, we have not found any counterexamples yet, either.

### 3.4 Jabr relaxation

Another SOCP relaxation of the ACOPF, called "Jabr relaxation", was proposed in [23]. It can be constructed from the $(S,V)$-formulation as follows:

1. transform cartesian coordinates $V^{\mathsf{r}}, V^{\mathsf{c}}$ to polar coordinates $v, \theta$ by replacing $V^{\mathsf{r}} = v\cos\theta$ and $V^{\mathsf{c}} = v\sin\theta$: this will result with nonlinear terms in $v_b v_a \cos(\theta_b - \theta_a)$ and $v_b v_a \sin(\theta_b - \theta_a)$;
2. define an index set $R = \{(b,b) \mid b \in B\} \cup \{(b,a) \mid (b,a,1) \in L\}$;
3. linearize (replace) the nonlinear terms with new variables $c_{ba} = v_b v_a \cos(\theta_b - \theta_a)$ and $s_{ba} = v_b v_a \sin(\theta_b - \theta_a)$ for all $(b,a) \in R$: this also yields $c_{ba} = c_{ab}$, $s_{ba} = -s_{ab}$, $c_{ba}^2 + s_{ba}^2 = v_b^2 v_a^2$ $(\star)$ for all $(b,a,1) \in L_0$, as well as $s_{bb} = 0$ and $c_{bb} = v_b^2$ $(\dagger)$ for each $b \in B$;
4. replace $v_b^2, v_a^2$ in $(\star)$ with $c_{bb}, c_{aa}$ by means of $(\dagger)$, and relax $(\star)$ to a convex (conic) constraint $c_{ba}^2 + s_{ba}^2 \le c_{bb} c_{aa}$;
5. replace $|V_b|^2$ in the voltage magnitude bounds with $c_{bb}$;
6. remark that $V_b V_a^* = c_{ba} + i s_{ba}$, and infer the phase difference bounds as $c_{ba} \ge 0$ and $\tan(\underline{\eta}_{bah}) c_{ba} \le s_{ba} \le \tan(\overline{\eta}_{bah}) c_{ba}$ for each $(b,a,h) \in L_0$;
7. the injected power variables $S_{bah}$ satisfy the linear equations:

$$\begin{aligned}
\forall (b,a,h) \in L_0 \quad (S_{bah})^{\mathsf{r}} &= (Y^{\mathsf{ff}}_{bah})^{\mathsf{r}} c_{bb} + (Y^{\mathsf{ft}}_{bah})^{\mathsf{r}} c_{ba} + (Y^{\mathsf{ft}}_{bah})^{\mathsf{c}} s_{ba} \\
\forall (b,a,h) \in L_0 \quad (S_{bah})^{\mathsf{c}} &= -(Y^{\mathsf{ff}}_{bah})^{\mathsf{c}} c_{bb} + (Y^{\mathsf{ft}}_{bah})^{\mathsf{r}} s_{ba} - (Y^{\mathsf{ft}}_{bah})^{\mathsf{c}} c_{ba} \\
\forall (b,a,h) \in L_1 \quad (S_{bah})^{\mathsf{r}} &= (Y^{\mathsf{tt}}_{abh})^{\mathsf{r}} c_{bb} + (Y^{\mathsf{tf}}_{abh})^{\mathsf{r}} c_{ba} + (Y^{\mathsf{tf}}_{abh})^{\mathsf{c}} s_{ba} \\
\forall (b,a,h) \in L_1 \quad (S_{bah})^{\mathsf{c}} &= -(Y^{\mathsf{tt}}_{abh})^{\mathsf{c}} c_{bb} + (Y^{\mathsf{tf}}_{abh})^{\mathsf{r}} s_{ba} - (Y^{\mathsf{tf}}_{abh})^{\mathsf{c}} c_{ba}.
\end{aligned}$$

### 3.5 ACOPF$_{\mathsf{L}}$ relaxations

We derive ACOPF$_{\mathsf{L}}$ relaxations from the $(S,V,X)$-relaxation, the $(S,V,X)$-$\frac{1}{2}$DDP and Jabr relaxations of the ACOPF, by employing the binary variables $y$ as in Sect. 2.1, i.e. by imposing Eq. (7)-(8) and minimizing Eq. (9). We linearize the bilinear terms $S_{bah} y_{bah}$ in the $(S,V,X)$-relaxation and the $(S,V,X)$-$\frac{1}{2}$DDP relaxation, by means of McCormick inequalities [24], introducing a new variable $z_{bah}$ for each $S_{bah} y_{bah}$ together with some new linear constraints involving $z_{bah}, S_{bah}$, and $y_{bah}$, as well as their bounds. The same is done for Jabr relaxation of the ACOPF$_\mathsf{L}$, by defining new variables related to each bilinear term $c_{bb}y_{bah}$, $c_{ba}y_{bah}$, and $s_{ba}y_{bah}$ respectively. A few preliminary results showed that the active lines do not form a connected set at the optimum. In order to enforce connectivity, we therefore also added a set of multicommodity flow constraints on added variables $f^{ba}_{deh}$, defined for each distinct pair $b,a \in B$ and line $(d,e,h) \in L$:

$$\forall b < a \in B \quad \sum_{(b,d,h)\in L} f^{ba}_{bdh} - \sum_{(d,b,h)\in L} f^{ba}_{dbh} = 1 \tag{12}$$

$$\forall b < a \in B \quad \sum_{(d,a,h)\in L} f^{ba}_{dah} - \sum_{(a,d,h)\in L} f^{ba}_{adh} = 1 \tag{13}$$

$$\forall b < a \in B, d \in B \smallsetminus \{b,a\} \quad \sum_{(e,d,h)\in L} f^{ba}_{edh} - \sum_{(d,e,h)\in L} f^{ba}_{deh} = 0, \tag{14}$$

as well as the linking constraints

$$\forall b < a \in B, (d,e,h) \in L \quad f^{ba}_{deh} \le y_{deh}. \tag{15}$$

In Table 1, we shall refer to the ACOPF$_\mathsf{L}$ relaxations from $(S,V,X)$-$\frac{1}{2}$DDP, and Jabr as “ddp”, and “Jabr” respectively.

## 4 Computational experiments

The standard reference testbed for computational assessments in ACOPF is the PGLib library [25], which also includes “case files” from MATPOWER [18]. We compare performances of the two convex MINLP relaxations of the ACOPF$_\mathsf{L}$ (ddp and Jabr) on the small case instances `case`$i$ for $i \in \{5, 9, 14, 18, 22, 24, 30\}$. Our implementation is carried out in AMPL [20]. We solve both formulations, which are of the Mixed-Integer SOCP sort, with CPLEX 12.9 [26], which is given 300s as maximum CPU time. Only instance “`case5`” is solved using Baron, because AMPL failed to successfully pass it to Cplex.

The results in Table 1 are obtained on a a 2.53GHz Intel(R) Xeon(R) CPU with 49.4 GB RAM. They show that 300s are only sufficient to obtain meaningful results for small instances.

An encouraging feature of the results in Table 1 is that the slacker ddp relaxation takes less time to solve than Jabr, provides a worst bound, but still identifies a valid connectivity for active lines for most of the instances. In Fig. 1, e.g., we report two solutions found by solving the ddp relaxation, which appear to be the same found by Jabr relaxation.

In Table 2 we report results from the $(S,V,X)$-relaxation of ACOPF$_\mathsf{L}$ on slightly larger instances, solved using CPLEX limited to 7200s. When solutions are found atypically quickly (e.g. `case69`, `case85`), it is because the networks have no cycles.

| name | lines | known | mod1 | mod2 | opt1 | opt2 | act1 | act2 | stat1 | stat2 | cpu1 | cpu2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `case5` | 6 | 17551.89 | ddp | Jabr | 0.00 | **14999.70** | 0 | 6 | infeas | limit | 29.67 | 300 |
| `case9` | 9 | 5296.67 | ddp | Jabr | 2244.81 | **5296.66** | 9 | 9 | solved | solved | **12.90** | 33.18 |
| `case14` | 20 | 8081.52 | ddp | Jabr | 0.00 | **8076.60** | 1 | 17 | infeas | solved | 0.10 | **55.76** |
| `case18` | 17 | 11.85 | ddp | Jabr | 0.00 | **11.85** | 17 | 17 | solved | solved | **0.25** | 0.34 |
| `case22` | 21 | 0.068 | ddp | Jabr | 0.00 | **0.068** | 21 | 21 | solved | solved | **0.34** | 0.37 |
| `case24` | 38 | 63352.20 | ddp | Jabr | 47359.8 | **63344.75** | 38 | 38 | limit | limit | 300 | 300 |
| `case30` | 41 | 576.89 | ddp | Jabr | 0.00 | **573.79** | 29 | 41 | limit | limit | 300 | 300 |

**Table 1** Numerical results limited to 300s using a single CPU processor. We report instance name, number of lines, known optimal value; then, for each relaxation type (mod1,mod2), we report obtained optimal value, number of active lines, solver status, CPU time. Best results are in boldface, invalid results are grayed (the solver could not find any feasible solution in the allotted time).

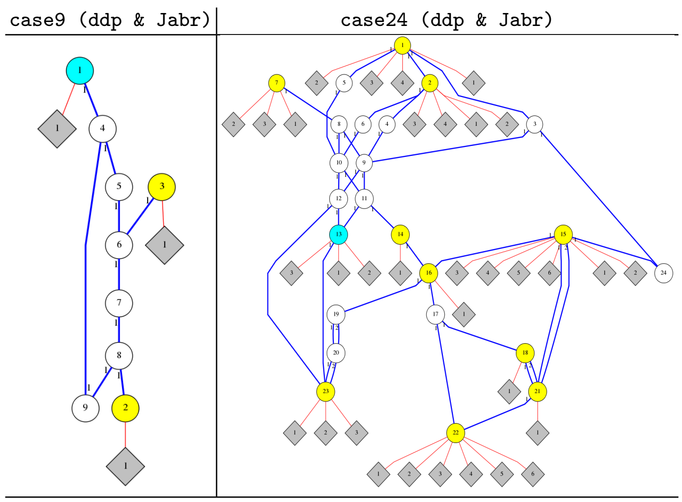


**Fig. 1** Two solutions from results in Table 1. Buses in circles, generators in parallelograms (buses with generators are colored, reference bus is colored differently); active lines are thick and colored.

| name | lines | known | opt | act | stat | cpu |
|---|---|---|---|---|---|---|
| `case24` | 38 | 63352.20 | 47272.28 | 34 | limit | 7200 |
| `case30` | 41 | 576.89 | 0.00 | 29 | limit | 7200 |
| `case39` | 46 | 41864.17 | 27417.26 | 46 | limit | 7200 |
| `case69` | 68 | 0.39 | 0.00 | 68 | solved | 8.26 |
| `case85` | 84 | 0.00 | 0.00 | 84 | solved | 18.75 |

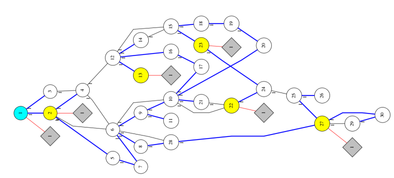

**Table 2** Computational results on the $(S,V,X)$-relaxation limited to 7200s (left). Nontrivial solution for `case30` (right).

## References

1. B. Gendron, T. Crainic, and A. Frangioni. Multicommodity capacitated network design. In B. Sansò and P. Soriano, editors, *Telecommunications Network Planning*, volume CRT 25th Anniversary Series, pages 1–32. Springer, New York, 1999.

2. M. Bruglieri and L. Liberti. Optimal running and planning of a biomass-based energy production process. *Energy Policy*, 36:2430–2438, 2008.
3. J. Glover, M. Sarma, and T. Overbye. *Power system analysis and design*. Cengage Learning, Stamford CT, 2010.
4. D. Bienstock. *Electrical Transmission System Cascades and Vulnerability: an Operations Research Viewpoint*. Number 22 in MOS-SIAM Optimization. SIAM, Philadelphia, 2016.
5. D. Molzahn and I. Hiskens. *A survey of relaxations and approximations of the power flow equations*. Number 4(1-2) in Foundations and Trends in Electric Energy Systems. now Publishers, Hanover, MA, 2019.
6. J.-C. Gilbert and C. Josz. Plea for a semidefinite optimization solver in complex numbers. Technical Report hal-01422932, HAL Archives-Ouvertes, 2017.
7. U. Knight. The logical design of electrical networks using linear programming methods. *Proceedings of IEE*, 107A(2):306–314, 1960.
8. R. Adams and M. Laughton. Optimal planning of power networks using mixed-integer programming: Part 1—Static and time-phased network synthesis. *Proceedings of IEE*, 121(2):139–146, 1974.
9. D. Bienstock and S. Mattia. Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization*, 4:115–141, 2007.
10. D. Bienstock and A. Verma. The *n-k* problem in power grids: new models, formulations and computation. *SIAM Journal on Optimization*, 20(5):2352–2380, 2010.
11. M. Ruiz, J. Maeght, A. Marié, P. Panciatici, and A. Renaud. A progressive method to solve large-scale AC optimal power flow with discrete variables and control of the feasibility. In *Proceedings of the Power Systems Computation Conference*, volume 18 of *PSCC*. IEEE, 2014.
12. E. Salgado, C. Gentile, and L. Liberti. Perspective cuts for the ACOPF with generators. In P. Daniele and L. Scrimali, editors, *New trends in emerging complex real-life problems*, volume 1 of *AIRO Springer Series*. Springer, New York, 2018.
13. E. Salgado, A. Scozzari, F. Tardella, and L. Liberti. Alternating current optimal power flow with generator selection. In J. Lee, G. Rinaldi, and R. Mahjoub, editors, *Combinatorial Optimization (Proceedings of ISCO 2018)*, volume 10856 of *LNCS*, pages 364–375, 2018.
14. A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.
15. D. Bienstock and A. Verma. Strong NP-hardness of AC power flows feasibility. *Operations Research Letters*, 47:494–501, 2019.
16. E.B. Fisher, R. O'Neill, and M.C. Ferris. Optimal Transmission Switching. *IEEE Transactions on Power Systems*, 23(3):1346–1355, 2008.
17. D. Molzahn, J. Holzer, B. Lesieutre, and C. DeMarco. Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Transactions on Power Systems*, 28(4):3987–3998, 2013.
18. R. Zimmermann and C. Murillo-Sánchez. *Matpower 7.0b1 User's Manual*. Power Systems Engineering Research Center, 2018.
19. D. Bienstock, M. Escobar, C. Gentile, and L. Liberti. Mathematical programming formulations for the alternating current optimal power flow problem. *4OR*, 18(3):249–292, 2020.
20. R. Fourer and D. Gay. *The AMPL Book*. Duxbury Press, Pacific Grove, 2002.
21. N.V. Sahinidis and M. Tawarmalani. *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2005.
22. S. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. *Izvestia Akademii Nauk USSR*, 6:749–754, 1931.
23. R. Jabr. Radial distribution load flow using conic programming. *IEEE Transactions on Power Systems*, 21(3):1458–1459, 2006.
24. G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10:146–175, 1976.
25. S. Babaeinejadsarookolaee, A. Birchfield, R. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, and et al. The power grid library for benchmarking AC optimal power flow algorithms. Technical Report 1908.02788, arXiv, 2019.
26. IBM. *ILOG CPLEX 12.9 User's Manual*. IBM, 2019.