Strengthening Convex Relaxations with Bound Tightening for Power Network Optimization

Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck

NICTA* - Optimisation Research Group, Australia
College of Engineering and Computer Science, Australian National University
{carleton.coffrin,hassan.hijazi,pvh}@nicta.com.au

Abstract. Convexification is a fundamental technique in (mixed-integer) nonlinear optimization and many convex relaxations are parametrized by variable bounds, i.e., the tighter the bounds, the stronger the relaxations. This paper studies how bound tightening can improve convex relaxations for power network optimization. It adapts traditional constraint-programming concepts (e.g., minimal network and bound consistency) to a relaxation framework and shows how bound tightening can dramatically improve power network optimization. In particular, the paper shows that the Quadratic Convex relaxation of power flows, enhanced by bound tightening, almost always outperforms the state-of-the-art Semi-Definite Programming relaxation on the optimal power flow problem.

Keywords: Continuous Constraint Networks, Minimal Network, Bound Consistency, Convex Relaxation, AC Power Flow, QC Relaxation, AC Optimal Power Flow

1 Introduction

In (mixed-integer) nonlinear optimization, convexification is used to obtain dual bounds, complementing primal heuristics. In many cases, these convex relaxations are parametrized by variable bounds and the tighter the bounds are, the stronger the relaxations. There is thus a strong potential for synergies between convex optimization and constraint programming. This paper explores these synergies in the context of power system optimization.

The power industry has been undergoing a fundamental transformation in recent years. Deregulation, the emergence of power markets, pressure for reduced capital investment, and the need to secure a clean sustainable energy supply all stress the importance of efficiency and reliability in the design and operation of power networks. As a result, optimization has become a critical component of the emerging *smart-grid* [28] and has resulted in millions of dollars in annual savings [32].

Power network applications range from long-term network design and investment tasks [21,7,12] to minute-by-minute operation tasks [23,14,19,16,17]. All

* NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

of these optimization problems share a common core, the Alternating Current (AC) power flow equations, which model the steady-state physics of power flows. These equations form a system of continuous non-convex nonlinear equations that prove to be a significant challenge for existing general-purpose optimization tools. It is thus not surprising that, in the last decade, significant attention has been devoted to developing computationally efficient convex relaxations.

The main contribution of this paper is to show that constraint programming can substantially improve the quality of convex relaxations for power flow applications. To obtain this result, the paper defines the concept of constraint relaxation networks and generalizes traditional consistency notions to these networks, including minimal network and bound consistency. These concepts, and the associated algorithms, are then applied to optimal power flow applications with and without load uncertainty. The experimental results demonstrate the significant value of bound tightening for power flow applications. In particular,

1. Bound tightening reduces the domains of the variables by as much as 90% in many cases.
2. In over 90% of the test cases considered, propagation over the convex relaxation was sufficient to close the optimality gap within 1%. Only 4 of the test cases considered remain open.
3. The network consistency algorithm improves the quality of the Quadratic Convex (QC) relaxation [18] considerably. The QC relaxation now outperforms, in the vast majority of the cases, the established state-of-the-art Semi-Definite Programming (SDP) relaxation on the optimal power flow problem.
4. Parallelization can significantly reduce the runtime requirements of bound tightening, making the proposed algorithms highly practical.

The rest of the paper is organized as follows. Section 2 reviews the AC power flow feasibility problem and introduces the notations. Section 3 reviews the state-of-the-art QC power flow relaxation, which is essential for building efficient consistency algorithms. Section 4 formalizes the idea of constraint relaxation networks and Section 5 applies this formalism to AC power flows. Section 6 studies the quality of bound tightening in this application domain and Section 7 evaluates the proposed methods on the ubiquitous AC Optimal Power Flow problem. Section 8 illustrates the potential of the proposed methods on power flow applications incorporating uncertainty and Section 9 concludes the paper.

2 AC Power Flow

A power network is composed of a variety of components such as buses, lines, generators, and loads. The network can be interpreted as a graph (N, E) where the set of buses N represent the nodes and the set of lines E represent the edges. Note that E is a set of directed arcs and E^R will be used to indicate those arcs in the reverse direction. To break numerical symmetries in the model and to allow easy comparison of solutions, a reference node $r \in N$ is also specified.

Every node $i \in N$ in the network has three properties, voltage $V_i = v_i \angle \theta_i$, power generation $S^g_i = p^g_i + \boldsymbol{i} q^g_i$, and power consumption $S^d_i = p^d_i + \boldsymbol{i} q^d_i$, all of which are complex numbers due to the oscillating nature of AC power. Each line $(i,j) \in E$ has an admittance $Y_{ij} = g_{ij} + \boldsymbol{i} b_{ij}$, also a complex number. These network values are connected by two fundamental physical laws, Kirchhoff's Current Law (KCL),

$$S^g_i - \boldsymbol{S^d_i} = \sum_{(i,j)\in E \cup E^R} S_{ij} \;\; \forall\, i \in N \tag{1}$$

and Ohm's Law,

$$S_{ij} = \boldsymbol{Y}^*_{ij}(V_i V_i^* - V_i V_j^*) \;\; \forall\, (i,j) \in E \cup E^R. \tag{2}$$

Note that bold values indicate parameters that are constant in the classic AC power flow problem and non-bold values are the decision variables.

In addition to these physical laws, the following operational constraints are required in AC power flows. Generator output limitations on S^g,

$$\boldsymbol{S^{gl}_i} \leqslant S^g_i \leqslant \boldsymbol{S^{gu}_i} \;\; \forall i \in N. \tag{3}$$

Line thermal limits on S_{ij},

$$|S_{ij}| \leqslant \boldsymbol{s^u_{ij}} \;\; \forall (i,j) \in E \cup E^R. \tag{4}$$

Bus voltage limits on V_i,

$$\boldsymbol{v^l_i} \leqslant |V_i| \leqslant \boldsymbol{v^u_i} \;\; \forall i \in N \tag{5}$$

and line phase angle difference limits on $V_i V_j^*$,

$$\boldsymbol{\theta^{\Delta l}_{ij}} \leqslant \angle\big(V_i V_j^*\big) \leqslant \boldsymbol{\theta^{\Delta u}_{ij}} \;\; \forall (i,j) \in E \tag{6}$$

Note that power networks are designed and operated so that $-\boldsymbol{\pi}/3 \leqslant \boldsymbol{\theta^{\Delta l}} \leqslant \boldsymbol{\theta^{\Delta u}} \leqslant \boldsymbol{\pi}/3$ [22] and values as low as $\boldsymbol{\pi}/18$ are common in practice [33]. Additionally the values of $\boldsymbol{v^l}, \boldsymbol{v^u}, \boldsymbol{s^u}$ must be positive as they are bounds on the magnitudes of complex numbers.

Combining all of these constraints and expanding them into their real-number representation yields the AC Power Flow Feasibility Problem (AC-PF) presented in Model 1. The input data is indicated by bold values and a description of the decision variables is given in the model. Constraint (7a) sets the reference angle. Constraints (7b)–(7c) capture KCL and constraints (7d)–(7e) capture Ohm's Law. Constraints (7f) link the phase angle differences on the lines to the bus variables and constraints (7g) enforce the thermal limit on the lines. This particular formulation of AC-PF is advantageous as the auxiliary variables θ^Δ, p, and q isolate the problem's non-convexities in constraints (7d)–(7e) and enable all but one of the operational constraints to be captured by the variable bounds. This continuous constraint satisfaction problem is NP-Hard in general [40,24] and forms a core sub-problem that underpins a wide variety of power network optimization tasks.

Model 1 The AC Power Flow Feasibility Problem (AC-PF)

variables:

$p^g_i \in (\boldsymbol{p}^{gl}_i, \boldsymbol{p}^{gu}_i)\;\; \forall i \in N$ - active power generation

$q^g_i \in (\boldsymbol{q}^{gl}_i, \boldsymbol{q}^{gu}_i)\;\; \forall i \in N$ - reactive power generation

$v_i \in (\boldsymbol{v}^l_i, \boldsymbol{v}^u_i)\;\; \forall i \in N$ - bus voltage magnitude

$\theta_i \in (-\infty, \infty)\;\; \forall i \in N$ - bus voltage angle

$\theta^\Delta_{ij} \in (\boldsymbol{\theta}^{\Delta l}_{ij}, \boldsymbol{\theta}^{\Delta u}_{ij})\;\; \forall (i,j) \in E$ - angle difference on a line (aux.)

$p_{ij} \in (-\boldsymbol{s}^u_{ij}, \boldsymbol{s}^u_{ij})\;\; \forall (i,j) \in E \cup E^R$ - active power flow on a line (aux.)

$q_{ij} \in (-\boldsymbol{s}^u_{ij}, \boldsymbol{s}^u_{ij})\;\; \forall (i,j) \in E \cup E^R$ - reactive power flow on a line (aux.)

subject to:

$$\theta_{\boldsymbol{r}} = 0 \tag{7a}$$

$$p^g_i - \boldsymbol{p}^d_i = \sum_{(i,j)\in E \cup E^R} p_{ij}\;\; \forall i \in N \tag{7b}$$

$$q^g_i - \boldsymbol{q}^d_i = \sum_{(i,j)\in E \cup E^R} q_{ij}\;\; \forall i \in N \tag{7c}$$

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta^\Delta_{ij}) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta^\Delta_{ij})\;\; (i,j) \in E \cup E^R \tag{7d}$$

$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta^\Delta_{ij}) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta^\Delta_{ij})\;\; (i,j) \in E \cup E^R \tag{7e}$$

$$\theta^\Delta_{ij} = \theta_i - \theta_j\;\; \forall (i,j) \in E \tag{7f}$$

$$p_{ij}^2 + q_{ij}^2 \leqslant (\boldsymbol{s}^u_{ij})^2\;\; \forall (i,j) \in E \cup E^R \tag{7g}$$

To address the computational difficulties of AC-PF, convex relaxations (i.e. polynomial time) have attracted significant interest in recent years. Such relaxations include the Semi-Definite Programming (SDP) [2], Second-Order Cone (SOC) [20], Convex-DistFlow (CDF) [13], and the recent Quadratic Convex (QC) [18] relaxations. To further improve these relaxations, this paper proposes consistency notions and associated propagation algorithms for AC power flows. A detailed evaluation on 57 AC transmission system test cases demonstrates that the propagation algorithms enable reliable and efficient methods for improving these relaxations on a wide variety of power network optimization tasks via industrial-strength convex optimization solvers (e.g., Gurobi, Cplex, Mosek). The next section reviews the QC relaxation in detail, which forms the core of the proposed propagation algorithms.

3 The Quadratic Convex (QC) Relaxation

The QC relaxation [18] was introduced to utilize the bounds on the voltage variables v and θ^Δ, which are ignored by the other relaxations. The key idea is to use the variable bounds to derive convex envelopes around the non-convex aspects of the AC-PF problem. The derivation begins by lifting the voltage product terms in to the higher dimensional W-space using the following equalities:

$$w_i = v_i^2 \;\; i \in N \tag{8a}$$
$$w_{ij}^R = v_i v_j \cos(\theta_{ij}^\Delta) \;\; \forall (i,j) \in E \tag{8b}$$
$$w_{ij}^I = v_i v_j \sin(\theta_{ij}^\Delta) \;\; \forall (i,j) \in E \tag{8c}$$

When Model 1 is lifted into this W-space, all of the remaining constraints are convex. On its own, this lifted model is a weak relaxation but the QC relaxation strengthens it by developing convex relaxations of the nonlinear equations (8a)–(8c) for the operational bounds on variables v and θ^Δ. The convex envelopes for the square and bilinear functions are well-known [27], i.e.,

$$\langle x^2 \rangle^T \equiv \begin{cases} \breve{x} \geqslant x^2 \\ \breve{x} \leqslant (\boldsymbol{x^u} + \boldsymbol{x^l})x - \boldsymbol{x^u}\boldsymbol{x^l} \end{cases} \tag{T-CONV}$$

$$\langle xy \rangle^M \equiv \begin{cases} \breve{xy} \geqslant \boldsymbol{x^l}y + \boldsymbol{y^l}x - \boldsymbol{x^l}\boldsymbol{y^l} \\ \breve{xy} \geqslant \boldsymbol{x^u}y + \boldsymbol{y^u}x - \boldsymbol{x^u}\boldsymbol{y^u} \\ \breve{xy} \leqslant \boldsymbol{x^l}y + \boldsymbol{y^u}x - \boldsymbol{x^l}\boldsymbol{y^u} \\ \breve{xy} \leqslant \boldsymbol{x^u}y + \boldsymbol{y^l}x - \boldsymbol{x^u}\boldsymbol{y^l} \end{cases} \tag{M-CONV}$$

Under the assumption that the phase angle difference bound is within $-\boldsymbol{\pi}/2 \leqslant \boldsymbol{\theta^{\Delta l}} \leqslant \boldsymbol{\theta^{\Delta u}} \leqslant \boldsymbol{\pi}/2$, relaxations for sine and cosine are given by

$$\langle \cos(x) \rangle^C \equiv \begin{cases} \breve{cx} \leqslant \cos(x) \\ \breve{cx} \geqslant \frac{\cos(\boldsymbol{x^l}) - \cos(\boldsymbol{x^u})}{(\boldsymbol{x^l} - \boldsymbol{x^u})}(x - \boldsymbol{x^l}) + \cos(\boldsymbol{x^l}) \end{cases} \tag{C-CONV}$$

$$\langle \sin(x) \rangle^S \equiv \begin{cases} \breve{sx} \leqslant \cos\left(\frac{\boldsymbol{x^m}}{2}\right)\left(x - \frac{\boldsymbol{x^m}}{2}\right) + \sin\left(\frac{\boldsymbol{x^m}}{2}\right) & \text{if } \boldsymbol{x^l} < 0 \wedge \boldsymbol{x^u} > 0 \\ \breve{sx} \geqslant \cos\left(\frac{\boldsymbol{x^m}}{2}\right)\left(x + \frac{\boldsymbol{x^m}}{2}\right) - \sin\left(\frac{\boldsymbol{x^m}}{2}\right) & \text{if } \boldsymbol{x^l} < 0 \wedge \boldsymbol{x^u} > 0 \\ \breve{sx} \leqslant \sin(x) & \text{if } \boldsymbol{x^l} \geqslant 0 \\ \breve{sx} \geqslant \frac{\sin(\boldsymbol{x^l}) - \sin(\boldsymbol{x^u})}{(\boldsymbol{x^l} - \boldsymbol{x^u})}(x - \boldsymbol{x^l}) + \sin(\boldsymbol{x^l}) & \text{if } \boldsymbol{x^l} \geqslant 0 \\ \breve{sx} \leqslant \frac{\sin(\boldsymbol{x^l}) - \sin(\boldsymbol{x^u})}{(\boldsymbol{x^l} - \boldsymbol{x^u})}(x - \boldsymbol{x^l}) + \sin(\boldsymbol{x^l}) & \text{if } \boldsymbol{x^u} \leqslant 0 \\ \breve{sx} \geqslant \sin(x) & \text{if } \boldsymbol{x^u} \leqslant 0 \end{cases} \tag{S-CONV}$$

where $\boldsymbol{x^m} = \max(-\boldsymbol{x^l}, \boldsymbol{x^u})$. These are a generalization of the relaxations proposed in [18] to support asymmetrical bounds on x. Utilizing these building blocks, convex relaxations for equations (8a)–(8c) can be obtained by composing relaxations of the subexpressions, for example, $w_{ij}^R \equiv \langle \langle v_i v_j \rangle^M \langle \cos(\theta_i - \theta_j) \rangle^C \rangle^M$. Lastly, the QC relaxation proposes to strengthen these convex relaxations with a valid second-order cone constraint [20,18,11],

$$(w_{ij}^R)^2 + (w_{ij}^I)^2 \leqslant w_i w_j \;\; \forall (i,j) \in E \tag{9}$$

Model 2 The QC Power Flow Feasibility Problem (QC-PF)

variables: Variables of Model 1

$st_{ij} \in (-1, 1) \;\; \forall (i,j) \in E$ - relaxation of the sine (aux.)

$ct_{ij} \in (0, 1) \;\; \forall (i,j) \in E$ - relaxation of the cosine (aux.)

$vv_{ij} \in (\boldsymbol{v}^{\boldsymbol{l}}_i \boldsymbol{v}^{\boldsymbol{l}}_j, \boldsymbol{v}^{\boldsymbol{u}}_i \boldsymbol{v}^{\boldsymbol{u}}_j) \;\; \forall (i,j) \in E$ - relaxation of the voltage product (aux.)

$w_i \in \left((\boldsymbol{v}^{\boldsymbol{l}}_i)^2, (\boldsymbol{v}^{\boldsymbol{u}}_i)^2\right) \;\; \forall i \in N$ - relaxation of the voltage square (aux.)

$w^R_{ij} \in (0, \infty) \;\; \forall (i,j) \in E$ - relaxation of the voltage and cosine product (aux.)

$w^I_{ij} \in (-\infty, \infty) \;\; \forall (i,j) \in E$ - relaxation of the voltage and sine product (aux.)

subject to: (7a)–(7c),(7f)–(7g)

$$\text{CONV}(w_i = v_i^2 \in (\boldsymbol{v}^{\boldsymbol{l}}_i, \boldsymbol{v}^{\boldsymbol{u}}_i)) \;\; \forall i \in N \tag{10a}$$

$$\text{CONV}(ct_{ij} = \cos(\theta^{\Delta}_{ij}) \in (\boldsymbol{\theta}^{\boldsymbol{\Delta l}}_{ij}, \boldsymbol{\theta}^{\boldsymbol{\Delta u}}_{ij})) \;\; \forall (i,j) \in E \tag{10b}$$

$$\text{CONV}(st_{ij} = \sin(\theta^{\Delta}_{ij}) \in (\boldsymbol{\theta}^{\boldsymbol{\Delta l}}_{ij}, \boldsymbol{\theta}^{\boldsymbol{\Delta u}}_{ij})) \;\; \forall (i,j) \in E \tag{10c}$$

$$\text{CONV}(vv_{ij} = v_i v_j \in (\boldsymbol{v}^{\boldsymbol{l}}_i, \boldsymbol{v}^{\boldsymbol{u}}_i) \times (\boldsymbol{v}^{\boldsymbol{l}}_j, \boldsymbol{v}^{\boldsymbol{u}}_j)) \;\; \forall (i,j) \in E \tag{10d}$$

$$\text{CONV}(w^R_{ij} = vv_{ij} ct_{ij} \in (\boldsymbol{vv}^{\boldsymbol{l}}_{ij}, \boldsymbol{vv}^{\boldsymbol{u}}_{ij}) \times (\boldsymbol{ct}^{\boldsymbol{l}}_{ij}, \boldsymbol{ct}^{\boldsymbol{u}}_{ij}) \;\; \forall (i,j) \in E \tag{10e}$$

$$\text{CONV}(w^I_{ij} = vv_{ij} st_{ij} \in (\boldsymbol{vv}^{\boldsymbol{l}}_{ij}, \boldsymbol{vv}^{\boldsymbol{u}}_{ij}) \times (\boldsymbol{st}^{\boldsymbol{l}}_{ij}, \boldsymbol{st}^{\boldsymbol{u}}_{ij}) \;\; \forall (i,j) \in E \tag{10f}$$

$$(w^R_{ij})^2 + (w^I_{ij})^2 \leqslant w_i w_j \;\; \forall (i,j) \in E \tag{10g}$$

$$p_{ij} = \boldsymbol{g}_{ij} w_i - \boldsymbol{g}_{ij} w^R_{ij} - \boldsymbol{b}_{ij} w^I_{ij} \;\; \forall (i,j) \in E \tag{10h}$$

$$q_{ij} = -\boldsymbol{b}_{ij} w_i + \boldsymbol{b}_{ij} w^R_{ij} - \boldsymbol{g}_{ij} w^I_{ij} \;\; \forall (i,j) \in E \tag{10i}$$

$$p_{ji} = \boldsymbol{g}_{ij} w_j - \boldsymbol{g}_{ij} w^R_{ij} + \boldsymbol{b}_{ij} w^I_{ij} \;\; \forall (i,j) \in E \tag{10j}$$

$$q_{ji} = -\boldsymbol{b}_{ij} w_j + \boldsymbol{b}_{ij} w^R_{ij} + \boldsymbol{g}_{ij} w^I_{ij} \;\; \forall (i,j) \in E \tag{10k}$$

The complete QC relaxation of the AC-PF problem is presented in Model 2 (QC-PF), which incorporates many of the components of Model 1. In the model, the constraint $\text{CONV}(y = f(x) \in D)$ is used to indicate that y lies in a convex relaxation of function f within the domain D. Constraints (10a)–(10f) implement the convex relaxations and constraints (10g) further strengthen these relaxations. Constraints (10h)–(10k) capture the line power flow in terms of the W-space variables.

The Impact of Tight Bounds in the QC Relaxation: Constraints (10a)–(10f) in Model 2 highlight the critical role that the bounds on v and θ^{Δ} play in the strength of the QC relaxation. Figure 1 illustrates this point by showing the convex relaxations for sine and cosine over the domains $\theta^{\Delta} \in (-\boldsymbol{\pi}/3, \boldsymbol{\pi}/3)$ and $\theta^{\Delta} \in (-\boldsymbol{\pi}/3, 0)$. This figure indicates two key points: (1) Although the reduction in the size of the bound is 50% in this case, the area inside of the convex relaxations have been reduced even more significantly; (2) Both the sine and cosine functions are monotonic when the sign of θ^{Δ} is known, which produces tight convex relaxations.

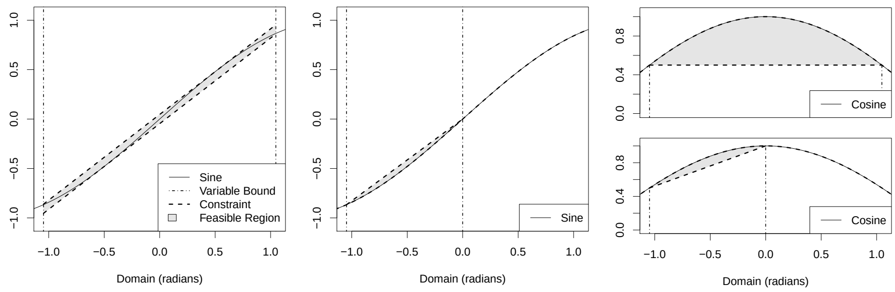

Fig. 1. The Impact of Variable Bounds on the Convex Relaxations.

4 Consistency of Constraint Relaxation Networks

As discussed in Section 2, the core of many applications in power network optimization is a continuous constraint network. Moreover, the QC relaxation of this continuous constraint network depends on the bounds of the variables. As a result, constraint propagation now has two benefits: On one hand, it reduces the domains of the variables while, on the other hand, it strengthens the relaxation. These two processes reinforce each other, since tighter constraints generate tighter bounds creating a virtuous cycle. This section generalizes traditional consistency notions to this new context through the concepts of constraint schemes and constraint relaxations. Since solutions to continuous constraint networks are real numbers and computer implementations typically rely on floating-point numbers, some care must be exercised in formalizing these notions. The formalization also assumes that only bound reasoning is of interest, since these are continuous constraint networks. However, the concepts generalize naturally to domain reasoning.

Continuous Constraint Networks Constraint networks are defined in terms of a set of variables $X = \{x_1, \ldots, x_n\}$ ranging over intervals $I = \{I_1, \ldots, I_n\}$ and a set of constraints. An interval $I = [l, u]$ denotes the set of real numbers $\{r \in \Re \mid l \leqslant r \leqslant u\}$. This paper only considers floating-point intervals, i.e., intervals whose bounds are floating-point numbers. If r is a real number, $[r]$ denotes the smallest floating-point interval containing r, $[r]^-$ the largest floating-point number no greater than r, and $[r]^+$ the smallest floating-point number no smaller than r. A variable assignment assigns to each variable x_i a value from its interval I_i. A constraint is a function $(X \to \Re) \to Bool$ which, given a variable assignment, returns a truth value denoting whether the assignment satisfies the constraint.

Definition 1 (Continuous Constraint Network (CCN)). *A continuous constraint network is a triple* (X, I, C) *where* $X = (x_1, \ldots, x_n)$ *is a collection of variables ranging over* $I = (I_1, \ldots, I_n)$ *and* C *is a set of constraints.*

Definition 2 (Solution to a CCN). *A solution to a CCN* $(X, \mathcal{I}, C)$*, where* $X = (x_1, \ldots, x_n)$ *and* $I = (I_1, \ldots, I_n)$*, is an assignment* $\sigma = \{x_1 \leftarrow v_1; \ldots; x_n \leftarrow$

$v_n\}$ such that $v_i \in I_i$ and for all $c \in C$: $c(\sigma)$ holds. The set of solutions to a CCN $\mathcal{P}$ is denoted by $\Sigma(\mathcal{P})$.

In the following we use $\max(x, \Sigma)$ to denote the maximum value of variable x in the assignments Σ, i.e., $\max(x, \Sigma) = \max_{\sigma \in \Sigma} \sigma(x)$, where $\sigma(x)$ denotes the value of variable x in assignment σ. The value $\min(x, \Sigma)$ is defined similarly. The following definition adapts the traditional concept of minimal constraint network [31] to continuous constraint networks.

Definition 3 (Minimal CCN). *A CCN $\mathcal{P} = (X, I, C)$, where $X = (x_1, \ldots, x_n)$ and $I = (I_1, \ldots, I_n)$, is minimal if, for each variable x_i, the interval $I_i = [l_i, u_i]$ satisfies $l_i = [\min(x_i, \Sigma(\mathcal{P}))]^- \ \wedge\ u_i = [\max(x_i, \Sigma(\mathcal{P}))]^+$.*

Note that the bounds are not necessarily solutions themselves but are as tight as the floating-point accuracy allows for. Given a CCN $\mathcal{P} = (X, I, C)$, its largest minimal network $\mathcal{P}' = (\mathcal{X}, \mathcal{I}_{\Updownarrow}, \mathcal{C})$ $(I_m \subseteq I)$ always exists and is unique since there are only finitely many floating-point intervals.

The concept of bound consistency [39] captures a relaxation of the minimal network: It only requires the variable bounds to be tight locally for each constraint.

Definition 4 (Bound Consistency for CCNs). *A CCN $\mathcal{P} = (X, I, C)$, where $X = (x_1, \ldots, x_n)$ and $I = (I_1, \ldots, I_n)$, is bound-consistent if each constraint c is bound-consistent with respect to I. A constraint c is bound-consistent with respect to I if the continuous constraint network $(X, I, \{c\})$ is minimal.*

Once again, given a CCN $\mathcal{P} = (X, I, C)$, its largest bound-consistent network $\mathcal{P} = (X, I_m, C)$ $(I_m \subseteq I)$ always exists and is unique. In the following, we use $minCCN(X, I, C)$ and $bcCCN(X, I, C)$ to denote these networks, i.e.,

$$\begin{aligned} \text{MINCCN}(X, I, C) &= \max\{I_m \subseteq I \mid (X, I_m, C) \text{ is minimal}\}, \\ \text{BCCCN}(X, I, C) &= \max\{I_m \subseteq I \mid (X, I_m, C) \text{ is bound-consistent}\}. \end{aligned}$$

Constraint Relaxation Networks The convex relaxations used in the QC relaxation depend on the variable bounds, i.e., the stronger the bounds the stronger the relaxations. Since the relaxations change over time, it is necessary to introduce new consistency notions: constraint schemes and constraint relaxations.

Definition 5 (Continuous Constraint Scheme). *A constraint scheme r is a function $\mathcal{I} \to (X \to \Re) \to Bool$ which, given a collection of intervals, returns a constraint. Moreover, the scheme r satisfies the following monotonicity property:*

$$I \subseteq I' \Rightarrow (r(I')(\sigma) \Rightarrow r(I)(\sigma))$$

for all collections of intervals I and I', and variable assignment σ.

The monotonicity property ensures that tighter bounds produce tighter constraints. Traditional constraints are constraint schemes that just ignore the initial bounds. A constraint relaxation is a constraint scheme that preserves the solutions to the original constraint.

Definition 6 (Constraint Relaxation). *A constraint scheme r is a relaxation of constraint c if, for all assignment σ and bounds* $I = ([\sigma(x_1)], \ldots, [\sigma(x_n)])$, *we have* $r(I)(\sigma) \Rightarrow c(\sigma)$.[1]

Example 1. Consider the constraint $c(x, y, z)$ which holds if $z = xy$. Given bounds $[x^l, x^u]$ and $[y^l, y^u]$ for variables x and y, the McCormick relaxation [27] is a constraint scheme specified by the collection of constraints in M-CONV. Note that this envelope ignores the bound on variable z. Additionally this constraint scheme is also a constraint relaxation of c because it is known to be the convex envelope of $z = xy$ for any bounds on x and y [27].

Definition 7 (Continuous Constraint Relaxation Network (CCRN)). *A constraint relaxation network is a triple* (X, I, R) *where* X *is a collection of variables ranging over* I *and* R *is a set of constraint relaxations.*

In the following, we use $R(I)$ to denote $\{r(I) \mid r \in R\}$ if R is a set of relaxations.

Consistency of Constraint Relaxation Networks We now generalize the concepts of minimal and bound-consistent networks to CCRNs. The definitions capture the fact that no additional bound tightening is possible for the relaxations induced by the bounds.

Definition 8 (Minimal CCRN). *A CCRN* $\mathcal{P} = (X, I, R)$, $X = (x_1, \ldots, x_n)$ *and* $I = (I_1, \ldots, I_n)$, *is minimal if the CCN network* $(X, I, R(I))$ *is.*

Definition 9 (Bound-Consistent CCRN). *A CCRN* $\mathcal{P} = (X, I, R)$, *where* $X = (x_1, \ldots, x_n)$ *and* $I = (I_1, \ldots, I_n)$, *is bound-consistent if the CCN network* $(X, I, R(I))$ *is.*

Once again, the largest minimal or bound-consistent network of a CCRN exists and is unique by monotonicity of constraint relaxations. In the following, we use $minCCRN(X, I, C)$ and $bcCCRN(X, I, C)$ to denote these networks, i.e.,

$$\begin{aligned} \textsc{min}\text{CCRN}(X, I, R) &= \max\{I_m \subseteq I \mid (X, I_m, R) \text{ is minimal}\}, \\ \textsc{bc}\text{CCRN}(X, I, R) &= \max\{I_m \subseteq I \mid (X, I_m, R) \text{ is bound-consistent}\}. \end{aligned}$$

The following property establishes the soundness of bound tightenings in CCRNs.

Proposition 1. *Let* (X, I, C) *be a CCN and let* (X, I, R) *be a CCRN such that* $R = \{r \mid c \in C \wedge r \text{ is a relaxation of } c\}$. *Then,*

$$\textsc{min}\text{CCN}(X, I, C) \subseteq \textsc{min}\text{CCRN}(X, I, R),$$

$$\textsc{bc}\text{CCN}(X, I, C) \subseteq \textsc{bc}\text{CCRN}(X, I, R).$$

[1] Note that some of the convex relaxations used in the QC relaxation are only valid within some bounds. This is easily captured by assuming that the constraint itself imposes these bounds.

```
minCCRN(X, I, R)
    I^n := I;
    repeat
        I^o := I^n;
        I^n := minCCN(X, I^o, R(I^o));
    until I^o = I^n
    return I^n;
```

Fig. 2. Computing the Minimal Continuous Constraint Relaxation Networks

```
bcCCN(X, I, C)
    I^n := I;
    repeat
        I^o := I^n;
        for all c ∈ C
            I^n_c := minCCN(X, I^o, {c});
        I^n := ⋂_{c∈C} I^n_c;
    until I^o = I^n
    return I^n;
```

Fig. 3. Computing the Largest Bound-Consistent Constraint Network.

The minimal and bound-consistent relaxation networks can be computed by a simple fixpoint algorithm that iterates the consistency algorithm over the increasingly tighter relaxation networks. Figure 2 depicts the algorithm for computing a minimal network. The algorithm is similar for bound consistency. Observe that the bound-consistency algorithm has a fixpoint algorithm embedded inside the top-level fixpoint.

4.1 Relation to Concepts in Global Optimization

The idea of bounds propagation for global optimization goes as far back as [6]: It was subsequently implemented in the Numerica system which also performs bound propagation on a linearization of the nonlinear constraints [37,38]. The notion of using bound reductions for improving convex relaxations of non-convex programs was first widely recognized in the Branch-and-Reduce (BnR) algorithm [34]. BnR is a natural extension of Branch-and-Bound over continuous domains, which includes additional steps to reduce the domains of the variables at each search node. This line of work has developed into two core bound reduction ideas: (1) Feasibility-Based Range Reduction (FBRR), which is concerned with pruning techniques based on feasibility information and (2) Optimality Based Range Reduction (OBRR), which develops bound reductions based on Lagrangian-duality arguments [35]. A variety of methods have been developed for FBRR and OBRR with various pruning strength and computational time tradeoffs [34,25,5]. However, all these methods are non-global bound reduction techniques and may be iterated until a desired level of consistency is achieved.

CCRNs and the associated consistency notions (minCCRN, bcCCRN) developed herein are examples of FBRR methods. The idea of computing minCCRN is discussed informally in [4] for the special case where the relaxation is a system of linear or convex equations (note that the algorithm in Figure 2 applies for any kind of CSP). It is often noted in the FBRR literature that just one iteration of the minCCRN is too costly to compute [4,35], let alone the full fixpoint. The preferred approach is to perform some bound propagation (not always to the fixpoint) on linear relaxations of the non-convex problem [35,25,5]. In fact, specialized algorithms have been proposed for computing bound consistency on purely linear systems for this purpose [4]. The linear bound-consistency computations discussed in [4,5] are weaker forms of the BCCCRN notion considered here since it does not explicitly mention re-linearizing the relaxation after bound propagation is complete and re-computing bounds consistency. It is important to note that the algorithm in Figure 2 seamlessly hybridizes the FBRR ideas from global optimization to CP systems, which include arbitrary global constraints that are outside the scope of purely mathematical programs. This advantage is utilized in the next section.

5 Constraint Relaxation Networks for Power Flows

This section discusses how to compute the largest minimal and bound-consistent networks for the relaxation model (X, I, R) defined by Model 2. Observe first that the convex relaxations used in Model 2 are all monotonic.

Proposition 2. *The convex relaxations T-CONV, M-CONV, C-CONV, and S-CONV are monotonic.*

Minimal Network The largest minimal network is computed by Algorithm QC-N which applies the fixpoint algorithm MINCCRN shown in Figure 2 to Model 2. The underlying MINCCN networks are computed by optimizing each variable independently, i.e.,

$$I_x^n := [\min_{\sigma:R(\sigma)} \sigma(x), \max_{\sigma:R(\sigma)} \sigma(x)];$$

Observe that this computation is inherently parallel, since all the optimizations are independent.

Bound-Consistent Network The largest bound-consistent network is computed by Algorithm QC-B which applies the bound-consistency counterpart of algorithm MINCCRN to Model 2. The BCCCN networks needed in this fixpoint algorithm are computed by the algorithm shown in Figure 3. Algorithm BCCCN computes the intervals I_c^n that are bound-consistent for each constraint $c \in C$ before taking the intersection of these intervals. The process is iterated until a fixpoint is obtained. This algorithm was selected because the bound-consistency computations can be performed in parallel.

Observe also that the QC-B algorithm is applied to a version of Model 2 using a global constraint

$$\text{line\_power\_qc}(p_{ij}, q_{ij}, v_i, \theta_i, p_{ji}, q_{ji}, v_j, \theta_j)$$

that captures constraints (7f)–(7g), (10a)–(10k) for each line $(i, j) \in E$. The use of this global constraint means that QC-B computes a stronger form of bounds consistency than one based purely on Model 2. This stronger level of consistency is necessary to obtain reasonable bound tightenings. Note that all the optimizations in algorithms QC-N and QC-B are convex optimization problems which can be solved in polynomial time.

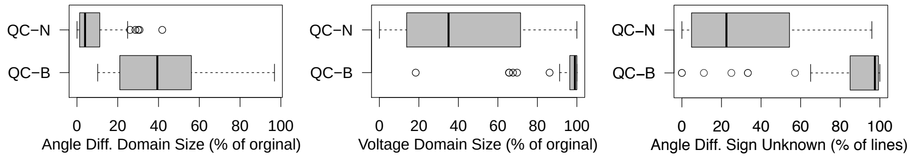

Fig. 4. QC Consistency Algorithms – Quality Analysis.

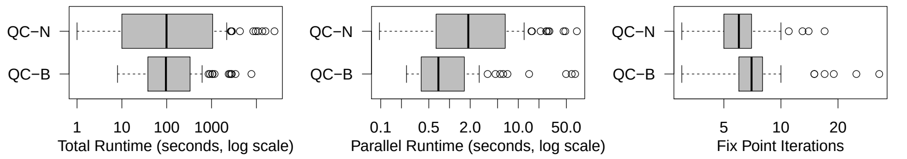

Fig. 5. QC Consistency Algorithms – Runtime Analysis.

6 Strength and Performance of the Bound Tightening

This section evaluates the benefits of QC-N and QC-B on the general feasibility problem in Model 1. Algorithms QC-N and QC-B were implemented in AMPL [15] using IPOPT 3.12 [41] to solve the convex nonlinear programs. The propagation algorithms were executed on Dell PowerEdge R415 servers with Dual 2.8GHz AMD 6-Core Opteron 4184 CPUs and 64GB of memory with a convergence tolerance of $\epsilon = 0.001$. Their performance is evaluated on 57 transmission system test cases from the NESTA 0.3.0 archive [10] ranging from 3 to 300 buses.

Figure 4 summarizes the results of QC-B and QC-N on three key metrics: the phase angle difference domains (θ^{Δ}), the voltage domains (v), and the number of lines where the sign of θ^{Δ} is determined. Each plot summarizes the distribution of 57 values as a standard *box-and-whisker* plot, where the width of the box reflects the first and third quartiles, the black line inside the box is the median, and the whiskers reflect min and max values up to 1.5 IQR with the remaining data points plotted as outliers. In these plots values to the left are preferable. The domain reduction of the QC-N approach is substantial, typically pruning the domain θ^{Δ} by 90% and the domain of v by 30% and determining the sign of θ^{Δ} for about half of the lines. Across all of the metrics, it is clear that QC-N has significant benefits over QC-B.

Figure 5 summarizes the runtime performance of QC-B and QC-N on three key metrics: Total CPU time (T_1), fully parallel CPU wall-clock time (T_∞), and the number of fixpoint iterations. The total runtimes of the QC-B and QC-N algorithms vary widely based on the size of the network under consideration and can range from seconds to hours. Fortunately, regardless of the size of the network, the number of iterations in the fixpoint computation is small (often less than 10). As a result, the parallel runtime of the algorithms scale well with the size of the network and rarely exceeds 1 minute, which is well within the runtime requirements of the majority of network optimization applications.[2]

7 Application to AC Optimal Power Flow

This section assesses the benefits of QC-B and QC-N on the ubiquitous AC Optimal Power Flow problem (AC-OPF) [29,30]. The goal of the AC-OPF is to find the cheapest way to satisfy the loads given the network flow constraints and generator costs functions, which are typically quadratic. If $\boldsymbol{c}_{2i}, \boldsymbol{c}_{1i}, \boldsymbol{c}_{0i}$ are the cost coefficients for generating power at bus $i \in N$, the AC-OPF objective function is given by,

$$\textbf{minimize:} \sum_{i \in N} \boldsymbol{c}_{2i}(p^g_i)^2 + \boldsymbol{c}_{1i}(p^g_i) + \boldsymbol{c}_{0i} \tag{11}$$

The complete non-convex AC-OPF problem is Model 1 with objective (11) and the QC relaxation of this problem is Model 2 with objective (11).

The goal is to compare five AC-OPF relaxations for bounding primal AC-OPF solutions produced by IPOPT, which only guarantees local optimality. The five relaxations under consideration are as follows:

1. QC - as defined in Model 2.
2. QC-B - BCCCRN for Model 2.
3. QC-N - MINCCRN for Model 2.
4. SDP - a state-of-the-art relaxation based on semi-definite programming [26].
5. SDP-N - the SDP relaxation strengthened with bounds from QC-N.

There is no need to consider other existing relaxations as the QC and SDP dominate them [11]. The computational environment and test cases are those of Section 6. SDPT3 4.0 [36] was used to solve the SDP models.

Table 7 presents the detailed performance and runtime results on all 57 test cases. They can be summarized as follows: (1) The optimality gaps of the QC relaxation are significantly reduced by both QC-N and QC-B; (2) QC-N closes the AC-OPF optimality gap to below 1% in 90% of the cases considered and closes 10 open test cases; (3) QC-N almost always outperforms the SDP relaxation in quality with comparable parallel runtimes; (4) For the test cases with significant optimality gaps, QC-N outperforms the SDP relaxation most often, even when the SDP relaxation is strengthened with QC-N bounds (i.e., SDP-N).

Overall, these results clearly establish QC-N is the new state-of-the-art convex relaxation of the AC-OPF. General purpose global optimization solvers (e.g., Couenne 0.4 [3] and SCIP 3.1.1 [1,8]) were also considered for comparison. Preliminary results indicated that these general purpose solvers are much slower than the dedicated power flow relaxations considered here and cannot produce competitive lower bounds on these networks with in 10 hours of computation.

[2] Dedicated high performance computational resources are commonplace in power system operation centers. The T_∞ runtime is realistic in these settings where high-level of reliability is critical.

Test Case	$/h AC	Optimality Gap (%) SDP-N	SDP	QC-N	QC-B	QC	T_∞ Runtime (sec.) AC	SDP-N	SDP	QC-N	QC-B	QC
case3_lmbd	5812	0.1	0.4	**0.1**	1.0	1.2	0.2	6.8	4.7	0.5	0.4	0.1
case4_gs	156	0.0	0.0	**0.0**	0.0	0.0	0.2	7.2	4.8	0.4	0.8	0.1
case5_pjm	17551	5.2	**5.2**	9.3	14.5	14.5	0.1	6.4	5.1	0.9	0.3	0.2
case6_c	23	0.0	0.0	**0.0**	0.3	0.3	0.0	6.9	5.4	1.3	0.4	0.1
case6_ww	3143	0.0	0.0	**0.0**	0.1	0.6	0.3	5.4	5.4	0.8	2.7	0.1
case9_wscc	5296	0.0	0.0	**0.0**	0.0	0.0	0.2	6.2	4.9	1.5	0.7	0.1
case14_ieee	244	0.0	0.0	**0.0**	0.1	0.1	0.1	4.8	5.2	2.0	0.4	0.1
case24_ieee_rts	63352	0.0	0.0	**0.0**	0.0	0.0	0.2	8.5	6.0	3.2	0.5	0.2
case29_edin	29895	0.0	**0.0**	0.0	0.1	0.1	0.4	8.2	7.8	15.8	1.4	1.1
case30_as	803	0.0	0.0	**0.0**	0.1	0.1	0.3	6.9	5.4	2.3	0.5	0.1
case30_fsr	575	0.0	**0.0**	0.1	0.3	0.4	0.2	5.5	6.1	2.2	1.0	0.2
case30_ieee	205	0.0	0.0	**0.0**	5.3	15.4	0.4	7.8	6.3	0.7	0.7	0.3
case39_epri	96505	0.0	0.0	**0.0**	0.0	0.0	0.2	6.5	7.1	2.1	0.6	0.2
case57_ieee	1143	0.0	0.0	**0.0**	0.1	0.1	0.1	11.4	9.1	5.1	0.9	0.4
case73_ieee_rts	189764	0.0	0.0	**0.0**	0.0	0.0	0.5	12.5	8.5	4.7	0.7	0.5
case118_ieee	3720	0.1	**0.1**	0.4	1.0	1.7	0.3	18.2	12.0	21.1	6.0	0.8
case162_ieee_dtc	4237	1.0	1.1	**0.7**	3.8	4.2	0.7	57.6	34.2	25.9	7.0	1.5
case189_edin	849	—	0.1	**0.1**	—	0.2	0.9	12.3	13.3	6.5	59.9	1.6
case300_ieee	16894	0.1	**0.1**	0.1	1.0	1.2	0.9	40.8	25.5	48.2	14.4	2.4
case3_lmbd__api	367	0.0	1.3	**0.0**	0.5	1.8	0.2	4.0	4.0	0.5	1.5	0.1
case4_gs__api	767	0.0	0.0	**0.0**	0.2	0.7	0.9	6.9	3.9	0.8	0.3	0.1
case5_pjm__api	2994	0.0	0.0	**0.0**	0.4	0.4	0.0	6.9	7.0	0.2	0.4	0.1
case6_c__api	807	0.0	0.0	**0.0**	0.5	0.5	0.6	5.3	5.4	0.3	0.4	0.1
case6_ww__api	273	—	0.0	**0.0**	2.1	13.1	0.2	4.5	15.0	0.4	0.4	0.1
case9_wscc__api	656	0.0	0.0	**0.0**	0.0	0.0	0.4	5.4	6.4	0.8	0.9	0.1
case14_ieee__api	323	0.0	**0.0**	0.2	1.3	1.3	0.1	6.4	4.7	0.5	0.4	0.1
case24_ieee_rts__api	6421	0.7	1.4	**0.3**	3.3	13.8	0.2	8.6	7.2	1.4	1.6	0.2
case29_edin__api	295764	—	—	**0.1**	0.4	0.4	0.3	12.6	7.8	28.4	1.1	3.2
case30_as__api	571	0.0	0.0	**0.0**	2.4	4.8	0.4	7.6	6.0	3.7	0.8	0.2
case30_fsr__api	372	3.6	11.1	**2.7**	42.8	46.0	0.2	7.9	6.7	1.4	0.4	0.2
case30_ieee__api	411	0.0	0.0	**0.0**	0.9	1.0	0.3	8.9	6.5	1.0	0.5	0.2
case39_epri__api	7466	0.0	0.0	**0.0**	0.8	3.0	0.1	9.2	6.5	4.9	1.9	0.2
case57_ieee__api	1430	0.0	0.1	**0.0**	0.2	0.2	0.4	8.8	8.1	3.2	0.6	0.4
case73_ieee_rts__api	20123	0.9	4.3	**0.1**	3.6	12.0	0.6	15.4	9.5	11.4	2.0	0.6
case118_ieee__api	10258	16.7	31.5	**11.8**	38.9	44.0	0.6	14.2	14.6	11.3	4.7	0.8
case162_ieee_dtc__api	6095	0.6	1.0	**0.1**	1.4	1.5	0.4	51.9	32.8	25.5	2.1	1.5
case189_edin__api	1971	—	0.1	**0.0**	—	5.6	0.3	14.3	13.5	8.3	67.1	1.1
case300_ieee__api	22825	0.0	**0.0**	0.2	0.6	0.8	0.9	47.5	28.5	71.1	3.6	2.6
case3_lmbd__sad	5992	0.1	2.1	**0.0**	0.2	1.2	0.1	5.1	4.2	0.2	0.9	0.1
case4_gs__sad	324	0.0	0.0	**0.0**	0.5	0.8	0.1	4.4	3.9	0.1	1.3	0.1
case5_pjm__sad	26423	0.0	0.0	**0.0**	0.7	1.1	0.1	5.7	5.3	0.2	0.4	0.1
case6_c__sad	24	0.0	0.0	**0.0**	0.4	0.4	0.1	6.7	4.6	0.2	0.3	0.1
case6_ww__sad	3149	0.0	0.0	**0.0**	0.1	0.3	0.1	5.9	5.4	0.2	0.2	0.1
case9_wscc__sad	5590	0.0	0.0	**0.0**	0.2	0.4	0.3	5.5	4.4	0.1	0.5	0.1
case14_ieee__sad	244	0.0	0.0	**0.0**	0.1	0.1	0.1	7.5	4.6	0.5	0.3	0.1
case24_ieee_rts__sad	79804	1.4	6.1	**0.1**	3.4	3.9	0.3	9.3	5.7	0.6	0.4	0.3
case29_edin__sad	46933	5.8	28.4	**0.9**	20.0	20.6	0.5	7.1	8.5	15.5	0.3	1.6
case30_as__sad	914	0.1	0.5	**0.0**	2.9	3.1	0.1	6.4	6.8	2.3	0.3	0.2
case30_fsr__sad	577	0.1	0.1	**0.1**	0.5	0.6	0.1	6.2	6.8	1.9	0.3	0.2
case30_ieee__sad	205	0.0	0.0	**0.0**	2.0	4.0	0.3	7.0	6.0	0.6	0.6	0.1
case39_epri__sad	97219	0.0	0.1	**0.0**	0.0	0.0	0.1	7.1	6.0	1.0	0.9	0.2
case57_ieee__sad	1143	0.0	0.0	**0.0**	0.1	0.1	0.4	8.8	7.6	1.9	0.8	0.3
case73_ieee_rts__sad	235241	2.4	4.1	**0.1**	3.1	3.5	0.3	9.7	8.4	3.6	0.6	0.8
case118_ieee__sad	4323	4.0	7.6	**1.4**	7.6	8.3	0.4	15.4	13.8	5.9	0.6	1.0
case162_ieee_dtc__sad	4368	1.7	3.6	**0.4**	5.9	6.9	0.9	46.8	37.7	27.3	2.1	1.4
case189_edin__sad	914	—	1.2	**0.5**	—	2.2	0.6	11.4	17.4	12.2	49.6	1.1
case300_ieee__sad	16912	0.1	**0.1**	0.1	0.8	1.2	0.9	25.2	30.8	45.6	5.5	2.4

Table 1. Quality and Runtime Results of Convex Relaxations on the AC-OPF Problem (**bold** - best in row (runtime used to break ties in quality), — - solving error)

Model 3 The AC-PF Program with Load Uncertainty (AC-PF-U)

variables: Variables of Model 1

$p^d_i \in (\boldsymbol{p}^{dl}_i, \boldsymbol{p}^{du}_i)\ \ \forall i \in N$ - active power load interval

$q^d_i \in (\boldsymbol{q}^{dl}_i, \boldsymbol{q}^{du}_i)\ \ \forall i \in N$ - reactive power load interval

$p^g_i \in (0, \boldsymbol{p}^{du}_i)\ \ \forall i \in N$ - active power generation interval

subject to: (7a), (7d)–(7g)

$$p^g_i - p^d_i = \sum_{(i,j)\in E\cup E^R} p_{ij}\ \ \forall i \in N \tag{12a}$$

$$q^g_i - q^d_i = \sum_{(i,j)\in E\cup E^R} q_{ij}\ \ \forall i \in N \tag{12b}$$

8 Propagation with Load Uncertainty

Loads in power systems are highly predictable. In transmission systems, it is commonplace for minute-by-minute load forecasts to be within 5% of the true values [9]. This high degree of predictability can be utilized by the bound tightening algorithms proposed here. Indeed, if the feasible set of Model 1 is increased to include a range of possible load values, determined by the forecast, then the algorithms compute a description of all possible future power flows. This section studies the power of bound propagation in this setting.

Model 3 presents an extension of Model 1 to incorporate load uncertainty. New decision variables for the possible load values are introduced (i.e., p^d, q^d) and their bounds come from the extreme values of the load forecasting model. The lower bounds on active power generation (p^g) are also increased to include 0, as generators may become inactive at some point in the future (e.g., due to scheduled maintenance or market operations). Constraints (12a)–(12b) incorporate the load variables into KCL. The other constraints remain the same as in Model 1. Because only the KCL constraints are modified in this formulation, the QC relaxation of Model 3 (QC-U) is similar to Model 1, as described in Section 3. For the experimental evaluation, the 57 deterministic test cases were extended into uncertain load cases by adopting a forecast model of ±5% of the deterministic load value.

Figure 6 compares the quality of minCCRN on the QC-U model (QC-U-N) to minCCRN in the deterministic case (QC-N) in order to illustrate the pruning loss due to uncertainty. The results indicate that, even when load uncertainty is incorporated, minCCRN still prunes the variable domains significantly, typically reducing the voltage angle domains by 80% and the voltage magnitude domains by 10%, and determining the sign of θ^Δ for about 30% of the lines. The domain reduction on θ^Δ in QC-U-N is particularly significant.

Figure 7 considers the AC-OPF and summarizes the optimality gaps produced under load certainty and uncertainty. QC-U-N produces significant im-

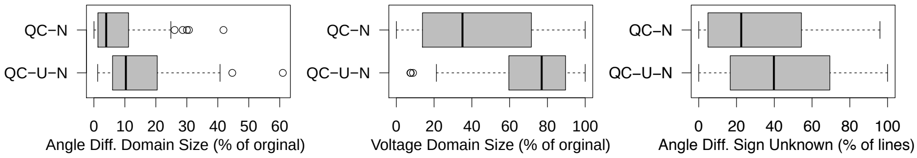

Fig. 6. QC Consistency Algorithms with Load Uncertainty – Quality Analysis.

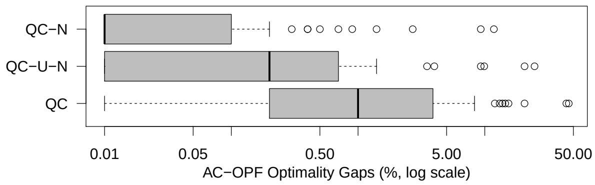

Fig. 7. Comparison of AC-OPF Bound Improvements of QC Variants.

provement in optimality gaps, moving from $< 5\%$ (QC) to less than $< 1\%$. Obviously, load certainty (QC-N) closes the remaining 1%.

9 Conclusion

This paper studied how bound tightening can improve convex relaxations by adapting traditional constraint-programming concepts (e.g., minimal network and bound consistency) to a relaxation framework. It showed that, on power flow applications, bound tightening over the QC relaxation can dramatically reduce variable domains. Moreover, on the ubiquitous AC-OPF problem, the QC relaxation, enhanced by bound tightening, almost always outperforms the state-of-the-art SDP relaxation on the optimal power flow problem. The paper also showed that bound tightening yields significant benefits under load uncertainty, demonstrating a breadth of applicability. These results highlight the significant potential synergies between constraint programming and convex optimization for complex engineering problems.

References

1. Achterberg, T.: Scip: solving constraint integer programs. Mathematical Programming Computation 1(1), 1–41 (2009), http://dx.doi.org/10.1007/s12532-008-0001-1
2. Bai, X., Wei, H., Fujisawa, K., Wang, Y.: Semidefinite programming for optimal power flow problems. International Journal of Electrical Power & Energy Systems 30(67), 383 – 392 (2008)
3. Belotti, P.: Couenne: User manual. Published online at https://projects.coin-or.org/Couenne/ (2009), accessed: 10/04/2015
4. Belotti, P., Cafieri, S., Lee, J., Liberti, L.: On feasibility based bounds tightening. Published online at http://www.optimization-online.org/DB_HTML/2012/01/3325.html (2012)

5. Belotti, P., Lee, J., Liberti, L., Margot, F., Wachter, A.: Branching and bounds tightening techniques for non-convex minlp. Optimization Methods Software 24(4-5), 597–634 (Aug 2009)
6. Benhamou, F., McAllester, D., Van Hentenryck, P.: Clp (intervals) revisited. Tech. rep., Brown University, Providence, RI, USA (1994)
7. Bent, R., Coffrin, C., Gumucio, R., Van Hentenryck, P.: Transmission network expansion planning: Bridging the gap between ac heuristics and dc approximations. Proceedings of the 18th Power Systems Computation Conference (PSCC'14), Wroclaw, Poland (2014)
8. Berthold, T., Heinz, S., Vigerske, S.: Extending a cip framework to solve miqcps. In: Lee, J., Leyffer, S. (eds.) Mixed Integer Nonlinear Programming, The IMA Volumes in Mathematics and its Applications, vol. 154, pp. 427–444. Springer New York (2012)
9. Chen, Y., Luh, P., Guan, C., Zhao, Y., Michel, L., Coolbeth, M., Friedland, P., Rourke, S.: Short-term load forecasting: Similar day-based wavelet neural networks. IEEE Transactions on Power Systems 25(1), 322–330 (February 2010)
10. Coffrin, C., Gordon, D., Scott, P.: NESTA, The NICTA Energy System Test Case Archive. CoRR abs/1411.0359 (2014), http://arxiv.org/abs/1411.0359
11. Coffrin, C., Hijazi, H., Van Hentenryck, P.: The QC Relaxation: Theoretical and Computational Results on Optimal Power Flow. CoRR abs/1502.07847 (2015), http://arxiv.org/abs/1502.07847
12. Coffrin, C., Van Hentenryck, P.: Transmission system restoration: Co-optimization of repairs, load pickups, and generation dispatch. International Journal of Electrical Power & Energy Systems (forthcoming) (2015)
13. Farivar, M., Clarke, C., Low, S., Chandy, K.: Inverter var control for distribution systems with renewables. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm). pp. 457–462 (Oct 2011)
14. Fisher, E., O'Neill, R., Ferris, M.: Optimal transmission switching. IEEE Transactions on Power Systems 23(3), 1346–1355 (2008)
15. Fourer, R., Gay, D.M., Kernighan, B.: AMPL: A Mathematical Programming Language. In: Wallace, S.W. (ed.) Algorithms and Model Formulations in Mathematical Programming, pp. 150–151. Springer-Verlag New York, Inc., New York, NY, USA (1989)
16. Fu, Y., Shahidehpour, M., Li, Z.: Security-constrained unit commitment with ac constraints*. IEEE Transactions on Power Systems 20(3), 1538–1550 (Aug 2005)
17. Hedman, K., Ferris, M., O'Neill, R., Fisher, E., Oren, S.: Co-optimization of generation unit commitment and transmission switching with n-1 reliability. In: Power and Energy Society General Meeting, 2010 IEEE. pp. 1–1 (July 2010)
18. Hijazi, H., Coffrin, C., Van Hentenryck, P.: Convex quadratic relaxations of mixed-integer nonlinear programs in power systems. Published online at http://www.optimization-online.org/DB_HTML/2013/09/4057.html (2013)
19. Hijazi, H., Thiebaux, S.: Optimal ac distribution systems reconfiguration. Proceedings of the 18th Power Systems Computation Conference (PSCC'14), Wroclaw, Poland (2014)
20. Jabr, R.: Radial distribution load flow using conic programming. IEEE Transactions on Power Systems 21(3), 1458–1459 (Aug 2006)
21. Jabr, R.: Optimization of ac transmission system planning. IEEE Transactions on Power Systems 28(3), 2779–2787 (Aug 2013)
22. Kundur, P.: Power System Stability and Control. McGraw-Hill Professional (1994)
23. Lavaei, J., Low, S.: Zero duality gap in optimal power flow problem. IEEE Transactions on Power Systems 27(1), 92 –107 (feb 2012)

24. Lehmann, K., Grastien, A., Van Hentenryck, P.: AC-Feasibility on Tree Networks is NP-Hard. IEEE Transactions on Power Systems (2015 (to appear))
25. Liberti, L.: Writing global optimization software. In: Liberti, L., Maculan, N. (eds.) Global Optimization, Nonconvex Optimization and Its Applications, vol. 84, pp. 211–262. Springer US (2006), http://dx.doi.org/10.1007/0-387-30528-9_8
26. Madani, R., Ashraphijuo, M., Lavaei, J.: Promises of conic relaxation for contingency-constrained optimal power flow problem. Published online at http://www.ee.columbia.edu/~lavaei/SCOPF_2014.pdf (2014), accessed: 22/02/2015
27. McCormick, G.: Computability of global solutions to factorable nonconvex programs: Part i convex underestimating problems. Mathematical Programming 10, 146–175 (1976)
28. Miller, J.: Power system optimization smart grid, demand dispatch, and microgrids. Published online at http://www.netl.doe.gov/smartgrid/referenceshelf/presentations/SE%20Dist%20Apparatus%20School_Final_082911_rev2.pdf (Sept 2011), accessed: 22/04/2012
29. Momoh, J., Adapa, R., El-Hawary, M.: A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches. IEEE Transactions on Power Systems 14(1), 96 –104 (feb 1999)
30. Momoh, J., El-Hawary, M., Adapa, R.: A review of selected optimal power flow literature to 1993. ii. newton, linear programming and interior point methods. IEEE Transactions on Power Systems 14(1), 105 –111 (feb 1999)
31. Montanari, U.: Networks of Constraints : Fundamental Properties and Applications to Picture Processing. Information Science 7(2), 95–132 (1974)
32. Ott, A.: Unit commitment in the pjm day-ahead and real-time markets. Published online at http://www.ferc.gov/eventcalendar/Files/20100601131610-Ott,%20PJM.pdf (June 2010), accessed: 22/04/2012
33. Purchala, K., Meeus, L., Van Dommelen, D., Belmans, R.: Usefulness of DC power flow for active power flow analysis. Power Engineering Society General Meeting pp. 454–459 (2005)
34. Ryoo, H., Sahinidis, N.: A branch-and-reduce approach to global optimization. Journal of Global Optimization 8(2), 107–138 (1996)
35. Sahinidis, N.: Global optimization and constraint satisfaction: The branch-and-reduce approach. In: Bliek, C., Jermann, C., Neumaier, A. (eds.) Global Optimization and Constraint Satisfaction, Lecture Notes in Computer Science, vol. 2861, pp. 1–16. Springer Berlin Heidelberg (2003)
36. Toh, K.C., Todd, M., Ttnc, R.H.: Sdpt3 – a matlab software package for semidefinite programming. Optimization Methods and Software 11, 545–581 (1999)
37. Van Hentenryck, P., McAllister, D., Kapur, D.: Solving Polynomial Systems Using a Branch and Prune Approach. SIAM Journal on Numerical Analysis 34(2) (1997)
38. Van Hentenryck, P., Michel, L., Deville, Y.: Numerica: a Modeling Language for Global Optimization. The MIT Press, Cambridge, Mass. (1997)
39. Van Hentenryck, P., Saraswat, V., Deville, Y.: The Design, Implementation, and Evaluation of the Constraint Language cc(FD). In: Constraint Programming: Basics and Trends. Springer Verlag (1995)
40. Verma, A.: Power grid security analysis: An optimization approach. Ph.D. thesis, Columbia University (2009)
41. Wächter, A., Biegler, L.T.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Mathematical Programming 106(1), 25–57 (2006)