IET Generation, Transmission & Distribution

Research Article

Recover feasible solutions for SOCP relaxation of optimal power flow problems in mesh networks

ISSN 1751-8687
Received on 18th December 2017
Revised 16th November 2018
Accepted on 18th January 2019
E-First on 21st March 2019
doi: 10.1049/iet-gtd.2018.6015
www.ietdl.org

Zhuang Tian[1], *Wenchuan Wu*[1] ✉

[1] *State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing, People's Republic of China*

✉ *E-mail: wuwench@tsinghua.edu.cn*

Abstract: The AC optimal power flow (OPF) problem is essential for the schedule and operation of power systems. Convex relaxation methods have been studied and used extensively to obtain an optimal solution to the OPF problem. When the exactness of convex relaxations is not guaranteed, it is important to recover a feasible solution for the convex relaxation methods. This paper presents an alternative convex approach (ACP) approach that can efficiently recover a feasible solution from the result of second-order cone programming (SOCP) relaxed OPF in mesh networks. The OPF problem is first formulated as a difference-of-convex programming (DCP) problem, then efficiently solved by a penalty convex concave procedure (CCP). By using the solution of a tightened SOCP OPF as an initial point, the proposed algorithm is able to find a global or near-global optimal solution to the AC OPF problem. Numerical tests show that the proposed method outperforms those semi-definite programming (SDP) and quadratically constrained quadratic programming (QCQP)–based algorithms.

Nomenclature

Indices and sets

Φ_b	sets of all buses
Φ_l	sets of all lines
$K(i)$	sets of buses connected to bus i

Parameters

G_{ij}	conductance of branch ij
B_{ij}	susceptance of branch ij
$G_{sh,i}$	shunt conductance at bus i
$B_{sh,i}$	shunt susceptance at bus i
p_i^d	active power demand at bus i
q_i^d	reactive power demand at bus i
p_i^l, p_i^u	active power capacity of generator at bus i
q_i^l, q_i^u	reactive power capacity of generator at to bus i
θ^u	maximum phase angle difference of each branch
S^u	maximum apparent power of each branch
V^l, V^u	voltage magnitude limit of each bus
s^l, s^u	range of s_{ij}, $s^l = -\sin\theta^u$, $s^u = \sin\theta^u$
c^l, c^u	range of c_{ij}, $c^l = \cos\theta^u$, $c^u = 1$
K^l, K^u	range of K_{ij}, $K^l = (V^l)^2c^l$, $K^u = (V^u)^2$
L^l, L^u	range of L_{ij}, $L^l = (V^u)^2s^l$, $L^u = (V^u)^2s^u$
τ	penalty parameters in convex concave procedure
δ	stopping criterion in convex concave procedure

Variables

V_i	voltage magnitude of bus i
θ_i	phase angle of bus i
θ_{ij}	phase angle difference of branch ij
p_{ij}	active power flow from bus i to bus j
q_{ij}	reactive power flow from bus i to bus j
p_i^g	active power provided by generator at bus i
q_i^g	reactive power provided by generator at bus i
U_i	square of V_i
K_{ij}	denotes $V_iV_j\cos\theta_{ij}$
L_{ij}	denotes $V_iV_j\cos\theta_{ij}$
s_{ij}	denotes $\sin\theta_{ij}$
c_{ij}	denotes $\cos\theta_{ij}$
ε	slack variables in convex concave procedure

1 Introduction

The AC optimal power flow (OPF) problem is essential for power systems to determine the operation point that best minimises generation cost, power losses, voltage fluctuations, and other crucial outcomes. It is a typical non-convex and NP-hard problem, for which the non-convexity mainly lies in the power flow equations. Traditional methods to solve OPF problems in transmission systems include linear approximations, the Newton–Raphson method and some heuristic algorithms, which either lack feasibility or cannot ensure optimality. With the increasing penetration of renewable generations, the OPF problem for power systems has drawn much attention in recent years. Decentralised optimisation and robust optimisation problems also require convex formulation of power flow equations so that the problem can be solved efficiently. The convex relaxation of OPF problems was first proposed in [1, 2] and has become an important research topic in the past 5 years.

Convex relaxation methods mainly include semi-definite programming (SDP) relaxation [1] and second-order cone programming (SOCP) relaxation [2]. These methods can find a lower bound of the original minimisation problem, and in certain circumstances, a feasible solution of the original problem can be recovered from the solution of convex relaxation methods. For SDP relaxation, if the rank one condition is satisfied, then the zero-duality gap can be guaranteed, hence a feasible solution is sure to be recovered [3]. For SOCP relaxation, a feasible solution can be recovered when the quadratic and arctangent equalities both hold [4, 5]. Under such circumstances, we say the convex relaxation is exact, and its solution is the global optimum of the original OPF problem. Many researchers have devoted efforts towards finding sufficient conditions for ensuring the exactness of, or strengthening, the convex relaxations.

In [6], a sufficient zero-duality-gap condition for SDP was found in resistive networks with active loads if over-satisfaction of the loads was permitted. In [7, 8], sufficient conditions for SDP in radial and mesh networks were discussed. In [9], the exactness of SDP for mesh networks was related to the modelling of the capacity of a power line. For the situation where OPF problem does not satisfy the sufficient conditions, such as load over-satisfaction, [10] introduced inequalities, and variable bounds to strengthen the convex relaxation.

In [11], the sufficient condition for SOCP in radial networks was proposed. If the objective function of the OPF problem is non-increasing in load, and there are no upper limits on load, then the solution of SOCP is exact for radial networks. In [12], a new sufficient condition was proposed such that, if there are no simultaneous active and reactive reverse power flows on non-leaf lines, then the SOCP relaxation is exact. In [13], the performance of SOCP in mesh networks was further studied. In [14, 15], a cycle-based formulation of angle constraints was proposed to enhance SOCP relaxation. By exploring the fact that angle differences sum up to zero over each cycle, the angle constraints were transformed into bilinear constraints. However, there has not been a method that guarantees the exactness of SOCP relaxation in mesh networks, because the conic relaxation and the angle relaxation must both be exact to ensure the feasibility of the SOCP solution in mesh networks, but the angle constraints are difficult to deal with due to the trigonometric functions.

While the above literatures discussed the exactness of convex relaxation methods, it is still an important issue that how to recover a feasible solution of the original OPF problem when the exactness of convex relaxation is not guaranteed. The motivation for feasibility recovery is to make convex relaxation more practical in sophisticated problems based on OPF, such as distributed optimisation or robust optimisation problems in power systems:

(i) In distributed optimisation problems [16], the convergence of distributed algorithms, such as ADMM, can only be guaranteed for convex problems [17]. In such problems, the convex-relaxed power flow equations are employed. So, the solution of distributed OPF must be recovered to a feasible solution to make the strategies practical.
(ii) In multi-stage robust optimisation problems, the sub-problem is usually reformulated using strong duality [18] or KKT conditions [19], which implies that the sub-problem must be convex. Since power flow constraints are usually involved in the sub-problem, these non-convex constraints should be convex relaxed. In this situation, a feasible solution should be recovered to ensure the solution of robust optimisation physically meaningful.

Since feasibility recovery problem is solved by distributed algorithms in distributed optimisation problem or treated as the sub-problem in multi-stage robust optimisation problem, it should also be formulated as a convex problem. Some studies have focused on recovering feasible solutions through convex optimisation using SDP or QCQP-based methods.

In [9], a penalised SDP method is proposed, the total amount of reactive power was added to the objective to force the rank to become one. In [20], the matrix rank is approximated by a continuous function and penalised in the objective function, then a majorisation-minimisation method is applied to solve the penalised SDP problem iteratively. In [21], moment relaxations were proposed for the OPF problem as a generalisation of SDP relaxation, and had the potential to find a global optimal solution using polynomial optimisation theory. Moment relaxations significantly increase the matrix size of semi-definite constraints, which is much more computational inefficient than SDP. In [22, 23], instead of forcing matrix rank to be one, they employed the quadratically constrained quadratic programming (QCQP) formulation of OPF problems, and applied convex concave procedure (CCP) to deal with the indefinite coefficient matrix. In [24], when the SOCP relaxation is inexact, the OPF problem in radial networks was first formulated as a difference-of-convex programming (DCP) problem, then solved as a sequence of convexified penalisation problems.

However, there is no method yet available to recover a feasible solution through convex optimisation for SOCP relaxation in mesh networks. This paper applies the CCP to the OPF problem in mesh networks and recovers a feasible and local optimal solution for SOCP relaxation. CCP is a powerful heuristic method for finding a local optimum of DCP problems [25], which was first introduced in [26, 27]. It iteratively linearises the concave parts of all constraints, thus solving a convex approximation of the DCP problem. In [28], penalty CCP was proposed to negate the need for an initial feasible point in the iteration. Penalty CCP usually benefits from a warm-start point, which makes good use of the solution solved by SOCP. The main contributions of this paper include:

(i) An alternative convex optimisation (ACP) algorithm is proposed that can efficiently recover a feasible solution from the result of SOCP relaxed OPF problem in mesh networks. The ACP algorithm first formulates OPF problem as a DCP problem, then solves the DCP problem by penalty CCP iteratively.
(ii) The convergence of ACP is proved. After ACP converges, if the slack variables all turn out to be zero, then the solution is guaranteed to be a KKT point of the original OPF problem. It is shown that ACP successfully converges to a KKT point of the original OPF problem in all the test cases.
(iii) Numerical tests are conducted on several benchmark systems using ACP and compared with other methods aimed to recover feasible solutions for SDP relaxation. It is shown that the proposed algorithm can find a global or near-global optimal solution within a few iterations, but the other recovery methods may only find worse results. Its computation speed is comparable to SOCP, which is far beyond SDP or QCQP-based recover methods.

The remainder of this paper is organised as follows. Section 2 describes the original non-convex model of OPF in mesh networks and a tightened SOCP relaxation, and Section 3 details the DCP formulation of the OPF problem and the ACP algorithm for feasible solution recovery. Section 4 outlines the test results of the algorithm using several IEEE test systems, and Section 5 concludes the paper.

2 OPF problem and convex relaxation

2.1 Original OPF problem

The OPF problem usually consists of convex functions of generator output, denoted by $C_i(p_i^g)$. This is described as:

$$\text{(Model 1) min } \sum C_i(p_i^g) \tag{1}$$

Subject to
(i) Branch power flow constraints

$$p_{ij} = G_{ij}V_i^2 - G_{ij}V_iV_j\cos\theta_{ij} - B_{ij}V_iV_j\sin\theta_{ij}, \quad \forall ij \in \Phi_l \tag{2}$$

$$q_{ij} = -B_{ij}V_i^2 + B_{ij}V_iV_j\cos\theta_{ij} - G_{ij}V_iV_j\sin\theta_{ij}, \quad \forall ij \in \Phi_l \tag{3}$$

$$\theta_{ij} = \theta_i - \theta_j, \quad \forall ij \in \Phi_l \tag{4}$$

(ii) Active and reactive power balance constraints for buses

$$p_i^g - p_i^d = G_{sh,i}V_i^2 + \sum_{j \in K(i)} p_{ij}, \quad \forall i \in \Phi_b \tag{5}$$

$$q_i^g - q_i^d = -B_{sh,i}V_i^2 + \sum_{j \in K(i)} q_{ij}, \quad \forall i \in \Phi_b \tag{6}$$

(iii) Generator operation constraints

$$p_i^l \le p_i^g \le p_i^u, \quad \forall i \in \Phi_b \tag{7}$$

$$q_i^l \le q_i^g \le q_i^u, \quad \forall i \in \Phi_b \tag{8}$$

(iv) Phase-angle difference limits

$$-\theta^u \le \theta_{ij} \le \theta^u, \quad \forall ij \in \Phi_l \tag{9}$$

(v) Branch thermal limits

$$p_{ij}^2 + q_{ij}^2 \le (S^u)^2, \quad \forall ij \in \Phi_l \tag{10}$$

(vi) Bus voltage limits

$$V^l \le V_i \le V^u, \quad \forall i \in \Phi_b \tag{11}$$

The original formulation of OPF problem is non-convex and the non-convexity comes from branch power flow constraints (2) and (3). The challenge of non-convexity in realistic power systems OPF also comes from transformer taps, capacitor etc., which has been discussed in [29, 30]. So, here, we mainly focus on the non-convex power flow constraints.

By defining new variables $K_{ij} = V_iV_j\cos\theta_{ij}$, $L_{ij} = V_iV_j\sin\theta_{ij}$ and $U_i = V_i^2$, constraints (2) and (3) can be transformed into an alternative form:

$$p_i^g - p_i^d = G_{sh,i}U_i + \sum_{j \in K(i)} p_{ij}, \quad \forall i \in \Phi_b \tag{12}$$

$$q_i^g - q_i^d = -B_{sh,i}U_i + \sum_{j \in K(i)} q_{ij}, \quad \forall i \in \Phi_b \tag{13}$$

$$p_{ij} = G_{ij}U_i - G_{ij}K_{ij} - B_{ij}L_{ij}, \quad \forall ij \in \Phi_l \tag{14}$$

$$q_{ij} = -B_{ij}U_i + B_{ij}K_{ij} - G_{ij}L_{ij}, \quad \forall ij \in \Phi_l \tag{15}$$

$$K_{ij}^2 + L_{ij}^2 = U_iU_j, \quad \forall ij \in \Phi_l \tag{16}$$

$$\theta_{ij} = \arctan(L_{ij}/K_{ij}), \quad \forall ij \in \Phi_l \tag{17}$$

For the OPF of a radial network, constraints (4) and (17) are not necessary because the optimal solution K_{ij} and L_{ij} will always recover a set of θ_{ij} and θ_{ij} that satisfy these two constraints. However, for the OPF of a meshed network, constraints (4) and (15) are necessary to ensure that θ_{ij} sums to zero over all cycles [13].

Constraint (17) is equivalent to:

$$\sin\theta_{ij}K_{ij} = \cos\theta_{ij}L_{ij}, \quad \forall ij \in \Phi_l \tag{18}$$

By introducing new variables s_{ij}, c_{ij}, (18) is equivalent to:

$$s_{ij} = \sin\theta_{ij}, \quad \forall ij \in \Phi_l \tag{19}$$

$$c_{ij} = \cos\theta_{ij}, \quad \forall ij \in \Phi_l \tag{20}$$

$$s_{ij}^2 + c_{ij}^2 = 1, \quad \forall ij \in \Phi_l \tag{21}$$

$$s_{ij}K_{ij} = c_{ij}L_{ij}, \quad \forall ij \in \Phi_l \tag{22}$$

With the above transformation, the OPF problem (Model 1) is equivalent to:

$$\text{(Model 2)} \ \min \sum C_i(p_i^g) \tag{23}$$

subject to (4), (7)–(16) and (19)–(22)

2.2 Tightened SOCP relaxation

The OPF problem (Model 2) is non-convex due to constraints (16) and (19)–(22). Constraint (16) can be relaxed to an inequality [11]:

$$\left\| \begin{matrix} 2K_{ij} \\ 2L_{ij} \\ U_i - U_j \end{matrix} \right\|_2 \le U_i + U_j, \quad \forall ij \in \Phi_l \tag{24}$$

Constraints (19) and (20) can be relaxed by convex envelopes for sine and cosine functions [31]:

$$s_{ij} \le \cos(\theta^u/2)(\theta_{ij} - \theta^u/2) + \sin(\theta^u/2), \quad \forall ij \in \Phi_l \tag{25}$$

$$s_{ij} \ge \cos(\theta^u/2)(\theta_{ij} + \theta^u/2) - \sin(\theta^u/2), \quad \forall ij \in \Phi_l \tag{26}$$

$$c_{ij} \le 1 - (1 - \cos(\theta^u))\theta_{ij}^2/(\theta^u)^2, \quad \forall ij \in \Phi_l \tag{27}$$

$$c_{ij} \ge \cos(\theta^u), \quad \forall ij \in \Phi_l \tag{28}$$

Constraint (21) can be relaxed to:

$$s_{ij}^2 + c_{ij}^2 \le 1, \quad \forall ij \in \Phi_l \tag{29}$$

For constraint (22), by introducing new variables m_{ij} and n_{ij}, it is equivalent to:

$$m_{ij} = s_{ij}K_{ij}, \quad \forall ij \in \Phi_l \tag{30}$$

$$n_{ij} = c_{ij}L_{ij}, \quad \forall ij \in \Phi_l \tag{31}$$

$$m_{ij} = n_{ij}, \quad \forall ij \in \Phi_l \tag{32}$$

Constraints (30) and (31) can be relaxed by McCormick envelopes for bilinear terms [32]:

$$m_{ij} \ge s^lK_{ij} + s_{ij}K^l - s^lK^l, \quad \forall ij \in \Phi_l \tag{33}$$

$$m_{ij} \ge s^uK_{ij} + s_{ij}K^u - s^uK^u, \quad \forall ij \in \Phi_l \tag{34}$$

$$m_{ij} \le s^lK_{ij} + s_{ij}K^u - s^lK^u, \quad \forall ij \in \Phi_l \tag{35}$$

$$m_{ij} \le s^uK_{ij} + s_{ij}K^l - s^uK^l, \quad \forall ij \in \Phi_l \tag{36}$$

$$n_{ij} \ge c^lL_{ij} + c_{ij}L^l - c^lL^l, \quad \forall ij \in \Phi_l \tag{37}$$

$$n_{ij} \ge c^uL_{ij} + c_{ij}L^u - c^uL^u, \quad \forall ij \in \Phi_l \tag{38}$$

$$n_{ij} \le c^lL_{ij} + c_{ij}L^u - c^lL^u, \quad \forall ij \in \Phi_l \tag{39}$$

$$n_{ij} \le c^uL_{ij} + c_{ij}L^l - c^uL^l, \quad \forall ij \in \Phi_l \tag{40}$$

Thus, the convex-relaxed OPF problem is expressed as follows:

$$\text{(Model 3)} \ \min \sum C_i(p_i^g) \tag{41}$$

subject to (4), (7)–(15), (24)–(29) and (32)–(40).

3 Feasible solution recovery algorithm

3.1 Difference-of-convex formulation

The relaxation exactness is barely guaranteed by convex-relaxed Model 3, because equality (19), (20) and (22) are hard to be satisfied by convex envelopes (25)–(28) and McCormick relaxation (33)–(40), so that a feasible solution cannot be recovered from the solution of Model 3 directly. On the other hand, if the bilinear constraints (16), (21), and (22) are satisfied and the trigonometric functions (19) and (20) are well approximated, then the solution will be feasible to the original OPF problem.

In order to satisfy the bilinear constraints (16), (21), and (22), we formulate them as difference-of-convex constraints, which can be solved by DCP algorithms effectively. In such formulation, the equalities are not easy to loosen as in convex relaxation. Take constraint (22) as an example, it can be written in an alternative form:

$$(s_{ij} + K_{ij})^2 - (s_{ij} - K_{ij})^2 = (c_{ij} + L_{ij})^2 - (c_{ij} - L_{ij})^2 \tag{42}$$

which is equivalent to two difference-of-convex constraints:

$$(s_{ij}+K_{ij})^2+(c_{ij}-L_{ij})^2-(s_{ij}-K_{ij})^2-(c_{ij}+L_{ij})^2 \le 0 \quad (43)$$

$$(s_{ij}-K_{ij})^2+(c_{ij}+L_{ij})^2-(s_{ij}+K_{ij})^2-(c_{ij}-L_{ij})^2 \le 0 \quad (44)$$

Considering constraints (16), (21) and (22), we can define the following convex functions:

$$f_{ij,1}(x) = (U_i + U_j)^2 \quad (45)$$

$$f_{ij,2}(x) = 1 \quad (46)$$

$$f_{ij,3}(x) = (s_{ij}+K_{ij})^2 + (c_{ij}-L_{ij})^2 \quad (47)$$

$$g_{ij,1}(x) = (2K_{ij})^2 + (2L_{ij})^2 + (U_i - U_j)^2 \quad (48)$$

$$g_{ij,2}(x) = s_{ij}^2 + c_{ij}^2 \quad (49)$$

$$g_{ij,3}(x) = (s_{ij}-K_{ij})^2 + (c_{ij}+L_{ij})^2 \quad (50)$$

Thus, constraints (16), (21) and (22) can be expressed as difference-of-convex constraints:

$$g_{ij,m}(x) - f_{ij,m}(x) \le 0, \quad \forall ij \in \Phi_l, m = 1,2,3 \quad (51)$$

$$f_{ij,m}(x) - g_{ij,m}(x) \le 0, \quad \forall ij \in \Phi_l, m = 1,2,3 \quad (52)$$

For the precise approximation of trigonometric functions (19) and (20), sixth-order Taylor expansion of the cosine function is utilised as follows:

$$c_{ij} = 1 - \theta_{ij}^2/2 + \theta_{ij}^4/24 - \theta_{ij}^6/720 \quad (53)$$

By introducing $\alpha_{ij} = \theta_{ij}^2$, $\beta_{ij} = \theta_{ij}^4$, $\gamma_{ij} = \theta_{ij}^6$, and define the following convex functions:

$$f_{ij,4}(x) = \alpha_{ij},\ g_{ij,4}(x) = \theta_{ij}^2 \quad (54)$$

$$f_{ij,5}(x) = \beta_{ij},\ g_{ij,5}(x) = \alpha_{ij}^2 \quad (55)$$

$$f_{ij,6}(x) = (\alpha_{ij}+\gamma_{ij})^2,\ g_{ij,6}(x) = (\alpha_{ij}-\gamma_{ij})^2 + (2\beta_{ij})^2 \quad (56)$$

(53) can be written in a difference-of-convex form:

$$c_{ij} = 1 - \alpha_{ij}/2 + \beta_{ij}/24 - \gamma_{ij}/720 \quad (57)$$

$$g_{ij,m}(x) - f_{ij,m}(x) \le 0, \quad \forall ij \in \Phi_l, m = 4,5,6 \quad (58)$$

$$f_{ij,m}(x) - g_{ij,m}(x) \le 0, \quad \forall ij \in \Phi_l, m = 4,5,6 \quad (59)$$

Here, only the cosine function (20) needs to be approximated because (21) is satisfied by difference-of-convex constraints (51) and (52).

It should be noted that in realistic power systems operation, θ^u is usually very small, i.e. <5°. In this situation, directly using $\sin(\theta_{ij}) = \theta_{ij}$ will also be a good approximation.

With difference-of-convex formulation (51), (52) for bilinear terms and accurate approximations (57)–(59) for trigonometric functions, OPF problem (Model 2) can be formulated as a DCP problem:

$$\text{(Model 4)}\ \min \sum C_i(p_i^g) \quad (60)$$

subject to (4), (7)–(15), (51), (52), (57)–(59).

Here, the constraints in (51) corresponding to $m = 1, 2$ and the constraints in (58) are convex. However, the constraints in (52) and (59) are non-convex.

Comparing Model 4 and the original OPF Model 2, all variables of Model 2 are included in Model 4, constraints (51) and (52) in Model 4 are equivalent to constraints (16), (20)–(22) in Model 2. The only different constraints between these two models are (19), (20) and (57)–(59). We consider constraints (57)–(59) as highly precise approximation of constraints (19)–(20) because the approximation error between (57) and (19), denoted by $|\cos\theta_{ij} - c_{ij}|$, is $<10^{-10}$ when $\theta_{ij} < 10^\circ$ and $<10^{-3}$ when $\theta_{ij} < 90^\circ$, which applies to power systems in most cases. Therefore, the feasible solution for Model 4 satisfies all of the constraints in Model 2, we can regard the solution of Model 4 as a feasible solution to Model 2.

3.2 Penalty convex–concave procedure

The OPF problem is formed as a DCP problem (Model 4) in part A; thus, penalty CCP can be applied to find a local optimum of Model 4. The procedure for penalty CCP is in two parts:

(i) Tighten the difference-of-convex constraints via partial linearisation. For example, $g_{ij,m}(x)$ can be linearised around point $x^{(0)}$ as

$$\hat{g}_{ij,m}(x, x^{(0)}) = g_{ij,m}(x^{(0)}) + \nabla g_{ij,m}(x^{(0)})^T (x - x^{(0)}) \quad (61)$$

Since $g_{ij,m}(x)$ is convex, we have $g_{ij,m}(x) \ge \hat{g}_{ij,m}(x, x^{(0)})$, and (52) can be tightened into a convex constraint

$$f_{ij,m}(x) - \hat{g}_{ij,m}(x, x^{(k)}) \le 0 \quad (62)$$

Constraint (62) reduces the feasible region of the original problem, which may lead to infeasibility, so part 2 is needed.

(ii) Relax constraint (62) by adding slack variables

$$f_{ij,m}(x) - \hat{g}_{ij,m}(x, x^{(k)}) \le \varepsilon \quad (63)$$

and penalise the sum of constraint violations in the objective function. By doing so, the problem is always feasible.

The steps for ACP OPF are described in Algorithm 1.

Algorithm 1: ACP OPF
Initialisation:

(i) Set the value of $x^{(0)}$ to the solution of Model 3.
(ii) Set $\tau^{(0)} > 0$, $\tau_{\max}$, $\mu > 1$ and $k = 0$.

Repeat
(i) Convexify

$$\hat{g}_{ij,m}(x, x^{(k)}) = g_{ij,m}(x^{(k)}) - \nabla g_{ij,m}(x^{(k)})^T (x - x^{(k)}), m = 1, \ldots, 6$$

$$\hat{f}_{ij,3}(x, x^{(k)}) = f_{ij,3}(x^{(k)}) - \nabla f_{ij,3}(x^{(k)})^T (x - x^{(k)})$$

(ii) Set the value of $x^{(k+1)}$ to the solution of

$$\text{(Model 5)}\ \min \sum C_i(p_i^g) + \tau^{(k)} \sum_{ij \in \Phi_l} \sum_{m=1}^{7} \varepsilon_{ij,m}^{(k)} \quad (64)$$

subject to (4), (7)–(15), (57)

$$f_{ij,m}(x) - \hat{g}_{ij,m}(x, x^{(k)}) \le \varepsilon_{ij,m}^{(k)}, \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \quad (65)$$

$$g_{ij,3}(x) - \hat{f}_{ij,3}(x, x^{(k)}) \le \varepsilon_{ij,7}^{(k)}, \forall ij \in \Phi_l \quad (66)$$

$$g_{ij,m}(x) - f_{ij,m}(x) \le 0, \quad \forall ij \in \Phi_l, m = 1,2,4,5,6 \quad (67)$$

$$\varepsilon_{ij,m}^{(k)} \ge 0, \quad \forall ij \in \Phi_l, m = 1, \ldots, 7 \quad (68)$$

(iii) Update $\tau^{(k+1)} = \min(\mu\tau^{(k)}, \tau_{\max})$(iv) Update iteration $k = k + 1$.
Until the stopping criterion is satisfied.

Model 5 is convex, and the number of second-order cone constraints, as well as quadratic constraints, grows linearly with the number of branches in the system. So, Model 5 can be solved easily and quickly using software packages such as Gurobi, CPLEX, or MOSEK. We have the following propositions regarding the convergence of ACP, proofs of which will be given in the Appendices.

Proposition 1: The objective value of Model 5 will converge.

Proposition 2: When the objective value of Model 5 converges after the k-th iteration, there exists a fixed optimal solution for every iteration after the k-th iteration.

Proposition 3: If the slack variables in Model 5 converge to zero, x^* will be a KKT point to Model 4, where (x^*, ε^*) denotes the optimal solution to Model 5.

According to Proposition 1, the stopping criterion of ACP OPF can be chosen as:

$$\frac{v^{(k)}(x^{(k)}, \varepsilon^{(k)}) - v^{(k+1)}(x^{(k+1)}, \varepsilon^{(k+1)})}{v^{(k+1)}(x^{(k)}, \varepsilon^{(k)})} \le \delta \tag{69}$$

when $k > \log_{\mu}(\tau_{\max}/\tau^{(0)})$, which indicates that the objective value converges.

When ACP converges, if the slack variables all turn out to be zero, then the solution of Model 5 is a feasible solution to Model 4. According to Proposition 3, the solution will be a KKT point to Model 4. Since the phase-angle difference is usually less 5° in power systems, Model 4 is considered to be equivalent to Model 2, then the solution will be a KKT point to the original OPF problem.

Although the objective value of Model 5 will converge, it may converge to an infeasible point of the original OPF problem if the slack variables are not equal to zero. The convergence behaviour of ACP depends mainly on two points:

(i) The penalty parameter $\tau_{\max}$.

According to [28], when $\tau_{\max}$ is preferred to be greater than the largest optimal dual variable related to Model 4, but this value is hard to be determined. In practice, we choose a large $\tau_{\max}$, meanwhile, $\tau_{\max}$ should not be too large in case of numerical problems.

(ii) The initial point.

A good starting point helps ACP finding a solution that all slack variables equal to zero. Since ACP aims to recover a feasible solution for SOCP relaxation, the initial point is chosen to be the result of convex-relaxed OPF Model 3, which is actually a good choice considering both optimality and computation speed. It should be clarified that the initial point for ACP means $x^{(k)}$ in (65) and (66) used for linearisation, while initial values for the whole problem is not needed because SOCP and ACP are both convex optimisation problems.

It is shown in the test results that, by choosing $\tau_{\max}$ and the initial point appropriately, ACP always converges to a feasible point where all the slack variables are equal to zero within a few iterations.

4 Numerical results

In this section, IEEE benchmark test systems were used to demonstrate the effectiveness of the proposed algorithm. First, the non-linear solver IPOPT was applied to find a local optimum of the original OPF problem. Then, two SDP-based heuristic models and a QCQP-based model aiming to recover feasible solutions for SDP relaxation were created to show whether they can achieve a feasible solution to the original OPF problem. Finally, the proposed ACP algorithm was tested to show its ability to recover a feasible solution for SOCP relaxation of the original OPF problem and compared with the other heuristic methods.

The SDP and QCQP-based heuristic methods used here are described as follows:

(i) Penalised SDP Relaxation in [9] (PSDP1). In this method, the total amount of reactive power is added to the objective function to force the matrix rank to become one.

(ii) Penalised SDP Relaxation in [20] (PSDP2). In this method, the matrix rank is approximated by a continuous function and penalised in the objective function. The penalised SDP problem is solved by majorisation-minimisation method iteratively.

(iii) CCP in [22] (QCCP). In this method, the OPF problem is first formulated as a QCQP, then the non-convex quadratic constraints are formulated as difference-of-convex inequalities and solved by CCP iteratively. The initial point of QCCP is provided by SDP relaxation and the difference-of-convex formulation of SDP rank one equality in [25] can be regarded as a specific decomposition of the quadratic constraints in QCQP.

The ACP algorithm, along with PSDP1, PSDP2, and QCCP, was implemented using YALMIP and MATLAB R2016a software. The SDP relaxation was implemented using sparse technique [33]. All the models were solved by MOSEK. Numerical tests were performed on a computer with an Intel® Core™ i5 (2.30 GHz) processor and 8 GB RAM. The original OPF problem was solved by MATPOWER using an IPOPT solver.

4.1 9-Bus test system

The IEEE 9-bus system consists of three generators and nine branches. The branch, bus, generator, and generator cost data of the system were taken from MATPOWER. There were three generators connected to buses 1, 2, and 3, and the total real and reactive power capacity were 0 to 820 MW and −900 to 900 MVar, respectively. The voltage of bus 1 was set to $1.0\angle 0°$. The lower and upper bounds of system bus voltages were 0.9 and 1.1 p.u., and the maximum phase-angle difference was 10°.

The solution of MATPOWER is assumed to be a benchmark solution to the original OPF problem, and the sub-optimality gap of the heuristic methods are defined as:

$$\text{Gap} = \frac{v^{\text{other}} - v^{\text{MP}}}{\left| v^{\text{MP}} \right|} \times 100\% \tag{70}$$

where v^{MP} is the objective value of the MATPOWER solution and v^{other} is the objective value of ACP, PSDP1, PSDP2, and QCCP.

To demonstrate the effectiveness of the proposed algorithm, different objective functions were tested:

(i) Generation cost minimisation (congested operation)

In this test case, we considered the cost of generators under congested operating conditions. The maximum apparent power for each branch was set to 120 MVA, and MATPOWER (MP) indicated that three branches were reaching its limit. Table 1 shows that ACP could recover a feasible solution to the original OPF problem from the result of tightened SOCP relaxation, as well as PSDP1, PSDP2, and QCCP, which were able to recover feasible solutions from SDP relaxations. While ACP and PSDP2 reached zero sub-optimality gap in this case, PSDP1 only recovered a near-global solution and QCCP recovered solutions far from global optimum.

The actual generation costs shown in Table 1 was calculated by the results of power flow, which used the power output of generators connected to buses 2 and 3 solved by each method as input to determine the power output of generator connected to bus 1 (slack bus). The active and reactive power output of each generator solved by MATPOWER (MP), power flow (PF), and ACP are shown in Table 2. From Tables 1 and 2, the generation cost and power output solved by ACP is the same as power flow, which proves that the solution recovered by ACP is a feasible solution, and the power output mismatch between ACP and MP is generally very small.

The computation time for each method is listed in Table 1. For the iterative methods, the first iteration was SOCPT or SDP relaxation that aimed to obtain an initial point for the heuristic methods. It can be observed that among these methods, ACP consumed much less computation time than the other three

Table 1 Numerical results of 9-bus system

	Obj. value	Gap, %	Iteration	Time, s	Actual cost (PF)
MP	5412.98	—	—	0.75	5412.98
ACP	5412.98	0.00	4	0.18	5412.98
PSDP1	5413.38	0.01	—	1.09	5413.38
PSDP2	5412.98	0.00	2	3.25	5412.98
QCCP	5430.72	0.33	5	5.38	5430.72

Table 2 Generator output of 9-bus system

Bus	Active power, MW			Reactive power, MVar		
	MP	ACP	PF	MP	ACP	PF
1	105.15	105.17	105.17	57.83	57.79	57.79
2	111.00	111.01	111.01	45.59	45.58	45.58
3	103.11	103.08	103.08	61.38	61.43	61.43

Table 3 Optimal voltage of 9-bus system

Bus	MATPOWER		ACP	
	V, p.u.	*θ*, °	V, p.u.	*θ*, °
1	1.000	0	1.000	0
2	1.100	−26.310	1.100	−26.305
3	1.100	−28.267	1.100	−28.262
4	0.931	−33.255	0.931	−33.251
5	0.942	−35.485	0.942	−35.482
6	1.054	−31.047	1.054	−31.043
7	1.030	−32.903	1.030	−32.899
8	1.047	−30.568	1.047	−30.564
9	0.926	−36.330	0.926	−36.327
maximum relative error, %			0.00	0.02

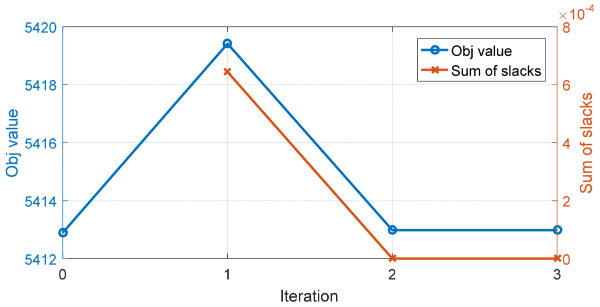

Fig. 1 *Convergence behavior of ACP for generation cost minimisation*

Table 4 Numerical results of 1354-bus system

	Obj. value	Gap, %	Iteration	Time, s	Actual loss (PF)
MP	984.28	—	—	2.79	984.28
ACP	967.64	−1.69	5	22.01	967.64

methods, because in each iteration, ACP solved a SOCP optimisation problem, while PSDP2 and QCCP solved a SDP optimisation problem. Although PSDP1 only needed to solve SDP once, the computation burden of SDP was much larger than SOCP even in a single iteration (Table 3).

Fig. 1 shows the objective values and sums of slack variables generated by ACP in each iteration: the parameters were set to $\tau^{(0)} = \tau_{\max} = 2500$ and $\delta = 10^{-6}$. It can also be seen that ACP converged in three iterations with the sum of slacks converged to zero, and the objective value generated by the ACP is non-increasing. Iteration 0 performed a SOCPT relaxation to obtain an initial point for ACP, which was not part of the ACP non-increasing sequence.

(ii) Generation cost minimisation (large phase angle difference)

The feasibility of ACP is depended on the accuracy of approximations (57)–(59), to test the behaviour of ACP when there exists large phase-angle differences in power systems, the reactance of branch 1–4 is changed form 0.0576 to 0.576 p.u., so that the phase-angle difference across this branch can be as large as 33.25°. In this test case, ACP converged in three iterations with slack variables all converged to zero and obtained the same objective as MATPOWER. The voltage magnitude and phase-angle data of each bus are shown in Table. Comparing the results of MATPOWER and ACP, the mismatches in voltage magnitudes and phase angles were both very small, which proved that the solution recovered by ACP was a feasible solution.

4.2 1354-Bus test system

The performance of ACP in large systems is tested on the PEGASE 1354-bus test system. The objective is loss minimisation and the numerical results of ACP and MATPOWER (MP) are shown in Table 4, the generator outputs of ACP is put into power flow program (PF) to verify whether the solution of ACP is feasible.

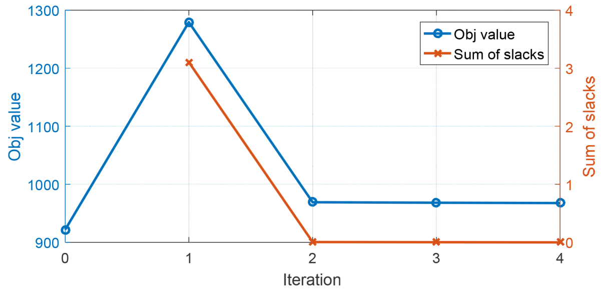

Fig. 2 *Convergence behavior of ACP for loss minimisation*

Table 5 Numerical results of IEEE benchmark systems

Test case	Ref. obj. value (MATPOWER)	Sub-optimality gap, %				Computation time, s				
		ACP	PSDP1	PSDP2	QCCP	MP	ACP	PSDP1	PSDP2	QCCP
loss minimisation problem, MW										
9	3.546	0.00	0.00	0.11	7.87	0.69	0.11	0.80	5.61	4.65
14	0.635	0.00	0.00	0.00	0.00	0.70	0.15	0.83	13.00	4.08
30	1.777	0.00	0.00	4.46	0.00	0.93	0.18	1.35	30.42	6.79
57	12.148	0.00	0.00	1.20	0.12	0.80	0.41	2.52	1883.82	26.05
118	10.667	0.00	0.00	*	0.83	1.09	0.92	10.96	>4 × 10^5	55.13
generation cost minimisation problem (congested operation) ($/h)										
9	5412.98	0.00	0.01	0.00	0.33	0.75	0.18	1.09	3.25	5.38
14	9252.28	0.00	0.00	×	0.00	0.77	0.32	0.94	5.48	4.28
30	582.79	0.00	0.07	0.02	0.14	1.13	5.31	2.23	58.05	38.46
57	43,697.64	0.00	0.00	×	0.00	0.76	1.46	4.08	1945.05	14.5
118	134,007.40	0.01	×	*	0.12	1.06	2.73	10.26	>4 × 10^5	62.99

× – Infeasible solution, *– numerical problems.

The convergence behaviour of ACP is shown in Fig. 2. The parameters were set to $\tau^{(0)} = \tau_{\max} = 100$ and $\delta = 10^{-6}$. ACP converged in four iterations with the sum of slacks converged to zero.

In this test case, ACP derives a lower objective value compared with MATPOWER, which means ACP performed better than IPOPT and find a solution more closer to the global optimal solution.

4.3 General test systems

The sub-optimality gap and computation time are listed in Table 5. Five test systems and two types of objective function were considered. Among the tested methods, ACP, PSDP1, PSDP2, and QCCP could find feasible solutions as good as MATPOWER (MP) in nine, seven, one, four of the ten test cases, respectively. Although none of the heuristic methods are guaranteed to recovery a feasible solution, the test results showed that ACP converged to feasible solutions with slack variables all equal to zero in all the test cases, which were also KKT points of the original OPF problem. On the other hand, the solution recovered by PSDP1 might not be a KKT point, and its behaviour was highly sensitive to the penalty parameter, which led to worse results than ACP. It can also be observed that ACP was generally much faster than SDP or QCQP-based methods. Since the setup time of YALMIP is highly influenced by the implementation technique but do not reflect the solving time of the model, only solver time is included in the computation time.

It needs to be pointed out that although we have shown the results and computation time of MATPOWER as reference, ACP is not intended to outperform any state-of-the-art OPF solvers. Instead, ACP seeks to recover a feasible solution for SOCP relaxation through convex optimisation. With this method, distributed optimisation and multi-stage robust optimisation problems incorporating convex-relaxed power flow equations can obtain a feasible and global or near-global optimal solution, so that the control effects will be guaranteed.

5 Conclusion

Here, an ACP algorithm was proposed to recover a global or near-global optimal solution for SOCP-relaxed OPF problem in mesh networks when the convex relaxation method is not exact. The OPF problem was first formulated as a DCP problem to maintain equality in the non-convex power flow equations, then solved efficiently by penalty CCP. A tightened SOCP relaxation of the OPF problem in mesh networks was also proposed to provide a good initial point for the ACP algorithm. Numerical results showed that the proposed algorithm could recover global or near-global optimal solutions for SOCP relaxation with various objective functions and generally performed better than the SDP or QCQP-based recovery methods in solution quality. The computational efficiency of the proposed algorithm was comparable to SOCP, which was far beyond the SDP or QCQP-based methods.

Since every iteration of ACP is a convex optimisation problem, the proposed method is suitable for more complicated optimisation problems in power systems such as distributed control or multi-stage robust optimisation, which require a convex formulation of both the power flow equations and the feasibility recovery model. The application of ACP in these problems deserves further investigation in our future research.

6 Acknowledgments

This work was supported in part by the National Key Research and Development Plan of China (Grant.2018YFB0904200) and in part by the National Science Foundation of China (Gant. 51725703)

7 References

[1] Bai, X., Wei, H., Fujisawa, K., *et al.*: 'Semidefinite programming for optimal power flow problems', *Int. J. Electr. Power Energy Syst.*, 2008, **30**, (6–7), pp. 383–392

[2] Jabr, R. A.: 'Radial distribution load flow using conic programming', *IEEE Trans. Power Syst.*, 2006, **21**, (3), pp. 1458–1459

[3] Lesieutre, B. C., Molzahn, D. K., Borden, A. R., *et al.*: 'Examining the limits of the application of semidefinite programming to power flow problems'. 2011 49th Annual Allerton Conf. on Communication, Control, and Computing (Allerton), Monticello, IL, 2011, pp. 1492–1499

[4] Jabr, R. A.: 'A conic quadratic format for the load flow equations of meshed networks', *IEEE Trans. Power Syst.*, 2007, **22**, (4), pp. 2285–2286

[5] Jabr, R.A.: ‘Optimal power flow using an extended conic quadratic formulation’, *IEEE Trans. Power Syst.*, 2008, **23**, (3), pp. 1000–1008

[6] Lavaei, J., Low, S. H.: ‘Zero duality Gap in optimal power flow problem’, *IEEE Trans. Power Syst.*, 2012, **27**, (1), pp. 92–107

[7] Low, S. H.: ‘Convex relaxation of optimal power flow—part I: formulations and equivalence’, *IEEE Trans. Control Netw. Syst.*, 2014, **1**, (1), pp. 15–27

[8] Low, S. H.: ‘Convex relaxation of optimal power flow—part II: exactness’, *IEEE Trans. Control Netw. Syst.*, 2014, **1**, (2), pp. 177–189

[9] Madani, R., Sojoudi, S., Lavaei, J.: ‘Convex relaxation for optimal power flow problem: mesh networks’, *IEEE Trans. Power Syst.*, 2015, **30**, (1), pp. 199–211

[10] Kocuk, B., Dey, S. S., Sun, X. A.: ‘Inexactness of SDP relaxation and valid inequalities for optimal power flow’, *IEEE Trans. Power Syst.*, 2016, **31**, (1), pp. 642–651

[11] Farivar, M., Low, S. H.: ‘Branch flow model: relaxations and convexification—part I’, *IEEE Trans. Power Syst.*, 2013, **28**, (3), pp. 2554–2564

[12] Huang, S., Wu, Q., Wang, J., *et al.*: ‘A sufficient condition on convex relaxation of AC optimal power flow in distribution networks’, *IEEE Trans. Power Syst.*, 2017, **32**, (2), pp. 1359–1368

[13] Farivar, M., Low, S. H.: ‘Branch flow model: relaxations and convexification—part II’, *IEEE Trans. Power Syst.*, 2013, **28**, (3), pp. 2565–2572

[14] Kocuk, B., Dey, S. S., Sun, X. A.: ‘Strong SOCP relaxations for optimal power flow’, *Oper. Res.*, 2016, **64**, (6), pp. 1177–1196

[15] Kocuk, B., Dey, S. S., Sun, X. A.: ‘New formulation and strong MISOCP relaxations for AC optimal transmission switching problem’, *IEEE Trans. Power Syst.*, 2017, **32**, (6), pp. 4161–4170

[16] Liu, H. J., Shi, W., Zhu, H.: ‘Distributed voltage control in distribution networks: online and robust implementations’, *IEEE Trans. Smart Grid*, 2018, **9**, (6), pp. 6106–6117

[17] Zheng, W., Wu, W., Zhang, B., *et al.*: ‘A fully distributed reactive power optimization and control method for active distribution networks’, *IEEE Trans. Smart Grid*, 2016, **7**, (2), pp. 1021–1033

[18] Lorca, Á., Sun, X. A.: ‘Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind’, *IEEE Trans. Power Syst.*, 2015, **30**, (4), pp. 1702–1713

[19] Zeng, B., Zhao, L.: ‘Solving two-stage robust optimization problems using a column-and-constraint generation method’, *Oper. Res. Lett.*, 2013, **41**, (5), pp. 457–446

[20] Liu, T., Sun, B., Tsang, D. H. K.: ‘Rank-one solutions for SDP relaxation of QCQPs in power systems’, *IEEE Trans. Smart Grid*, 2019, **10**, (1), pp. 5–15

[21] Molzahn, D. K., Hiskens, I. A.: ‘Moment-based relaxation of the optimal power flow problem’. 2014 Power Systems Computation Conf., Wroclaw, 2014, pp. 1–7

[22] Zamzam, A. S., Sidiropoulos, N. D., Dall'Anese, E.: ‘Beyond relaxation and Newton-Raphson: solving AC OPF for multi-phase systems with renewables’, *IEEE Trans. Smart Grid*, 2018, **9**, (5), pp. 3966–3975

[23] Merkli, S., Domahidi, A., Jerez, J., *et al.*: ‘Fast AC power flow optimization using difference of convex functions programming’, *IEEE Trans. Power Syst.*, 2018, **33**, (1), pp. 363–372

[24] Wei, W., Wang, J., Li, N., *et al.*: ‘Optimal power flow of radial networks and its variations: a sequential convex optimization approach’, *IEEE Trans. Smart Grid*, 2017, **8**, (6), pp. 2974–2987

[25] Park, J., Boyd, S.: ‘General heuristics for nonconvex quadratically constrained quadratic programming’, arXiv preprint arXiv:1703.07870, 2017

[26] Yuille, L., Rangarajan, A.: ‘The concave-convex procedure’, *Neural Comput..*, 2003, **15**, (4), pp. 915–936

[27] Smola, J., Vishwanathan, S. V. N., Hofmann, T.: ‘Kernel methods for missing variables’. Proc. 10th Int. Workshop Artif. Intell. Stat., Barbados, Mar. 2005, pp. 325–332

[28] Lipp, T., Boyd, S.: ‘Variations and extension of the convex-concave procedure’, *Optim. Eng.*, 2016, **17**, (2), pp. 263–287

[29] Wu, W., Tian, Z., Zhang, B.: ‘An exact linearization method for OLTC of transformer in branch flow model’, *IEEE Trans. Power Syst.*, 2017, **32**, (3), pp. 2475–2476

[30] Tian, Z., Wu, W., Zhang, B., *et al.*: ‘Mixed-integer second-order cone programing model for VAR optimisation and network reconfiguration in active distribution networks’, *IET Gener. Transm. Distrib.*, 2016, **10**, (8), pp. 1938–1946

[31] Hijazi, H., Coffrin, C., Van Hentenryck, P.: ‘Convex quadratic relaxations of mixed-integer nonlinear programs in power systems’. Tech. Rep., NICTA, ACT Australia, Canberra, 2013

[32] McCormick, G.: ‘Computability of global solutions to factorable nonconvex programs: part I convex underestimating problems’, *Math. Program.*, 1976, **10**, (1), pp. 147–175

[33] Madani, R., Ashraphijuo, M., Lavaei, J.: ‘Promises of conic relaxation for contingency-constrained optimal power flow problem’, *IEEE Trans. Power Syst.*, 2016, **31**, (2), pp. 1297–1307

7 Appendix

Proposition 1: The objective value of Model 5 will converge.

Proof: Suppose $(x^{(k)}, \varepsilon^{(k)})$ is the optimal solution to Model 5 in iteration k.

We will first prove that $(x^{(k)}, \varepsilon^{(k)})$ is a feasible solution to Model 5 in iteration $k + 1$. Since the different constraints in iteration k and $k + 1$ are (65) and (66), it suffices to show that $(x^{(k)}, \varepsilon^{(k)})$ satisfies (65) and (66) in iteration $k + 1$. That is to prove:

$$f_{ij,m}(x^{(k)}) - \hat{g}_{ij,m}(x^{(k)}, x^{(k)}) \le \varepsilon^{(k)}_{ij,m}, \quad m = 1, \ldots, 6 \tag{71}$$

$$g_{ij,3}(x^{(k)}) - \hat{f}_{ij,3}(x^{(k)}, x^{(k)}) \le \varepsilon^{(k)}_{ij,7} \tag{72}$$

As $(x^{(k)}, \varepsilon^{(k)})$ is the optimal solution to iteration k, we have

$$f_{ij,m}(x^{(k)}) - \hat{g}_{ij,m}(x^{(k)}, x^{(k-1)}) \le \varepsilon^{(k)}_{ij,m} \tag{73}$$

The convexity of $g_{ij,m}(x)$ gives

$$f_{ij,m}(x^{(k)}) - g_{ij,m}(x^{(k)}) \le f_{ij,m}(x^{(k)}) - \hat{g}_{ij,m}(x^{(k)}, x^{(k-1)}) \tag{74}$$

Substituting $g_{ij,m}(x^{(k)}) = \hat{g}_{ij,m}(x^{(k)}, x^{(k)})$ into (74), together with (73), we have

$$f_{ij,m}(x^{(k)}) - \hat{g}_{ij,m}(x^{(k)}, x^{(k)}) \le \varepsilon^{(k)}_{ij,m} \tag{75}$$

Thus, (71) holds, and (72) can be proved in a similar way. So $(x^{(k)}, \varepsilon^{(k)})$ is a feasible solution to Model 5 in iteration $k + 1$.

We will now show that the objective value is non-increasing. Let $v^{(k)}(x, \varepsilon)$ denote the objective function of Model 5 in iteration k. When $k > \log_{\mu}(\tau_{\max}/\tau^{(0)})$, $\tau^{(k)} = \tau_{\max}$, the objective function (64) will not change, which means

$$v^{(k+1)}(x, \varepsilon) = v^{(k)}(x, \varepsilon) \tag{76}$$

Since $(x^{(k)}, \varepsilon^{(k)})$ is a feasible solution to Model 5 in iteration $k + 1$ and $(x^{(k+1)}, \varepsilon^{(k+1)})$ is the optimal solution, it follows that

$$v^{(k+1)}(x^{(k+1)}, \varepsilon^{(k+1)}) \le v^{(k+1)}(x^{(k)}, \varepsilon^{(k)}) = v^{(k)}(x^{(k)}, \varepsilon^{(k)}) \tag{77}$$

This shows that the objective value is non-increasing. Since both $\sum C_i(p_i^g)$ and ε have lower bounds, the objective value will converge, which completes the proof. □

Proposition 2: When the objective value of Model 5 converges after the k-th iteration, there exists a fixed optimal solution for every iteration after the k-th iteration.

Proof: According to Proposition 1, the objective value of Model 5 will converge. Suppose the objective value converges after the k-th iteration, which means

$$v^{(k)}(x^{(k)}, \varepsilon^{(k)}) = v^{(k+1)}(x^{(k+1)}, \varepsilon^{(k+1)}) \tag{78}$$

Since we have proved in Proposition 1 that $(x^{(k)}, \varepsilon^{(k)})$ is a feasible solution to Model 5 in iteration $k + 1$ and the objective value $v^{(k+1)}(x^{(k)}, \varepsilon^{(k)}) = v^{(k)}(x^{(k)}, \varepsilon^{(k)})$, therefore

$$v^{(k+1)}(x^{(k)}, \varepsilon^{(k)}) = v^{(k+1)}(x^{(k+1)}, \varepsilon^{(k+1)}) \tag{79}$$

Since the objective function $v^{(k+1)}(x, \varepsilon)$ is not strictly convex, if the problem of $k + 1$-th iteration have multiple optimal solutions, it is possible that $(x^{(k)}, \varepsilon^{(k)})$ is not identical to $(x^{(k+1)}, \varepsilon^{(k+1)})$. According to (79), it is also certain that $(x^{(k)}, \varepsilon^{(k)})$ is one of the optimal solutions to Model 5 in $k + 1$-th iteration. Therefore, the optimal solution of the k-th iteration will also be an optimal solution of the $k + 1$-th iteration, furtherly, it is a fixed optimal solution for every iteration after the k-th iteration. □

Proposition 3: If the slack variables in Model 5 converge to zero, x^* will be a KKT point to Model 4, where (x^*, ε^*) denotes the optimal solution to Model 5.

Proof: By denoting $\sum C_i(p_i^g)$ as $c(x)$, $\tau^{(k)}\sum_{ij\in\Phi_l}\sum_{m=1}^{7}\varepsilon_{ij,m}^{(k)}$ as $e(\varepsilon)$ in objective function, inequality constraint functions in (4), (7)–(15), (57) as $u_i(x)$, equality constraints in (4), (7)–(15), (57) as $v_i(x)$, constraint (65) as $p_{ij,m}(x,\varepsilon)$, constraint (66) and (67) as $q_{ij,m}(x,\varepsilon)$, we can reformulate Model 5 as:

$$\text{(Model 5) min } c(x)+e(\varepsilon) \tag{80}$$

subject to

$$u_i(x)\le 0,\ i=1,\dots,N_u \tag{81}$$

$$v_i(x)=0,\ i=1,\dots,N_v \tag{82}$$

$$p_{ij,m}(x,\varepsilon)\le 0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{83}$$

$$q_{ij,m}(x,\varepsilon)\le 0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{84}$$

$$-\varepsilon\le 0,\quad \forall ij\in\Phi_l, m=1,\dots,7 \tag{85}$$

According to Proposition 2, we can choose (x^*,ε^*) as an fixed optimal solution for Model 5, which means when Model 5 converges, $g_{ij,m}(x)$ and $f_{ij,m}(x)$ are linearised at x^*, such that the gradients of the inequality constraints (83)–(85) are linearly dependent at (x^*,ε^*). Notice that when $\varepsilon_{ij,7}=0$, together with $p_{ij,m}(x,\varepsilon)=0$ and $q_{ij,m}(x,\varepsilon)=0$, there must be $\varepsilon_{ij,m}\ge 0$ for $m=1,\dots,6$, we can consider the following model in the proof:

$$\text{(Model 6) min } c(x)+e(\varepsilon) \tag{86}$$

subject to (81), (82)

$$p_{ij,m}(x,\varepsilon)=0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{87}$$

$$q_{ij,m}(x,\varepsilon)=0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{88}$$

$$\varepsilon=0,\quad \forall ij\in\Phi_l, m=7 \tag{89}$$

Since every feasible solution to Model 6 is feasible to Model 5, and (x^*,ε^*) is a feasible solution to Model 6, it is obvious that (x^*,ε^*) is also the optimal solution to Model 6. While the gradients of the constraints (83)–(85) are linearly dependent at (x^*,ε^*) in Model 5, we will prove that the gradients of constraints (87)–(89) are linearly independent at (x^*,ε^*) in Model 6.

The gradients of constraint (87) at (x^*,ε^*) are:

$$\left.\begin{bmatrix}\nabla_x p_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,m}} p_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,7}} p_{ij,m}(x,\varepsilon)\end{bmatrix}\right\|_{(x,\varepsilon)=(x^*,\varepsilon^*)}=\begin{bmatrix}\nabla f_{ij,m}(x^*)-\nabla g_{ij,m}(x^*)\\ -1\\ 0\end{bmatrix},\quad m=1,\dots,6 \tag{90}$$

The gradients of constraint (88) at (x^*,ε^*) are:

$$\left.\begin{bmatrix}\nabla_x q_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,m}} q_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,7}} q_{ij,m}(x,\varepsilon)\end{bmatrix}\right\|_{(x,\varepsilon)=(x^*,\varepsilon^*)}=\begin{bmatrix}\nabla g_{ij,m}(x^*)-\nabla f_{ij,m}(x^*)\\ 0\\ 0\end{bmatrix},\quad m=1,2,4,5,6 \tag{91}$$

$$\left.\begin{bmatrix}\nabla_x q_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,m}} q_{ij,m}(x,\varepsilon)\\ \nabla_{\varepsilon_{ij,7}} q_{ij,m}(x,\varepsilon)\end{bmatrix}\right\|_{(x,\varepsilon)=(x^*,\varepsilon^*)}=\begin{bmatrix}\nabla g_{ij,m}(x^*)-\nabla f_{ij,m}(x^*)\\ 0\\ -1\end{bmatrix},\quad m=3 \tag{92}$$

The gradients of constraint (89) at (x^*,ε^*) are:

$$\left.\begin{bmatrix}\nabla_x \varepsilon_{ij,7}\\ \nabla_{\varepsilon_{ij,m}} \varepsilon_{ij,7}\\ \nabla_{\varepsilon_{ij,7}} \varepsilon_{ij,7}\end{bmatrix}\right\|_{(x,\varepsilon)=(x^*,\varepsilon^*)}=\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{93}$$

It is easy to check that the gradients of (87) itself are linearly independent, as well as the gradients of (88) and (89). If the gradients of (87)–(89) are linearly dependent at (x^*,ε^*), there must exist constants $\alpha_1,\alpha_2,\alpha_3$ which are not all zeros for $\forall m\in\{1,2,4,5,6\}$ and $\forall ij\in\Phi_l$, such that

$$\alpha_1\begin{bmatrix}\nabla f_{ij,m}(x^*)-\nabla g_{ij,m}(x^*)\\ -1\\ 0\end{bmatrix}+\alpha_2\begin{bmatrix}\nabla g_{ij,m}(x^*)-\nabla f_{ij,m}(x^*)\\ 0\\ 0\end{bmatrix}+\alpha_3\begin{bmatrix}0\\0\\1\end{bmatrix}=0 \tag{94}$$

and there must exist constants β_1,β_2,β_3 which are not all zeros for $m=3$ and $\forall ij\in\Phi_l$, such that

$$\beta_1\begin{bmatrix}\nabla f_{ij,m}(x^*)-\nabla g_{ij,m}(x^*)\\ -1\\ 0\end{bmatrix}+\beta_2\begin{bmatrix}\nabla g_{ij,m}(x^*)-\nabla f_{ij,m}(x^*)\\ 0\\ -1\end{bmatrix}+\beta_3\begin{bmatrix}0\\0\\1\end{bmatrix}=0 \tag{95}$$

The only solution to (94) is $\alpha_1=\alpha_2=\alpha_3=0$ and the only solution to (95) is $\beta_1=\beta_2=\beta_3=0$, so the gradients of constraints (87)–(89) are linearly independent at (x^*,ε^*).

Constraints (81) and (82) are conventional optimal power flow constraints whose gradients are linearly independent. Due to the introduction of slack variables ε, it is easy to check that the gradients of (81) and (82) are linearly independent with (87)–(89). So, the gradients of the constraints in Model 6 are linearly independent at (x^*,ε^*).

By denoting constraint (52) and (59) as $p'_{ij,m}(x)$, constraint (51) and (58) as $q'_{ij,m}(x)$, and notice that these four constraints are equivalent to:

$$p'_{ij,m}(x)=0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{96}$$

We can reformulate Model 4 as:

$$\text{(Model 4) min } c(x) \tag{97}$$

subject to (81), (82)

$$p'_{ij,m}(x)=0,\quad \forall ij\in\Phi_l, m=1,\dots,6 \tag{98}$$

With the above transformation, Model 4 is composed of the conventional optimal power flow constraints and the Taylor expansion equations. Since these two parts of constraints are linearly independent with themselves and they have largely different variables, it is easy to check that the gradients of the constraints in Model 4 is linearly independent at x^*.

Since the objective functions and constraint functions of Model 4 and Model 6 are continuously differentiable at x^* and (x^*,ε^*), and that Model 4 and Model 6 satisfy the linear independence constraint qualification. We can derive the KKT conditions for these two models.

(x^*,ε^*) is the optimal solution to Model 6, according to KKT conditions, there exist constants λ_i, $\mu_{ij,m}^p$, $\mu_{ij,m}^q$, μ_i and μ_ε such that

$$\begin{aligned}&\nabla c(x^*) + \nabla e(\varepsilon^*) + \sum_{i=1}^{N_u} \lambda_i \nabla u_i(x^*) + \sum_{i=1}^{N_v} \mu_i \nabla v_i(x^*)\\ &+ \sum_{ij \in \Phi_l} \sum_{m=1}^{6} (\mu_{ij,m}^p \nabla p_{ij,m}(x^*, \varepsilon^*) + \mu_{ij,m}^q \nabla q_{ij,m}(x^*, \varepsilon^*))\\ &+ \sum_{ij \in \Phi_l} \mu_\varepsilon \nabla \varepsilon_{ij,7} = 0\end{aligned} \tag{99}$$

$$u_i(x^*) \le 0,\ i = 1, \ldots, N_u \tag{100}$$

$$\lambda_i u_i(x^*) = 0,\ i = 1, \ldots, N_u \tag{101}$$

$$v_i(x^*) = 0,\ i = 1, \ldots, N_v \tag{102}$$

$$\varepsilon = 0,\ \forall ij \in \Phi_l, m = 7 \tag{103}$$

$$p_{ij,m}(x^*, \varepsilon^*) = 0, \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \tag{104}$$

$$q_{ij,m}(x^*, \varepsilon^*) = 0, \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \tag{105}$$

$$\lambda_i \ge 0 \tag{106}$$

Since $g_{ij,m}(x)$ and $f_{ij,m}(x)$ in Model 6 are linearised at x^*, we have

$$p_{ij,m}(x^*, \varepsilon^*) = p'_{ij,m}(x^*), \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \tag{107}$$

$$\nabla_x p_{ij,m}(x^*, \varepsilon^*) = \nabla_x p'_{ij,m}(x^*), \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \tag{108}$$

$$\nabla_x q_{ij,m}(x^*, \varepsilon^*) = -\nabla_x p_{ij,m}(x^*, \varepsilon^*), \quad \forall ij \in \Phi_l, m = 1, \ldots, 6 \tag{109}$$

Consider the partial derivatives with respect to x in (3)–(17), there will be (see (110)) . Substitute (107) into (104), (108) and (109) into (110), there will be

$$\begin{aligned}&\nabla_x c(x^*) + \sum_{i=1}^{N_u} \lambda_i \nabla_x u_i(x^*) + \sum_{i=1}^{N_v} \mu_i \nabla_x v_i(x^*)\\ &+ \sum_{ij \in \Phi_l} \sum_{m=1}^{6} (\mu_{ij,m}^p - \mu_{ij,m}^q) \nabla_x p'{}_{ij,m}(x^*) = 0\end{aligned} \tag{111}$$

$$u_i(x^*) \le 0,\ i = 1, \ldots, N_u \tag{112}$$

$$\lambda_i u_i(x^*) = 0,\ i = 1, \ldots, N_u \tag{113}$$

$$v_i(x^*) = 0,\ i = 1, \ldots, N_v \tag{114}$$

$$p'_{ij,m}(x^*) = 0, \quad \forall ij \in \Phi_l,\ m = 1, \ldots, 6 \tag{115}$$

$$\lambda_i \ge 0 \tag{116}$$

Constraints (111)–(116) are exactly the KKT conditions of Model 4, so x^* is a KKT point of Model 4. □

$$\begin{aligned}&\nabla_x c(x^*) + \sum_{i=1}^{N_u} \lambda_i \nabla_x u_i(x^*) + \sum_{i=1}^{N_v} \mu_i \nabla_x v_i(x^*)\\ &+ \sum_{ij \in \Phi_l} \sum_{m=1}^{6} (\mu_{ij,m}^p \nabla_x p_{ij,m}(x^*, \varepsilon^*) + \mu_{ij,m}^q \nabla_x q_{ij,m}(x^*, \varepsilon^*)) = 0\end{aligned} \tag{110}$$