Combinatorial Optimization of Alternating Current Electric Power Systems

Suggested Citation: Sid Chi-Kin Chau, Khaled Elbassioni and Majid Khonji (2018), "Combinatorial Optimization of Alternating Current Electric Power Systems", : Vol. 3, No. 1-2, pp 1–139. DOI: 10.1561/3100000017.

Sid Chi-Kin Chau
Australian National University
sid.chau@anu.edu.au

Khaled Elbassioni
Khalifa University of Science and Technology
khaled.elbassioni@ku.ac.ae

Majid Khonji
Khalifa University of Science and Technology
majid.khonji@ku.ac.ae

now
the essence of knowledge
Boston — Delft

Contents

Combinatorial Optimization of Alternating Current Electric Power Systems

Sid Chi-Kin Chau[1], Khaled Elbassioni[2] and Majid Khonji[3]

[1] *Australian National University; sid.chau@anu.edu.au*
[2] *Khalifa University; khaled.elbassioni@ku.ac.ae*
[3] *Khalifa University; majid.khonji@ku.ac.ae*

ABSTRACT

In the era of dynamic smart grid with fluctuating demands and uncertain renewable energy supplies, it is crucial to continuously optimize the operational cost and performance of electric power grid, while maintaining its state within the stable operating limits. Nonetheless, a major part of electric power grid consists of alternating current (AC) electric power systems, which exhibit complex behavior with non-linear operating constraints. The optimization of AC electric power systems with dynamic demands and supplies is a very challenging problem for electrical power engineers.

The hardness of optimization problems of AC electric power systems stems from two issues: (1) *non-convexity* involving complex-valued entities of electric power systems, and (2) combinatorial constraints involving *discrete* control variables. Without proper theoretical tools, heuristic methods or general numerical solvers had been utilized traditionally to tackle these problems, which do not provide theoretical guarantees of the achieved solutions with respect to the true optimal solutions. There have been recent advances in

Sid Chi-Kin Chau, Khaled Elbassioni and Majid Khonji (2018), "Combinatorial Optimization of Alternating Current Electric Power Systems", : Vol. 3, No. 1-2, pp 1–139. DOI: 10.1561/3100000017.

applying convex relaxations to tackle non-convex problems of AC electric power systems. On the other hand, discrete combinatorial optimization is rooted in theoretical computer science, which typically considers linear constraints, instead of those non-linear constraints in AC electric power systems.

To bridge power systems engineering and theoretical computer science, this monograph presents a comprehensive study of combinatorial optimization of AC electric power systems with (inelastic) discrete demands. The main idea of this monograph is to draw on new extensions of discrete combinatorial optimization with linear constraints, like knapsack and unsplittable flow problems. We present approximation algorithms and inapproximability results for various settings from (1) basic single-capacitated AC electric power systems, to (2) constant-sized AC electric grid networks with power flows, and (3) scheduling of AC electric power. This monograph aims to establish a foundation for the inter-disciplinary problems of power systems engineering and theoretical computer science.

1

Introduction

1.1 Need for Optimization in Smart Power Grid

The electric power grid has been an indispensable part of our society, empowering the economic and social activities in every aspect of our daily lives. Our society is consuming a tremendous amount of energy at an increasing rate. There has been a drastic surge in global energy consumption. As a result, the power grid needs to undergo transformations to meet the new challenges for a more sustainable society:

- *Deregulation of Power Industry*: Replacing the monopolized industry of power grid in generation, transmission and distribution by decentralized operators with heterogeneous requirements.

- *Decarbonization and Incorporation of Renewable Energy*: Transitioning from fossil fuel energy to environment-friendly but uncertain renewable energy supplies.

- *Demand Responsiveness*: Shifting the traditional power grid that is engineered for peak demands to be more demand responsive, such that grid operators and end users can react to variable grid resources by dynamic pricing and electricity markets.

- *Inefficiency Elimination*: Reducing the energy loss in power generation and transmission by employing technologies, such as Combined Heat-and-Power (CHP) generation and Flexible AC Transmission Systems (FACTS).

- *Disruption Protection*: Enabling more robust control against outage and power failures by incorporating autonomous microgrids and emergence demand response management.

These transformations will create a *smarter* power grid with improved energy-efficiency, responsiveness and stability. In particular, there is a need for *continuous optimization* in smart grid that can react rapidly to dynamic situations in presence of fluctuating demands and uncertain renewable energy. In the past, the operations of power grid relied on careful a-priori planning, under the assumptions of static demands and predictable circumstances. In the era of dynamic smart grid, self-optimization with adaptive control is more crucial to its operations.

There are several factors for consideration in the optimization of power grid operations:

- *Scale*: Power grid is connected to an increasing number of users and loads, with growing presence of electric vehicles and smart appliances. These demands have to be optimally coordinated and regulated in a large-scale manner.

- *Time*: The fluctuations of renewable energy supplies and demands under dynamic pricing occur more significantly in a shorter timescale. Power grid needs to adapt to intermittency rapidly.

- *Performance*: A variety of performance objectives ought to be considered by different parties among energy suppliers, transmitters, distributors, regulators and residential/commercial end users.

- *Stability*: The stability operating constraints of the power grid need be adhered and validated from time to time to ensure reliable operations.

Therefore, it is critical to continuously optimize the power grid under various performance objectives in a scalable and responsive manner, while maintaining its state within the stable operating limits.

However, a power grid is a large complex system. In particular, a major part of the power grid is composed of alternating current (AC) electric power systems, which exhibit complex behavior with non-linear operating constraints. The effective management and control operations of AC electric power systems involve very challenging problems that baffle electrical power engineers. The hardness of optimization problems in AC electric power systems stems mainly from two issues:

1. *Non-Convex Constraints* involving complex-valued variables and parameters of AC electric power systems.

2. *Combinatorial Constraints* involving *discrete* control variables for the operation of power systems.

Traditionally, heuristic methods or general numerical solvers had been utilized for the combinatorial optimization problems of AC electric power systems, without proper theoretical analyses on the performance, efficiency and optimality of the results. Some of these methods return inefficient algorithms that are not scalable in larger systems, or fail to provide guarantees on the deviation of output solutions from the true optimal solutions.

Combinatorial optimization has been extensively studied in theoretical computer science, with diverse applications in operations research and engineering science beyond computing systems. Hence, it is imperative to draw on the related tools from theoretical computer science to study the problems arising from smart grid. In particular, there are recent advances in approximation algorithms with provable approximation ratios that can be applied in combinatorial power systems problems.

This monograph aims to establish an interdisciplinary bridge between power systems engineering and theoretical computer science by relating the practical and challenging problems in electric power systems with the modern theoretical tools from computer science. The proper understanding of these hard problems in electric power systems can advance the frontiers of both communities. Particularly, this monograph is tailored for these two groups of audience:

- For *Power System Engineers*, it introduces the concepts and results of approximation algorithms, and applies them to solve electric power systems problems.

- For *Computer Scientists*, it provides an exposition of a class of challenging combinatorial problems in electric power systems.

Before presenting the approximation algorithms for AC power systems in the subsequent chapters, this section first explains the basics of AC electric power systems, and then some standard terminology of approximation algorithms in the literature.

1.2 Basics of AC Electric Power Systems

First, we give an example scenario of power consumption scheduling problem as illustrated in Fig. 1.1. There are multiple households and electric vehicles connecting to the power grid with dynamic renewable energy supplies. In each household, there are electric appliances that can only be controlled by switching on or off. For charging electric vehicles, there are currently three main categories of charging infrastructure standards: *Level 1* charging with cord-set singe-phase connections to a regular household outlet, *Level 2* wall-mount three-phase connections, and *Level 3* DC fast charging. It is worth noting that none of these current popular charging standards allows continuously controllable charging power at an arbitrary rate. To ensure reliable charging, there requires a delicate control system for the supplied charging power. Hence, the charging power normally varies within a limited discrete set of nearly constant values (Gan *et al.*, 2012). The scheduling of power consumption with discrete controls is a natural combinatorial optimization problem that is studied in this monograph.

1.2.1 Notations

This section presents the basics of electric power systems. More details of electric power systems can be found in a standard power systems textbook (e.g., Grainger and Stevenson, 1994). An electric power system is characterized by an electric network with nodes (also called buses) and edges (also called lines). A power flow in an electric network is described

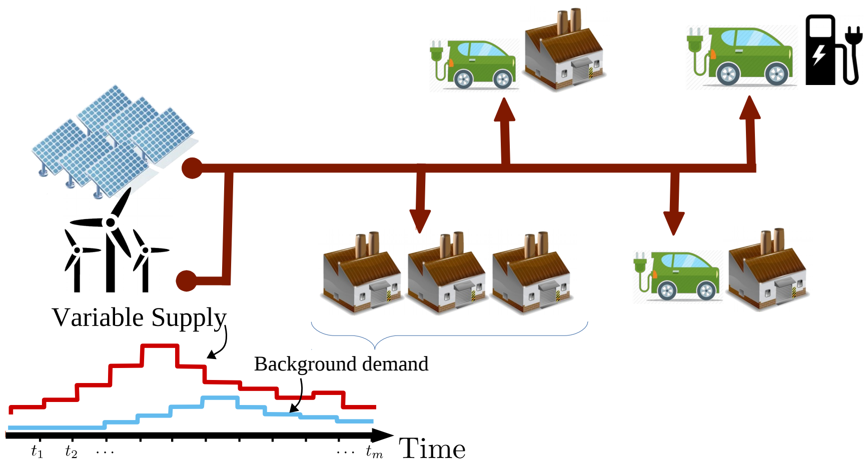


Figure 1.1: An illustration of power consumption scheduling problem.

by physical quantities such as current, voltage and power. We represent an electric network by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of nodes and $\mathcal{E}$ denotes a set of edges. We index the nodes in $\mathcal{V}$ by $\{0, 1..., m\}$, where $m \triangleq |\mathcal{V}|$. Node 0 usually carries a special meaning (called slack bus). If $\mathcal{G}$ represents an electric distribution network, then node 0 usually denotes the generation source or the feeder to the main grid. Let $\mathcal{V}^+ \triangleq \mathcal{V}\backslash\{0\}$. We fix an *arbitrary* orientation on the edges, and think of $\mathcal{G}$ as a directed graph. For convenience, we choose an orientation such that $\mathcal{G}$ forms a directed *acyclic* graph where the "power flow" from node 0 to the rest of nodes in $\mathcal{V}^+$[1]. Thus, in the rest of monograph, we assume that the orientation of a directed edge (i, j) designates that the current or power flows from i to j.

For node $i \in \mathcal{V}$, we denote its *voltage* by V_i. For each edge $e = (i, j) \in \mathcal{E}$, we denote its *current* from i to j by $I_{i,j}$, its *transmitted power* by $S_{i,j}$, and its *impedance* by $z_{i,j}$. In direct current (DC) electric systems, all quantities belong to the set of real numbers (denoted by $\mathbb{R}$); whereas in alternating current (AC) electric systems, these quantities

[1]Such orientation can always be obtained by first finding a spanning tree $\mathcal{T}$ on $\mathcal{V}$ and rooting it at node 0, then orienting all edges of $\mathcal{T}$ away from 0, with end points on directed paths in $\mathcal{T}$, and then orienting all other edges arbitrarily.

belong to the set of complex numbers (denoted by $\mathbb{C}$). Usually, the voltage V_0 at node 0 is normalized as $V_0 = 1$.

For a complex number $\nu \in \mathbb{C}$, we denote the *magnitude* of ν by $|\nu|$, the *phase angle* (or argument) that ν makes with the real axis by $\angle\nu$, and the complex *conjugate* of ν by ν^*. We sometimes write $\nu^{\mathrm{R}} \triangleq \mathrm{Re}(\nu)$ for the real part and $\nu^{\mathrm{I}} \triangleq \mathrm{Im}(\nu)$ for the imaginary part of ν. For $\nu, \nu' \in \mathbb{C}$, we write $\nu \leq \nu'$ to mean $\nu^{\mathrm{R}} \leq \nu'^{\mathrm{R}}$ and $\nu^{\mathrm{I}} \leq \nu'^{\mathrm{I}}$.

There are several basic laws governing the relationships of the quantities $V_i, I_{i,j}, z_{i,j}, S_{i,j}$ in an electric network:

- *Ohm's Law*: For each $(i,j) \in \mathcal{E}$,
$$V_i - V_j = z_{i,j} I_{i,j}. \tag{1.1}$$
- *Kirchhoff's Current Law*: For node $i \in \mathcal{V}$,
$$\sum_{(i,j)\in\mathcal{E}} I_{i,j} = 0. \tag{1.2}$$
- *Electric Power Formula*: For each $(i,j) \in \mathcal{E}$,
$$S_{i,j} = V_i I_{i,j}^*. \tag{1.3}$$

Additionally, by convention, the following *skew symmetry* relation holds:
$$I_{i,j} = -I_{j,i}. \tag{1.4}$$

Each node $i \in \mathcal{V}$ is associated with a power injection/extraction s_i, which represents the net power injecting to or extracting from the electric network at node i. The real part $\mathrm{Re}(s_i)$ represents the so-called *active* power, while the imaginary part $\mathrm{Im}(s_i)$ represents the *reactive* power. The *apparent* power is defined as the magnitude $|s_i| = \sqrt{(\mathrm{Re}(s_i))^2 + (\mathrm{Im}(s_i))^2}$ of s_i. For power injection (i.e., power generation), $\mathrm{Re}(s_i) \leq 0$; whereas for power extraction (i.e., power demands or loads), $\mathrm{Re}(s_i) \geq 0$. We note the sign of power injection/extraction is sometimes reversed in the power systems literature. For an inductor, $\mathrm{Im}(s_i) \geq 0$; whereas for a capacitor, $\mathrm{Im}(s_i) \leq 0$. Note that transmission lines are usually resistive or inductive, namely, $\mathrm{Re}(z_{i,j}) \geq 0$ and $\mathrm{Im}(z_{i,j}) \geq 0$. The *power factor* of a power demand s_i is defined as

$\mathtt{PF}(s_i) \triangleq \frac{\mathrm{Re}(s_i)}{|s_i|}$. As required by common power electronic standards (e.g., National Electrical Code, 2005), most appliances and equipment have a bounded power factor $\mathtt{PF}(s_i) \geq 0.8$, (roughly, $\angle s_i \leq \frac{\pi}{4}$).

1.2.2 Power Flow Model

A power flow model summarizes the state of power flows, considering Kirchhoff's current law with respect to the power injection/extraction. There are several ways of describing a power flow model.

Bus Injection Model

The *Bus Injection Model* (BIM) considers the power injection (or extraction), s_j, at each node (i.e., bus) $j \in \mathcal{V}^+$:

$$s_j = \sum_{(i,j)\in\mathcal{E}} V_j I_{i,j}^* - \sum_{(j,l)\in\mathcal{E}} V_j I_{j,l}^*, \qquad \forall j \in \mathcal{V}, \tag{1.5}$$

$$V_i - V_j = z_{i,j} I_{i,j}, \qquad \forall (i,j) \in \mathcal{E}. \tag{1.6}$$

Branch Flow Model

Alternatively, the *Branch Flow Model* (BFM) (Baran and Wu, 1989a; Baran and Wu, 1989b) considers the transmitted power ($S_{i,j}$) through each edge $(i,j) \in \mathcal{E}$:

$$s_j = \sum_{(i,j)\in\mathcal{E}} \left(S_{i,j} - z_{i,j}|I_{i,j}|^2\right) - \sum_{(j,l)\in\mathcal{E}} S_{j,l}, \qquad \forall j \in \mathcal{V}, \tag{1.7}$$

$$V_i - V_j = z_{i,j} I_{i,j}, \qquad \forall (i,j) \in \mathcal{E}, \tag{1.8}$$

$$S_{i,j} = V_i I_{i,j}^*, \qquad \forall (i,j) \in \mathcal{E}. \tag{1.9}$$

For completeness, set $s_0 = -\sum_{(0,i)\in\mathcal{E}} S_{0,i}$. Note that the power flows are interpreted as from node 0 toward the rest of nodes[2] in $\mathcal{V}^+$.

The Branch Flow Model provides a convenient way to simplify the notations. One can drop the phase angles, and replace $V_i = |V_i|e^{\angle V_i}$ and

[2]BFM can be also expressed using the opposite orientation toward node 0: $s_j = \sum_{(l,j)\in\mathcal{E}} \left(S_{l,j} - z_{l,j}|I_{l,j}|^2\right) - \sum_{(j,i)\in\mathcal{E}} S_{j,i}$. As shown in Low (2014a), there is a bijection between the models of the two orientations, since $S_{j,i} = -S_{i,j} + z_{i,j}|I_{i,j}|^2$ and $I_{i,j} = -I_{j,i}$.

$I_{i,j} = |I_{i,j}|e^{\angle I_{i,j}}$ by simply $|V_i|$ and $|I_{i,j}|$, respectively. This gives us a relaxed model as follows.

Branch Flow Model with Angle Relaxation

Let $v_i = |V_i|^2$ and $\ell_{i,j} = |I_{i,j}|^2$. The *Branch Flow Model with angle relaxation* omits the phase angles:

$$s_j = \sum_{(i,j)\in\mathcal{E}} (S_{i,j} - z_{i,j}\ell_{i,j}) - \sum_{(j,l)\in\mathcal{E}} S_{j,l}, \qquad \forall j \in \mathcal{V}, \tag{1.10}$$

$$v_i - v_j = 2\mathrm{Re}(z_{i,j}^* S_{i,j}) - |z_{i,j}|^2 \ell_{i,j}, \qquad \forall (i,j) \in \mathcal{E}, \tag{1.11}$$

$$|S_{i,j}|^2 = v_i \ell_{i,j}, \qquad \forall (i,j) \in \mathcal{E}. \tag{1.12}$$

BFM with angle relaxation can be derived from BIM as follows. We rewrite (1.3) by taking the complex conjugate of both sides:

$$I_{i,j} = \frac{S_{i,j}^*}{V_i^*} \Rightarrow \; \ell_{i,j} = |I_{i,j}|^2 = \frac{|S_{i,j}|^2}{|V_i|^2} = \frac{|S_{i,j}|^2}{v_i}, \tag{1.13}$$

which is equivalent to (1.12). Substituting (1.9) in (1.8), we obtain

$$V_j = V_i - I_{i,j} z_{i,j} = V_i - \frac{S_{i,j}^*}{V_i^*} z_{i,j}. \tag{1.14}$$

Taking the magnitude square of both sides in (1.14), and using (1.13)[3]:

$$\begin{aligned} v_j = |V_j|^2 &= |V_i|^2 + |\tfrac{S_{i,j}^*}{V_i^*} z_{i,j}|^2 - 2\mathrm{Re}(V_i^* \tfrac{S_{i,j}^*}{V_i^*} z_{i,j}) \\ &= v_i^2 + \ell_{i,j} |z_{i,j}|^2 - 2\mathrm{Re}(z_{i,j}^* S_{i,j}), \end{aligned} \tag{1.15}$$

which is equivalent to (1.12).

As shown in Farivar and Low (2013a) and Farivar and Low (2013b), it is always possible to recover $(V_i, I_{i,j})_{(i,j)\in\mathcal{E}}$ from $(v_i, \ell_{i,j})_{(i,j)\in\mathcal{E}}$, when $\mathcal{G}$ is a *tree* network.

In the rest of monograph, unless otherwise stated, we assume that $\mathcal{G}$ is a tree network, and hence, we will use BFM with angle relaxation (or simply called Branch Flow Model) for brevity.

---

[3] Using the relation $|a + b|^2 = (a + b)^*(a + b) = |a|^2 + |b|^2 + a^*b + b^*a = |a|^2 + |b|^2 + 2\mathrm{Re}(a^*b) = |a|^2 + |b|^2 + 2\mathrm{Re}(b^*a)$, for complex numbers $a, b \in \mathbb{C}$.

Simplified DistFlow Model

In BFM with angle relaxation, if we assume $z_{i,j}\ell_{i,j} \to 0$, for example, because of negligible $z_{i,j}$ at each edge, then we obtain a simplified model called *DistFlow* model:

$$s_j = \sum_{(i,j)\in\mathcal{E}} S_{i,j} - \sum_{(j,l)\in\mathcal{E}} S_{j,l}, \qquad \forall j \in \mathcal{V}, \tag{1.16}$$

$$v_i - v_j = 2\mathrm{Re}(z^*_{i,j} S_{i,j}), \qquad \forall (i,j) \in \mathcal{E}, \tag{1.17}$$

$$|S_{i,j}|^2 = v_i \ell_{i,j}, \qquad \forall (i,j) \in \mathcal{E}. \tag{1.18}$$

The DistFlow model provides an "upper bound" for the power flow in BFM, because it ignores the power consumed on transmission lines.

1.2.3 Optimal Power Flow Problem

The *optimal power flow* (OPF) problem is a fundamental problem in power systems engineering, which was introduced in 1962 (Carpentier, 1962; Carpentier, 1979), and since then has received considerable attention (see Frank *et al.* (2012a) and Frank *et al.* (2012b) for a survey).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a radial (tree) electric distribution network. Node 0 is called the *root*. Since $\mathcal{G}$ is a tree, $|\mathcal{V}^+| = |\mathcal{E}| = m$. We consider a particular tree topology in which a single feeder is attached to the root 0, via a single edge $(0, 1)$. See an illustration in Fig. 1.2. Hence (1.10) in BFM (with angle relaxation) becomes

$$S_{i,j} = s_j + z_{i,j}\ell_{i,j} + \sum_{(j,l)\in\mathcal{E}} S_{j,l}, \qquad \forall (i,j) \in \mathcal{E}, \tag{1.19}$$

$$S_{0,1} = -s_0. \tag{1.20}$$

Control Variables

Instead of assigning a single power injection/extraction to each node, we consider a general setting where a set of users are attached to each node. We assume that the power demand of each user can be controlled individually. Let $\mathcal{N} = [n] \triangleq \{1, \ldots, n\}$ be the set of all users, where $|\mathcal{N}| = n$. Denote the set of users attached node j by $\mathcal{U}_j \subseteq \mathcal{N}$. Let

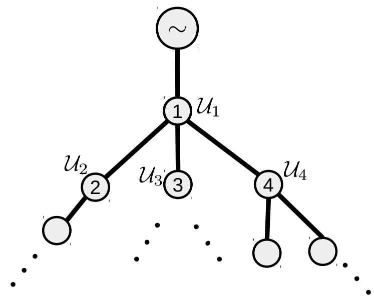


Figure 1.2: An illustration of the considered tree topology.

$\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ be the subtree rooted at node i. Let the set of users within subtree $\mathcal{G}_j$ be $\mathcal{N}_j \triangleq \cup_{j \in \mathcal{V}_j} \mathcal{U}_j$.

By a slight abuse of notation, the demand for user k is represented by $s_k \in \mathbb{C}$. In this monograph, we consider only consumer users, such that $\mathrm{Re}(s_k) \geq 0$ (but $\mathrm{Im}(s_k)$ may be negative) $\forall k \in \mathcal{N}$. Hence, it follows that the total power injection $\mathrm{Re}(s_0) \leq 0$.

Among the users, some have *discrete* power demands, denoted by $\mathcal{I} \subseteq \mathcal{N}$. A discrete demand s_k, for $k \in \mathcal{I}$, takes values from a discrete set $\mathcal{S}_k \subseteq \mathbb{C}$. We assume that $\mathbf{0} \in \mathcal{S}_k$, for all $k \in \mathcal{I}$, so that a discrete demand can be completely shut off. A special case is the binary case $\mathcal{S}_k \triangleq \{\mathbf{0}, \overline{s}_k\}$, where a demand s_k can be either completely satisfied at level $\overline{s}_k \in \mathbb{C}$ or dropped, e.g., a piece of equipment that is either switched on with a fixed power consumption rate or completely off.

The rest of the users, denoted by $\mathcal{F} \triangleq \mathcal{N} \backslash \mathcal{I}$, have *continuous* demands, defined by *convex* sets $\mathcal{S}_k$, for $k \in \mathcal{F}$; a typical example is a set defined by box constraints: $\mathcal{S}_k \triangleq \{s_k \in \mathbb{C} : \ \underline{s}_k \leq s_k \leq \overline{s}_k\}$, for given lower and upper bounds $\underline{s}_k$ and $\overline{s}_k$.

Operating Constraints of Power Systems

Recall that $S_{i,j}$ is the power flowing from node i toward j. Note that $S_{i,j}$ is not symmetric, namely, $|S_{i,j}|$ is not equivalent to $|S_{j,i}|$, the power flowing in the opposite direction. There are the following common operating constraints of power systems:

- *Power Generation Constraint*: $|s_0| \le \overline{s}_0$.
- *Power Capacity Constraints*: $|S_{i,j}| \le \overline{S}_{i,j}$, $|S_{j,i}| \le \overline{S}_{i,j}$, $\forall (i,j) \in \mathcal{E}$.
- *Current Thermal Constraints*: $\ell_{i,j} \le \overline{\ell}_{i,j}$, $\forall (i,j) \in \mathcal{E}$.
- *Voltage Constraints*: $\underline{v}_j \le v_j \le \overline{v}_j$, $\forall j \in \mathcal{V}^+$.

In the above constraints, $\underline{v}_j, \overline{v}_j \in \mathbb{R}^+$ are the minimum and maximum allowable voltage magnitude squares at node j, and $\overline{S}_{i,j}, \overline{\ell}_{i,j} \in \mathbb{R}^+$ are the maximum allowable apparent power and current on edge (i,j), respectively. By (1.20), the power generation constraint is implicitly captured by power capacity constraints as $|s_0| = |S_{0,1}| \le \overline{S}_{0,1}$.

Note that reverse power constraint $|S_{j,i}| \le \overline{S}_{i,j}$ can be reformulated as $|S_{i,j} - z_{i,j}\ell_{i,j}| \le \overline{S}_{i,j}$.

Objective Functions

In the following, a subscript is omitted from a variable to denote its vector form, for example, $S \triangleq (S_{i,j})_{(i,j)\in\mathcal{E}}, \ell \triangleq (\ell_{i,j})_{(i,j)\in\mathcal{E}}$, $s \triangleq (s_k)_{k\in\mathcal{N}}, v \triangleq (v_j)_{j\in\mathcal{V}^+}$.

The goal of OPF is to find an assignment for the demand vector s that optimizes a certain *non-negative* objective function. We consider two versions of objective functions: (1) a concave objective that represents the benefit (or utility) of power flow, and (2) a convex objective that represents the cost (or disutility) of power flow.

For utility based objective, we denote the objective function by:

$$f(s_0, s) = f_0(-\mathrm{Re}(s_0)) + \sum_{k\in\mathcal{N}} f_k(\mathrm{Re}(s_k)), \tag{1.21}$$

where $f_0 : \mathbb{R}^+ \to \mathbb{R}^+$ is non-negative and *non-increasing* (note that $\mathrm{Re}(s_0) \le 0$), and f_k is non-negative and *non-decreasing* (a user's utility increases as more power is allocated to the user, while the generator's utility decreases as more power is generated).

For cost based objective, we denote the objective function by:

$$h(s_0, s) = h_0(-\mathrm{Re}(s_0)) + \sum_{k\in\mathcal{N}} h_k(\mathrm{Re}(s_k)), \tag{1.22}$$

where $h_0 : \mathbb{R}^+ \to \mathbb{R}^+$ is non-negative and *non-decreasing*, and f_k is non-negative and *non-increasing* (thus modeling the fact that each user prefers maximum demand).

Note that for finding an optimal solution, both versions are equivalent, as one can set $f_0(-\mathrm{Re}(s_0)) = C - h_0(-\mathrm{Re}(s_0))$ and $f_k(\mathrm{Re}(s_k)) = C - h_k(\mathrm{Re}(s_k))$, for $k \geq 1$, where C is a sufficiently large constant. Nonetheless, there is a significant difference in terms of finding an approximation solution. See Section 1.3.1 for details.

Problem Formulation

We formulate OPF using BFM (with angle relaxation). The goal of OPF is to maximize the utility objective function $f(s_0, s)$ (or minimize the cost objective function $h(s_0, s)$) subject to the operating constraints of power systems.

The inputs are the voltage, current and transmitted power limits $\Big[v_0, (\underline{v}_j, \overline{v}_j)_{j\in\mathcal{V}^+}, (\overline{S}_{i,j}, \overline{\ell}_{i,j}, z_{i,j})_{(i,j)\in\mathcal{E}}, (\mathcal{S}_k)_{k\in\mathcal{N}}\Big]$, whereas the outputs are the control decision variables and power flow states, (s_0, s, S, v).

The maximization version of OPF is defined by the mixed integer programming problem (OPF) with Cons. (1.23)-(1.31). To define the minimization version of OPF (denoted by $\mathrm{OPF}_{\min}$), one replaces $\max_{s_0,s,S,v,\ell} f(s_0, s)$ by $\min_{s_0,s,S,v,\ell} h(s_0, s)$.

Note that there are two sources of *non-convexity* in this formulation: the quadratic equality constraints (1.23) and the discrete constraints for $k \in \mathcal{I}$ in (1.30).

1.3 Basics of Combinatorial Optimization

This monograph employs combinatorial optimization techniques to provide efficient approximation algorithms for AC electric power systems with discrete demands. The area of approximation algorithms is well-studied in theoretical computer science (see, e.g., Vazirani, 2010). In the following, we recall some standard terminology from this area.

$$
\begin{aligned}
\text{(OPF)} \quad & \max_{s_0, s, S, v, \ell} \ f(s_0, s) & & \\
\text{subject to} \quad & \ell_{i,j} = \frac{|S_{i,j}|^2}{v_i}, & \forall (i,j) \in \mathcal{E}, & \quad (1.23)\\
& S_{i,j} = \sum_{k \in \mathcal{U}_j} s_k + \sum_{l:(j,l) \in \mathcal{E}} S_{j,l} + z_{i,j}\ell_{i,j}, & \forall (i,j) \in \mathcal{E}, & \quad (1.24)\\
& S_{0,1} = -s_0, & & \quad (1.25)\\
& v_j = v_i + |z_{i,j}|^2 \ell_{i,j} - 2\mathrm{Re}(z_{i,j}^* S_{i,j}), & \forall (i,j) \in \mathcal{E}, & \quad (1.26)\\
& \underline{v}_j \le v_j \le \overline{v}_j, & \forall j \in \mathcal{V}^+, & \quad (1.27)\\
& |S_{i,j}| \le \overline{S}_{i,j}, \ |S_{j,i}| \le \overline{S}_{i,j}, & \forall (i,j) \in \mathcal{E}, & \quad (1.28)\\
& \ell_{i,j} \le \overline{\ell}_{i,j}, & \forall (i,j) \in \mathcal{E}, & \quad (1.29)\\
& s_k \in \mathcal{S}_k, & \forall k \in \mathcal{N}, & \quad (1.30)\\
& v_j \in \mathbb{R}^+, \forall j \in \mathcal{V}^+, \ell_{i,j} \in \mathbb{R}^+, S_{i,j} \in \mathbb{C}, & \forall (i,j) \in \mathcal{E}. & \quad (1.31)
\end{aligned}
$$

1.3.1 Approximation Solutions

Consider a maximization problem $\mathcal{A}$ with *non-negative* objective function $f(\cdot)$, let F be a feasible solution to $\mathcal{A}$ and $F^\star$ be an optimal solution to $\mathcal{A}$. $f(F)$ refers to the objective value of F. Let $\textsc{Opt}(\mathcal{A}) = f(F^\star)$ be the objective value of $F^\star$. It is common to measure the quality of a proposed feasible solution F by the approximation ratio α between the objective of this solution and that of an optimal solution $F^\star$.

Definition 1.1. For $\alpha \in (0, 1)$, an α-approximation to maximization problem $\mathcal{A}$ is a feasible solution F such that

$$f(F) \ge \alpha \cdot \textsc{Opt}(\mathcal{A}).$$

A (polynomial-time) algorithm that, for any given instance of the problem, produces a feasible solution achieving this ratio is called an α-approximation algorithm.

Similarly, for a minimization problem $\mathcal{B}$ with non-negative cost function $h(\cdot)$, let H be a feasible solution to and $H^\star$ be an optimal solution to $\mathcal{B}$. $h(H)$ refers to the cost of H. Let $\textsc{Opt}(\mathcal{B}) = h(H^\star)$ be the cost of $H^\star$.

Definition 1.2. For $\alpha' > 1$, an α'-approximation to minimization problem $\mathcal{B}$ is a feasible solution H such that

$$c(H) \leq \alpha' \cdot \textsc{Opt}(\mathcal{B}).$$

A (polynomial-time) algorithm that, for any given instance of the problem, produces a feasible solution achieving this ratio is called an α'-approximation algorithm.

Note that given a minimization problem $\mathcal{B}$, one can define a maximization problem $\mathcal{A}$, by setting $f(\cdot) = C - h(\cdot)$, for some constant C such that $f(\cdot)$ is non-negative. Although both problems are equivalent in the sense of finding an optimal solution, algorithms for finding α-approximation solutions may be very different in the two cases. In combinatorial optimization, there are numerous such examples of minimization and maximization versions of the same problems having completely different approximation algorithms and approximation ratios. One example is the minimum and maximum *traveling salesman* problems (see, e.g., Vazirani, 2010).

1.3.2 Resource-augmented Approximation Solutions

A more relaxed definition of an approximation solution is (α, β)-approximation, which also allows violation of certain constraints, parametrized by β. Consider a maximization problem $\mathcal{A}$ with a multivariate constraint function $g(\cdot)$. Suppose that a feasible solution F to $\mathcal{A}$ is required to satisfy $\underline{g} \leq g(F) \leq \overline{g}$.

Definition 1.3. For $\alpha \in (0, 1)$ and $\beta \geq 1$, an (α, β)-approximation solution to maximization problem $\mathcal{A}$ is a solution F such that

$$f(F) \geq \alpha \cdot \textsc{Opt}(\mathcal{A}),$$
$$\frac{1}{\beta} \cdot \underline{g} \leq g(F) \leq \beta \cdot \overline{g}.$$

A (polynomial-time) algorithm that, for any given instance of the problem, produces an (α, β)-approximation solution is called an (α, β)-approximation algorithm.

Definition 1.4. For $\alpha' > 1$ and $\beta \geq 1$, an (α', β)-approximation solution to minimization problem $\mathcal{B}$ is a solution H such that

$$h(H) \leq \alpha' \cdot \textsc{Opt}(\mathcal{B}),$$
$$\tfrac{1}{\beta} \cdot \underline{g} \leq g(H) \leq \beta \cdot \overline{g}.$$

A (polynomial-time) algorithm that, for any given instance of the problem, produces an (α, β)-approximation solution is called an (α, β)-approximation algorithm.

Note that α-approximation is $(\alpha, 1)$-approximation.

1.3.3 Polynomial-time Approximation Scheme (PTAS)

In particular, a *polynomial-time approximation scheme* (PTAS) is a $(1-\epsilon)$-approximation algorithm to a maximization problem, or a $(1+\epsilon)$-approximation algorithm to a minimization problem, for any $\epsilon > 0$. The running time of a PTAS is polynomial in the input size for every fixed $\epsilon > 0$, but the exponent of the polynomial might depend on $1/\epsilon$. Namely, a PTAS allows a parametrized approximation ratio in the running time.

A resource-augmented PTAS is a $(1 - \epsilon, 1 + \epsilon)$-approximation algorithm for a maximization problem, and a $(1 + \epsilon, 1 + \epsilon)$-approximation algorithm for a minimization problem, for any $\epsilon > 0$. Again the running time of such a PTAS is polynomial in the input size for every fixed $\epsilon > 0$.

1.3.4 Fully Polynomial-time Approximation Scheme (FPTAS)

An even stronger notion is a *fully polynomial-time approximation scheme* (FPTAS), which is the same as a PTAS but requires the running time to be polynomial in both input size and $1/\epsilon$.

Similarly, we define a resource augmented FPTAS, as a $(1-\epsilon, 1+\epsilon)$-approximation algorithm for a maximization problem, and a $(1+\epsilon, 1+\epsilon)$-approximation algorithm for a minimization problem, for any $\epsilon > 0$, with the running time being polynomial in the input size and $1/\epsilon$. We will refer to these as $(1 - \epsilon, 1 + \epsilon)$-FPTAS and $(1 + \epsilon, 1 + \epsilon)$-FPTAS, respectively.

1.3.5 Quasi Polynomial-time Approximation Scheme (QPTAS)

A weaker notion of a PTAS is a *quasi-polynomial-time approximation scheme* (QPTAS), which has time complexity $n^{\text{polylog}(n)}$ for each fixed $\epsilon > 0$, where n is the input size.

The notions of α-approximation, (α, β)-approximation, PTAS, FPTAS and QPTAS can be applied to OPF.

1.3.6 Polytopes and Linear Programming

A *convex polytope* $\mathcal{P}$ in $\mathbb{R}^n$ is the set of points satisfying a finite number of linear inequalities: $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \le b\}$, for a given matrix $A \in \mathbb{R}^{m\times n}$ and vector $b \in \mathbb{R}^m$. Given a set of points $P = \{p_1, \ldots, p_r\} \subseteq \mathbb{R}^n$, the *convex hull* of P, denoted by cvxhull(P) is the set of all *convex combinations* of points in P:

$$\text{cvxhull}(P) \triangleq \Big\{ \sum_{i=1}^{r} \lambda_i p_i \ \mid \ \sum_{i=1}^{r} \lambda_i = 1,\ \lambda_i \ge 0\ \forall i \Big\}. \tag{1.32}$$

By the well-known Minkowski-Weyl theorem (see, e.g., Schrijver, 1986), any polytope $\mathcal{P} \subseteq \mathbb{R}^n$ can be represented as the *convex hull* of the set of its *extreme points*, also called *vertices* or *basic feasible solutions* (BFSs) of $\mathcal{P}$.

Linear programming (LP) is the problem of maximizing or minimizing a *linear* objective function subject to *linear* constraints. Linear programs (LPs) can be solved efficiently (in polynomial-time assuming rational input of finite precision), see Bertsimas and Tsitsiklis (1997) for an introduction to LP.

The following lemma will be used in our approximation algorithms.

Lemma 1.1 (see, e.g., Bertsimas and Tsitsiklis, 1997; Schrijver, 1986)**.** Consider the following LP:

$$\text{(LP)} \quad \max_{x\in[0,1]^n} \ c^T x \tag{1.33}$$

$$\text{subject to} \quad Ax \le b, \tag{1.34}$$

where A is an $m \times n$ matrix and b is an m-dimensional vector. Then

1. there is an optimal basic feasible solution;

2. any basic feasible solution $x^\star$ has at most m *fractional* components. Namely, $\left|\left\{i \in \{1, ..., n\} \mid x^\star_i \in (0,1)\right\}\right| \le m$.

1.3.7 Second Order Cone Programming

A *Second-order cone program* (SOCP) is a convex optimization problem in which a *linear* objective function is maximized or minimized subject to ℓ_2*-norm* constraints of the following form:

$$\text{(SOCP)} \quad \max_{x \in \mathbb{R}^n} \; c^T x \tag{1.35}$$

$$\text{subject to} \quad \|A_i x + b_i\|_2 \le d_i^T x + f_i, \qquad \forall i \in \{1, ..., m\}, \tag{1.36}$$

where $A_i \in \mathbb{R}^{n_i \times n}$, $b_i \in \mathbb{R}^{n_i}$, $c, d_i \in \mathbb{R}^n$ and $f_i \in \mathbb{R}$.

There are also efficient polynomial-time algorithms for solving (approximately) SOCPs; (see, e.g., Boyd and Vandenberghe, 2004). In fact, such algorithms can find a near-feasible solution x' that satisfies the constraints within an absolute error $\delta > 0$ (that is, $\|A_i x + b_i\|_2 \le d_i^T x + f_i + \delta$), such that $c^T x' \ge \text{OPT}^\star - \delta$, in polynomial time in the input size (including the bit complexity) and $\log \frac{1}{\delta}$, where $\text{OPT}^\star$ is the optimal objective value of (SOCP).

In many cases, it is possible to convert such an approximately feasible solution x' to an exactly feasible solution without losing much in the approximation guarantee; see, for example, Section 2.3.1. For simplicity in this monograph, unless otherwise stated, we will assume that the convex programming solver returns an *exact* optimal solution.

1.4 Organization

This monograph covers approximation algorithms and inapproximability results for various settings of AC electric systems in the following chapters:

- (Chapter 2) Basic single-capacitated AC electric power systems to establish the foundation of a more sophisticated electric grid.

- (Chapter 3) Constant-sized AC electric grid networks with power flows and common operating constraints of power systems.
- (Chapter 4) Scheduling of AC electric power that involves temporal optimization with heterogeneous users' preferences.

Moreover, we provide hardness results in Chapter 5 for the above settings to show that our approximation algorithms are among the best achievable in theory. We also provide simulation studies of our algorithms in several practical case studies in Chapter 6. Finally, we conclude this monograph with an outline of several on-going extensions and future work in Chapter 7.

1.5 Notes

The optimal power flow (OPF) problem was introduced in 1962 (Carpentier, 1962; Carpentier, 1979), and since then has been studied extensively (see Frank *et al.* (2012a) and Frank *et al.* (2012b) for a survey). There are several formulations of OPF, with subtle differences. For example, Gan *et al.* (2015) and Huang *et al.* (2017) adopt the opposite flow orientation from leaves to root. Also, Huang *et al.* (2017) implicitly considers power capacity constraints in one direction only. Our formulation explicitly considers bi-directional power capacity constraints. Although Gan *et al.* (2015) considers the possibility of discrete power injections, it provides efficient algorithm for finding the optimal solutions only in the *continuous* case, under some assumptions. For the minimization version of OPF, Gan *et al.* (2015) and Huang *et al.* (2017) consider only non-increasing objective functions for the exactness of convex relaxation. However, convex objective function is required for solving OPF.

2

Single-capacitated AC Electric Power Systems

Undoubtedly, OPF is a hard problem. Rather than tackling such a hard problem in its complete form, we first tackle a simplified setting, considering a single power capacity constraint, called *Complex-demand Knapsack Problem* (CKP). For simplicity in this chapter, we only consider a *discrete* set of demands $\mathcal{N} = \mathcal{I}$ and assume the *binary* case. That is, we assume $|\mathcal{I}| = n$, and $\mathcal{S}_k = \{0, \overline{s}_k\}$ for all $k \in \mathcal{I}$.

2.1 Preliminaries of the Knapsack Problem

We define the Complex-demand Knapsack Problem (CKP) as follows.

$$
\begin{aligned}
\text{(CKP)} \quad & \max_{x} \sum_{k \in \mathcal{I}} u_k x_k \\
\text{subject to} \quad & \Big| \sum_{k \in \mathcal{I}} \overline{s}_k x_k \Big| \le \overline{S}, & (2.1) \\
& x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}. & (2.2)
\end{aligned}
$$

In essence, CKP is a one-link setting of OPF, with negligible line impedance and the absence of voltage constraints. The objective function is $u(x) \triangleq \sum_k u_k x_k$, where $u_k \in \mathbb{R}^+$ is the respective utility of user k, if her/his demand $\overline{s}_k$ is completely satisfied.

For $X \subseteq \mathcal{I}$, let $u(X) \triangleq \sum_{k \in X} u_k$ and $s(X) \triangleq \sum_{k \in X} \overline{s}_k$. CKP can be reformulated as $\max_X u(X)$ subject to $|s(X)| \leq \overline{S}$.

CKP is related to the well-known class of *knapsack* problems. First, define the 1-Dimensional Knapsack Problem (1-KP) as follows.

$$
\begin{aligned}
\text{(1-KP)} \quad & \max_x \sum_{k \in \mathcal{I}} u_k x_k \\
\text{subject to} \quad & \sum_{k \in \mathcal{I}} \hat{s}_k x_k \leq \overline{S}, && (2.3) \\
& x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}, && (2.4)
\end{aligned}
$$

where $\hat{s}_k \in \mathbb{R}^+$ is a positive real number for each $k \in \mathcal{I}$.

Evidently, 1-KP is a subproblem of CKP, when we set $\mathrm{Im}(\overline{s}_k) = 0$ for all k. 1-KP is known to be NP-complete, and so is CKP.

1-KP can be generalized to be *multi-dimensional*. Define the m-Dimensional Knapsack Problem (m-KP) as follows.

$$
\begin{aligned}
(m\text{-KP}) \quad & \max_x \sum_{k \in \mathcal{I}} u_k x_k \\
\text{subject to} \quad & \sum_{k \in \mathcal{I}} \hat{s}_k^r x_k \leq \overline{S}^r, \ \forall r \in \{1, ..., m\}, && (2.5) \\
& x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}, && (2.6)
\end{aligned}
$$

where $\overline{S}^r$ is the capacity for the r-th dimension. Each m-KP is a linear integer program, and m-KP is a special case of $(m+1)$-KP for all m.

We note that there is an FPTAS for 1-KP. On the other hand, for $m \geq 2$, there is no FPTAS for m-KP unless P=NP, but there is a PTAS for every fixed m. See Kellerer *et al.* (2010) for a comprehensive treatment of knapsack problems.

Next, we also define the minimization version of CKP as follows.

$$
\begin{aligned}
(\text{CKP}_{\min}) \quad & \min_x \sum_{k \in \mathcal{I}} c_k (1 - x_k) \\
\text{subject to} \quad & \Big| \sum_{k \in \mathcal{I}} \overline{s}_k x_k \Big| \leq \overline{S}, && (2.7) \\
& x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}. && (2.8)
\end{aligned}
$$

In this chapter, we will provide approximation algorithms for CKP and $\text{CKP}_{\min}$.

We note that CKP is *rotationally invariant*, that is, the problem remains the same when the arguments of all demands are rotated by the same angle. Thus, we may, without loss of generality, assume that one of the demands, say $\overline{s}_1$, is aligned along the positive real axis, and define a class of sub-problems for CKP, by restricting the maximum phase angle (i.e., the argument) that any other demand makes with $\overline{s}_1$. In particular, we will write CKP$[\phi_1, \phi_2]$ for the restriction of problem CKP subject to $\phi_1 \leq \max_{k\in\mathcal{I}} \angle\overline{s}_k \leq \phi_2$, where $\angle\overline{s}_k \in [0, \pi]$ is the angle that $\overline{s}_k$ makes with $\overline{s}_1$. We remark that in realistic settings of power systems, the active power demand is positive (i.e., $\overline{s}_k^{\mathrm{R}} \geq 0$), but the power factor (i.e., $\frac{\overline{s}_k^{\mathrm{R}}}{|\overline{s}_k|}$) is bounded by a certain threshold (see, e.g., National Electrical Code, 2005), which is equivalent to restricting the arguments of complex-valued demands within a certain range.

From a computational point of view, we will need to specify how the inputs are described. Throughout this monograph, we will assume that each of the demands is given by its real and imaginary components, represented as rational numbers.

2.2 Greedy Approximation Algorithm

In this section, we present a greedy algorithm for (maximizing) CKP$[0, \frac{\pi}{2}]$, which is based on a similar greedy algorithm for 1-KP. The basic idea is to first sort the demands in a *non-increasing* order, according to the *efficiency ratio* defined by $\frac{u_k}{|\overline{s}_k|}$. Then we pack the demands greedily in this order whenever feasible without exceeding the capacity $\overline{S}$. Call such a solution X. Then we find the single demand with the highest utility ($\max_{k\in\mathcal{I}}\{u_k\}$), breaking ties arbitrarily. Lastly, we output the solution with the higher utility between X and the highest utility demand. This naturally generalizes the greedy algorithm for 1-KP, which only considers each $\overline{s}_k$ as a real number. The greedy algorithm is called `GreedyRatio`, which is presented in Algorithm 1.

Evidently, `GreedyRatio` outputs a feasible solution in $O(n \log n)$ time. Let $\phi = \max_{k\in\mathcal{I}}\{\angle\overline{s}_k\}$ be the maximum phase angle that a demand makes with the real axis. More precisely, `GreedyRatio` provides a solution that in the worst possible case is at least $\frac{1}{2}\cos\frac{\phi}{2}$- factor of the

Algorithm 1 `GreedyRatio`$[(u_k, \overline{s}_k)_{k\in\mathcal{I}}, \overline{S}]$

Sort $\mathcal{I}$ by the efficiency ratios, such that

$$\frac{u_1}{|\overline{s}_1|} \geq \frac{u_2}{|\overline{s}_2|} \geq ... \geq \frac{u_n}{|\overline{s}_n|} \tag{2.9}$$

$X \leftarrow \varnothing$
for $k \in \mathcal{I}$ **do**
  if $\left|\sum_{k'\in X} \overline{s}_{k'} + \overline{s}_k\right| \leq \overline{S}$ **then**
    $X \leftarrow X \cup \{k\}$
  end if
end for
Set $j \leftarrow \arg\max_{k\in\mathcal{I}} u_k$
if $u(X) \geq u_j$ **then**
  return X
else
  return $\{j\}$
end if

optimal solution for $\text{CKP}[0, \frac{\pi}{2}]$, as described in the following theorem.

Theorem 2.1. `GreedyRatio` is a $(\frac{1}{2}\cos\frac{\phi}{2})$-approximation algorithm for $\text{CKP}[0, \frac{\pi}{2}]$ that runs in $O(n \log n)$ time.

Proof. Denote by $u^{\texttt{GR}}$ the utility of the output solution of `GreedyRatio`. Let $X^\star \subseteq \mathcal{I}$ be an optimal solution of $\text{CKP}[0, \frac{\pi}{2}]$, and the $\text{O{\scriptsize PT}} \triangleq \sum_{k\in X^\star} u_k$ be the corresponding total utility.

Consider the complex-valued demands are substituted in $\text{CKP}[0, \frac{\pi}{2}]$ by its real-valued magnitude and the binary decision variable x_k is relaxed to be non-negative real values instead of only integers values (i.e., $(x_k)_{k\in\mathcal{I}} \in [0,1]^n$), as formulated in the following linear programming problem:

$$(\text{LP1}) \qquad \max_{x_k\in[0,1]} \sum_{k\in\mathcal{I}} u_k x_k \tag{2.10}$$

$$\text{subject to} \qquad \sum_{k\in\mathcal{I}} |\overline{s}_k| x_k \leq \overline{S} \tag{2.11}$$

Denote by $X^\star_{\text{lp}} \subseteq \mathcal{I}$ an optimal solution of LP1 and by $\textsc{Opt}_L \triangleq \sum_{k \in X^\star_{\text{lp}}} u_k$ the corresponding total utility. In chapter 2 of Kellerer *et al.* (2010) it was shown that the optimal solution to LP1 problem can be determined easily, since the problem admits the greedy choice property. In other words, a global optimum of the problem can be achieved by choosing a series of locally optimal choices. The greedy choice for LP1 problem is to select demands in the sorted order defined by Eqn. (2.9). Assume at some iteration t adding the next demand to X that causes capacity constraint violation, that is

$$\sum_{k=1}^{t-1} |\overline{s}_k| \leq \overline{S} \text{ and } \sum_{k=1}^{t} |\overline{s}_k| > \overline{S}. \tag{2.12}$$

The greedy execution is stopped at this point and the remaining capacity $\overline{S} - \sum\limits_{k=1}^{t-1} |\overline{s}_k|$ is occupied by the corresponding fractional part of the t-th user's power demand. Observe that the preceding greedy strategy is the adapted version of `GreedyRatio` algorithm for the relaxed $\text{CKP}[0, \frac{\pi}{2}]$ problem with continuous decision variable $(x_k)_{k \in \mathcal{I}} \in [0,1]^n$ and complex-valued demands. Let $\hat{p} \triangleq \sum\limits_{k=1}^{t-1} u_k$ and $u_{\max} \triangleq \max_{k \in \mathcal{N}} u_k$. It was also shown in Kellerer *et al.* (2010) that

$$\textsc{Opt}_{\text{lp}} \triangleq \hat{p} + (\overline{S} - \sum_{k=1}^{t-1} |\overline{s}_k|) \frac{u_t}{|\overline{s}_t|} \leq \hat{p} + u_{\max}. \tag{2.13}$$

Evidently, $u^{\texttt{GR}} \geq \hat{p}$. This gives

$$\textsc{Opt}_{\text{lp}} \leq u^{\texttt{GR}} + u_{\max}. \tag{2.14}$$

On the other hand, by Lemma 2.2 below it follows that

$$\cos \tfrac{\phi}{2} \cdot \sum_{i \in X^\star} |\overline{s}_i| \leq \Big| \sum_{i \in X^\star} \overline{s}_i \Big| \leq \overline{S}\,, \tag{2.15}$$

since ϕ is restricted to be at most $\frac{\pi}{2}$. Note that the subset $X^\star$ becomes a feasible solution to LP1 if the relaxed decision variable is set to $x_i = \cos \frac{\phi}{2}$ for $\forall i \in X^\star$ and $x_i = 0$ otherwise. This implies that

$$\textsc{Opt}_{\text{lp}} \geq \cos \tfrac{\phi}{2} \cdot u(X^\star) = \cos \tfrac{\phi}{2} \cdot \textsc{Opt}. \tag{2.16}$$

In `GreedyRatio`, $u^{\texttt{GR}} \geq u_{\max}$, and hence by Eqns. (2.14) and (2.16) it follows that

$$u^{\texttt{GR}} \geq \frac{1}{2} \cos \tfrac{\phi}{2} \cdot \textsc{Opt}. \tag{2.17}$$

□

Lemma 2.2. Given a set of 2D vectors $\{\nu_i \in \mathbb{R}^2\}_{i=1}^n$

$$\frac{\sum_{i=1}^n |\nu_i|}{\left|\sum_{i=1}^n \nu_i\right|} \leq \sec \tfrac{\phi}{2},$$

where ϕ is the maximum angle between any pair of vectors and $0 \leq \phi \leq \frac{\pi}{2}$.

Proof. If $\phi = 0$ then the statement is trivial, therefore we assume otherwise. We prove $\frac{(\sum_{i=1}^n |\nu_i|)^2}{|\sum_{i=1}^n \nu_i|^2} \leq \frac{2}{\cos\phi + 1}$ by induction (notice that $\sec \frac{\phi}{2} = \sqrt{\frac{2}{\cos\phi+1}}$). First, we expand the left-hand side by

$$\frac{\sum_{i=1}^n |\nu_i|^2 + 2\sum_{1\leq i<j\leq n} |\nu_i|\cdot|\nu_j|}{\sum_{i=1}^n |\nu_i|^2 + 2\sum_{1\leq i<j\leq n} |\nu_i|\cdot|\nu_j|\Big(\sin(\phi_i)\sin(\phi_j) + \cos(\phi_i)\cos(\phi_j)\Big)}$$
$$= \frac{\sum_{i=1}^n |\nu_i|^2 + 2\sum_{1\leq i<j\leq n} |\nu_i|\cdot|\nu_j|}{\sum_{i=1}^n |\nu_i|^2 + 2\sum_{1\leq i<j\leq n} |\nu_i|\cdot|\nu_j|\cos(\phi_i - \phi_j)}, \tag{2.18}$$

where ϕ_i is the angle that ν_i makes with the x axis.

Consider the base case: $n = 2$. Eqn. (2.18) becomes

$$\frac{|\nu_1|^2 + |\nu_2|^2 + 2|\nu_1|\cdot|\nu_2|}{|\nu_1|^2 + |\nu_2|^2 + 2|\nu_1|\cdot|\nu_2|\cos\phi} = f\Big(\frac{|\nu_2|}{|\nu_1|}\Big), \tag{2.19}$$

where $g(x) \triangleq \frac{1+x^2+2x}{1+x^2+2x\cos\phi}$. The first derivative is given by

$$g'(x) = \frac{(1+x^2+2x\cos\phi)(2x+2) - 1 + x^2 + 2x)(2x + 2\cos\phi)}{(1+x^2+2x\cos\phi)^2} \tag{2.20}$$

Note that $g'(x)$ is zero only when $x = 1$. Hence, $g(1)$ is an extremum point. We compare $g(1)$ with $g(x)$ at the boundaries $x \in \{0, \infty\}$:

$$g(1) = \frac{2}{\cos\phi + 1} \geq g(0) = \lim_{x\to\infty} g(x) = 1 \tag{2.21}$$

Therefore, $g(x)$ has a global maximum of $\frac{2}{\cos\phi+1}$.

Next, we proceed to the inductive step. We assume

$$\frac{\sum_{i=1}^{r-1}|\nu_i|}{|\sum_{i=1}^{r-1}\nu_i|} \leq \sqrt{\frac{2}{\cos\phi+1}},$$

where $r \in \{1,\dots,n\}$. Without loss of generality, we assume $\phi_2 \geq \phi_3 \geq \cdots \geq \phi_n \geq \phi_1$. Rewrite Eqn. (2.18) as

$$\frac{(\sum_{i=1}^{r}|\nu_i|)^2}{\sum_{i=1}^{r}|\nu_i|^2 + 2\sum_{1\leq i<j<r}|\nu_i||\nu_j|\cos(\phi_i-\phi_j) + 2|\nu_r|\sum_{1\leq i<r}|\nu_i|\cos(\phi_i-\phi_r)} \tag{2.22}$$

Let $g(\phi_r)$ be the denominator of Eqn. (2.22). We take the second derivative of $g(\phi_r)$:

$$g''(\phi_r) = -2|\nu_r|\sum_{1\leq i<r}|\nu_i|\cos(\phi_i-\phi_r) \tag{2.23}$$

Notice that $\cos(\phi_i-\phi_r) \geq 0$, therefore the second derivative is always negative. This indicates that all local extrema in $[0,\phi_{r-1}]$ of $g(\phi_n)$ are local maxima. Hence, the minimum occurs at the boundaries:

$$\min_{\phi_r\in[0,\phi_{r-1}]} g(\phi_r) \in \{g(0), g(\phi_{r-1})\} \tag{2.24}$$

If $\phi_r \in \{0,\phi_r\}$, then there must exist at least a pair of vectors in $\{\nu_i\}_{i=1}^{r}$ with the same angle. Combining these two vectors into one, we can obtain an instance with $r-1$ vectors. Hence, by the inductive hypothesis, the same bound holds up to r vectors. □

2.3 PTAS

Although the greedy algorithm is fast, it cannot provide a close-to-optimal solution. In this section, we present PTAS for maximizing and minimizing CKP$[0,\frac{\pi}{2}]$. First, we consider maximizing CKP$[0,\frac{\pi}{2}]$.

2.3.1 Maximizing CKP

We provide a $(1-\epsilon)$-approximation for CKP$[0,\frac{\pi}{2}]$, called `CKP-PTAS`, which is presented in Algorithm 2.

Define a convex relaxation of CKP (denoted by rCKP), such that $x_k \in \{0,1\}$ is replaced by $x_k \in [0,1]$ for all $k \in \mathcal{I}$. We define another convex relaxation that will be used in the PTAS denoted by $\text{rCKP}[I_1, I_0]$ which is equivalent to rCKP, subject to partial substitution such that $x_k = 1$, for all $k \in I_1$ and $x_k = 0$, for all $k \in I_0$, where $I_1, I_0 \subseteq \mathcal{I}$ such that $I_1 \cap I_0 = \varnothing$:

$$(\text{rCKP}[I_1, I_0]) \quad \max_{x_k \in [0,1]} \sum_{k \in \mathcal{I}} u_k x_k \tag{2.25}$$

$$\text{subject to } \Big(\sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{R}} \cdot x_k\Big)^2 + \Big(\sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{I}} \cdot x_k\Big)^2 \le \overline{S}^2, \tag{2.26}$$

$$x_k = 1, \qquad \forall k \in I_1, \tag{2.27}$$

$$x_k = 0, \qquad \forall k \in I_0. \tag{2.28}$$

The above relaxation is a SOCP, since Cons. (2.26) can be written as

$$\left\| \begin{pmatrix} \overline{s}_1^{\mathrm{R}} & \cdots & \overline{s}_n^{\mathrm{R}} \\ \overline{s}_1^{\mathrm{I}} & \cdots & \overline{s}_n^{\mathrm{I}} \end{pmatrix} x \right\|_2 \le \overline{S}.$$

As mentioned in Section 1.3.7, this relaxation can be solved approximately in polynomial time using standard convex programming algorithms (see e.g., Boyd and Vandenberghe, 2004). In fact, such algorithms can find a feasible solution x' to the convex relaxation such that $u(x') \ge \text{Opt}^\star - \delta$, in time polynomial in the input size (including the bit complexity) and $\log \frac{1}{\delta}$, where $\text{Opt}^\star$ is the optimal objective value of $\text{rCKP}[I_1, I_0]$. Notice that $\text{Opt}^\star \ge \text{Opt} \ge \bar{u} \triangleq \max_k u_k$, and hence setting δ to $\frac{\epsilon}{2} \cdot \bar{u}$ assures that $u(x') \ge (1 - \frac{\epsilon}{2}) \cdot \text{Opt}^\star$.

Given a feasible solution $x^\star$ to $\text{rCKP}[I_1, I_0]$, a restricted set of demands $R \subseteq \mathcal{I}$, and vectors $\overline{S}^1, \overline{S}^2 \in \mathbb{R}^{+m}$, we define the following linear programming relaxation, denoted by $\text{LP}[\overline{S}^1, \overline{S}^2, x^\star, R]$:

$$(\text{LP}[\overline{S}^1, \overline{S}^2, x^\star, R]) \quad \max_{x_k \in [0,1]} \sum_{k \in \mathcal{I}} u_k x_k \tag{2.29}$$

$$\text{subject to } \sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{R}} \cdot x_k \le \overline{S}^1, \tag{2.30}$$

$$\sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{I}} \cdot x_k \le \overline{S}^2, \tag{2.31}$$

$$x_k = x_k^\star, \qquad \forall k \in R. \tag{2.32}$$

`CKP-PTAS` proceeds as follows. We guess $I_1 \subseteq \mathcal{I}$ to be the set of *largest*-utility $\frac{4}{\epsilon}$ demands in the optimal solution; this defines an excluded set of demands $I_0 \subseteq \mathcal{I} \setminus I_1$ whose utilities exceed one of the utilities in I_1 (Step 4). For each such I_1 and I_0, we solve the convex program $\text{RCKP}[I_1, I_0]$ and obtain a $(1-\frac{\epsilon}{2})$-approximation x' (note that the feasibility of the convex program is guaranteed by the conditions in Step 3). The real and imaginary projections over all time slots of solution x', denoted by $L^{\mathrm{R}} \in \mathbb{R}_+^m$ and $L^{\mathrm{I}} \in \mathbb{R}_+^m$, are used to define the linear program $\mathrm{LP}[L^{\mathrm{R}}, L^{\mathrm{I}}, x', I_1 \cup I_0]$ over the restricted set of demands $I_1 \cup I_0$. We solve the linear program in Step 7, and then round down the solution corresponding to demands $k \in \mathcal{I}$ in Step 9. Finally, we return a solution $\hat{x}$ that attains the maximum utility among all the obtained solutions.

Algorithm 2 `CKP-PTAS`$[(u_k, \overline{s}_k)_{k\in\mathcal{I}}, \overline{S}, \epsilon]$

1: $u_{\max} \leftarrow 0$
2: **for** each set $I_1 \subseteq \mathcal{I}$ such that $|I_1| \le \frac{4}{\epsilon}$ **do**
3:     **if** $\left|\sum_{k\in I_1} \overline{s}_k\right| \le \overline{S}$ **then**
4:         $I_0 \leftarrow \{k \in \mathcal{I} \setminus I_1 \mid u_k > \min_{(k')\in I_1} u_{k'}\}$
5:         $x' \leftarrow$ (Near) optimal solution of $\text{RCKP}[I_1, I_0]$
▷ Obtain a $(1-\frac{\epsilon}{2})$-approximation
6:         $L^{\mathrm{R}} \leftarrow \sum_{k\in\mathcal{I}} \overline{s}_k^{\mathrm{R}} \cdot x'_k$; $L^{\mathrm{I}} \leftarrow \sum_{k\in\mathcal{I}} \overline{s}_k^{\mathrm{I}} \cdot x'_k$
7:         $x'' \leftarrow$ Optimal BFS of $\mathrm{LP}[L^{\mathrm{R}}, L^{\mathrm{I}}, x', I_1 \cup I_0]$
▷ Round down the LP solution
8:         **if** $u_{\max} < u((\lfloor x''_k \rfloor)_{k\in\mathcal{I}})$ **then**
9:             $\hat{x} \leftarrow (\lfloor x''_k \rfloor)_{k\in\mathcal{I}}$
10:             $u_{\max} \leftarrow u(\hat{x})$
11:         **end if**
12:     **end if**
13: **end for**
14: **return** $\hat{x}$

Theorem 2.3. For any fixed ϵ, `CKP-PTAS` obtains a $(1-\epsilon)$-approximation for $\text{CKP}[0, \frac{\pi}{2}]$ in polynomial time.

Proof. One can easily see that the running time of `CKP-PTAS` is polynomial in size of the input, for any given ϵ. We now argue that the solution

$\hat{x}$ is $(1-\epsilon)$-approximation for CKP$[0, \frac{\pi}{2}]$. Let $x^\star$ be the optimal solution for CKP$[0, \frac{\pi}{2}]$ of utility $\textsc{Opt} \triangleq u(x^\star)$. Define $I^\star \triangleq \{k \in \mathcal{I} \mid x^\star_k = 1\}$. By the feasibility of $x^\star$, in Step 5 `CKP-PTAS` obtains

$$u(x') \geq (1 - \tfrac{\epsilon}{2}) \cdot \textsc{Opt}^\star \geq (1 - \tfrac{\epsilon}{2}) \cdot \textsc{Opt}, \tag{2.33}$$

where $\textsc{Opt}^\star$ is the optimal value of rCKP$[I_1, I_0]$ for some I_1 equal to the highest $\frac{4}{\epsilon}$ utility demands in $I^\star$, and $I_0 \cap I^\star = \varnothing$. If $|I^\star| \leq \frac{4}{\epsilon}$, then obviously $\hat{x} = x'' = x'$ and $u(x') \geq (1 - \frac{\epsilon}{2})\textsc{Opt}$.

Now suppose $|I^\star| > \frac{4}{\epsilon}$. We observe x' is a feasible solution for LP$[L^{\rm R}, L^{\rm I}, x', I_1 \cup I_0]$ (namely, Cons. (2.30)-(2.32) are tight when x' is substituted). Therefore, the optimal solution x'' of LP$[L^{\rm R}, L^{\rm I}, x', I_1 \cup I_0]$ satisfies

$$u(x'') \geq u(x'). \tag{2.34}$$

By Lemma 1.1, LP$[L^{\rm R}, L^{\rm I}, x', I_1 \cup I_0]$ has an optimal basic feasible solution (BFS) with at most 2 fractional components, and for any fractional component $k \in \mathcal{I} \setminus (I_1 \cup I_0)$, $u_k \leq \min_{k' \in I_1} u_{k'} \leq \frac{\sum_{k' \in I_1} u_{k'}}{|I_1|}$. Therefore, rounding down x'' gives

$$\begin{aligned} u(\hat{x}) &\geq u(x'') - 2\frac{\sum_{k \in I_1} u_k}{|I_1|} \geq (1 - \tfrac{\epsilon}{2}) u(x'') \\ &\geq (1 - \tfrac{\epsilon}{2})^2 \cdot \textsc{Opt} \geq (1 - \epsilon) \cdot \textsc{Opt}, \end{aligned} \tag{2.35}$$

where (2.35) follows by Eqns. (2.33)-(2.34). It remains to show that $\hat{x}$ is feasible. Since $\hat{x}$ is obtained by rounding down x'',

$$\Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm R}_k \cdot \hat{x}_k\Big)^2 + \Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm I}_k \cdot \hat{x}_k\Big)^2 \tag{2.36}$$

$$\leq \Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm R}_k \cdot x''_k\Big)^2 + \Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm I}_k \cdot x''_k\Big)^2$$

$$\leq (L^{\rm R})^2 + (L^{\rm I})^2 = \Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm R}_k x'_k\Big)^2 + \Big(\sum_{k \in \mathcal{I}} \overline{s}^{\rm I}_k x'_k\Big)^2 \leq \overline{S}^2, \tag{2.37}$$

where (2.37) follow by the feasibility of x'' and x' respectively. Hence, Cons. (2.1) is satisfied. Finally, since all fractional components of x'' in Step 9 are rounded down, Cons. (2.2) are also satisfied. □

Remark 2.1. The above proof shows that we do not need to solve $\mathrm{LP}[L^{\mathrm{R}}, L^{\mathrm{I}}, x', I_1 \cup I_0]$; starting from x', we only need to get a BFS with the same (or better) objective value, which can be reduced to solving systems of linear equations (Schrijver, 1986).

2.3.2 Minimizing CKP

Lastly, we consider minimizing $\mathrm{CKP}[0, \frac{\pi}{2}]$. Define the minimizing versions of $\mathrm{rCKP}[I_1, I_0]$ and $\mathrm{LP}[\overline{S}^1, \overline{S}^2, x^\star, R]$ as follows.

$$
\begin{aligned}
(\mathrm{rCKP}_{\min}[I_1, I_0]) \quad & \min_{x_k \in [0,1]} \sum_{k \in \mathcal{I}} c_k (1 - x_k) && (2.38)\\
\text{subject to } & \Big(\sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{R}} \cdot x_k\Big)^2 + \Big(\sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{I}} \cdot x_k\Big)^2 \le \overline{S}^2, && (2.39)\\
& x_k = 1, \qquad \forall k \in I_1, && (2.40)\\
& x_k = 0, \qquad \forall k \in I_0. && (2.41)
\end{aligned}
$$

$$
\begin{aligned}
(\mathrm{LP}_{\min}[\overline{S}^1, \overline{S}^2, x^\star, R]) \quad & \min_{x_k \in [0,1]} \sum_{k \in \mathcal{I}} c_k (1 - x_k) && (2.42)\\
\text{subject to } & \sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{R}} \cdot x_k \le \overline{S}^1, && (2.43)\\
& \sum_{k \in \mathcal{I}} \overline{s}_k^{\mathrm{I}} \cdot x_k \le \overline{S}^2, && (2.44)\\
& x_k = x_k^\star, \qquad \forall k \in R.. && (2.45)
\end{aligned}
$$

The PTAS is presented in Algorithm 3. For $x \in [0,1]^n$, we write $c(x) \triangleq \sum_{k \in \mathcal{I}} c_k (1 - x_k)$.

Theorem 2.4. For any fixed $\epsilon > 0$, $\texttt{CKP-PTAS}_{\min}$ obtains a $(1+\epsilon)$-approximation for $\mathrm{CKP}_{\min}[0, \frac{\pi}{2}]$ in polynomial time.

Proof. The proof follows a similar approach as in Theorem 2.3, but via the LPs $\texttt{CKP-PTAS}_{\min}[(c_k, \overline{s}_k)_{k \in \mathcal{I}}, \overline{S}, \epsilon]$ and $(\mathrm{LP}_{\min}[\overline{S}^1, \overline{S}^2, x^\star, R])$. As before, we guess $I_1 \subseteq \mathcal{I}$ to be the set of *largest*-cost $\frac{4}{\epsilon}$ demands in the optimal solution; this defines an excluded set of demands $I_0 \subseteq \mathcal{I} \setminus I_1$

Algorithm 3 $\texttt{CKP-PTAS}_{\min}[(c_k, \overline{s}_k)_{k\in\mathcal{I}}, \overline{S}, \epsilon]$

1: $c_{\min} \leftarrow \infty$
2: **for** each set $I_1 \subseteq \mathcal{I}$ such that $|I_1| \le \frac{4}{\epsilon}$ **do**
3:     **if** $\left|\sum_{k\in I_1} \overline{s}_k\right| \le \overline{S}$ **then**
4:         $I_1 \leftarrow \{k \in \mathcal{I} \setminus I_0 \mid c_k > \min_{(k')\in I_1} c_{k'}\}$
5:         $x' \leftarrow$ (Near) optimal solution of $\textsc{r}\mathrm{CKP}_{\min}[I_1, I_0]$
                $\triangleright$ Obtain a $(1+\frac{\epsilon}{4})$-approximation
6:         $L^{\mathrm{R}} \leftarrow \sum_{k\in\mathcal{I}} \overline{s}^{\mathrm{R}}_k \cdot x'_k$; $L^{\mathrm{I}} \leftarrow \sum_{k\in\mathcal{I}} \overline{s}^{\mathrm{I}}_k \cdot x'_k$
7:         $x'' \leftarrow$ Basic feasible solution of $\mathrm{LP}[L^{\mathrm{R}}, L^{\mathrm{I}}, x', I_1 \cup I_0]$
                $\triangleright$ Round down the LP solution
8:         **if** $c_{\min} > c\big((\lfloor x''_k \rfloor)_{k\in\mathcal{I}}\big)$ **then**
9:             $\hat{x} \leftarrow (\lfloor x''_k \rfloor)_{k\in\mathcal{I}}$
10:             $c_{\min} \leftarrow c(\hat{x})$
11:         **end if**
12:     **end if**
13: **end for**
14: **return** $\hat{x}$

whose utilities exceed one of the utilities in I_1 (Step 4). Then we proceed as in `CKP-PTAS`. As in (2.35), rounding down x'' in gives

$$\begin{aligned} c(\hat{x}) &\le c(x'') + 2\frac{\sum_{k\in I_1} c_k}{|I_1|} \le (1+\tfrac{\epsilon}{2})c(x'') \\ &\le (1+\tfrac{\epsilon}{2})(1+\tfrac{\epsilon}{4}) \cdot \textsc{Opt} \le (1+\epsilon) \cdot \textsc{Opt}, \end{aligned}$$

where the third inequality follows from the definition of x' in Step 5 of the algorithm. □

2.4 Resource-augmented FPTAS

In the previous section, we have restricted our attention to the setting where all demands lie in the positive quadrant in the complex plane (i.e., $\mathrm{CKP}[0, \frac{\pi}{2}]$). In this section, we extend our study to the second quadrant ($\mathrm{CKP}[0, \pi-\varepsilon]$) for any $\varepsilon > 0$, that is, we assume $\angle\overline{s}_k \le \pi - \varepsilon$ for all $k \in \mathcal{I}$.

We note that Chapter 5 shows that CKP$[0, \pi]$ is inapproximable and there is no $(\alpha, 1)$-approximation for CKP$[0, \pi - \varepsilon]$. Therefore, we can at best obtain a resource-augmented approximation, and furthermore the running time should depend on the maximum angle $\phi \triangleq \max_k \angle \overline{s}_k$. For convenience, we let $\theta = \max\{\phi - \frac{\pi}{2}, 0\}$. (see Fig. 2.1 for an illustration).

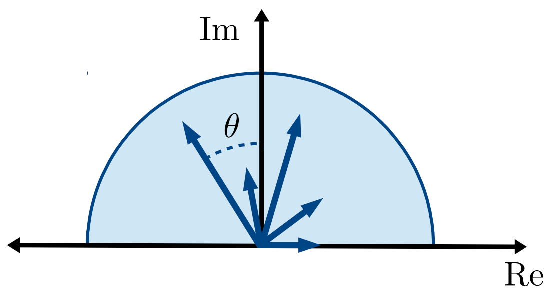


Figure 2.1: All demands lie in the shaded area. We measure $\theta = \phi - \frac{\pi}{2}$ from the imaginary axis.

We present a $(1, 1+\epsilon)$-approximation for CKP$[0, \pi-\varepsilon]$ in `CKP-bFPTAS` that is polynomial in both $\frac{1}{\epsilon}$ and n (i.e., FPTAS). We assume that $\tan\theta$ is bounded polynomial $P(n) \geq 1$ in n. As we shall also see in Chapter 5, without such an assumption, a resource-augmented FPTAS is unlikely to exist.

Let $\mathcal{I}_+ \triangleq \{k \in \mathcal{I} \mid \overline{s}_k^{\mathrm{R}} \geq 0\}$ and $\mathcal{I}_- \triangleq \{k \in \mathcal{I} \mid \overline{s}_k^{\mathrm{R}} < 0\}$ be the subsets of users with demands in the first and second quadrants respectively. Consider any solution $X \subseteq \mathcal{I}$ to CKP$[0, \pi - \varepsilon]$ and define $X_+ \triangleq \{k \mid \overline{s}_k^{\mathrm{R}} \geq 0, k \in X\}$ and $X_- \triangleq \{k \mid \overline{s}_k^{\mathrm{R}} < 0, k \in X\}$ as the subsets of users with demands having non-negative and negative real components respectively.

The basic idea of `CKP-bFPTAS` is to guess the total projections of the optimal solution on the real and imaginary axes, denoted by X_+^* and X_-^*, respectively. We can use $\tan\theta$ to upper bound the total projections for any feasible subset X as follows.

$$\sum_{k \in X} \overline{s}_k^{\mathrm{I}} \leq \overline{S}, \quad \sum_{k \in X_-} -\overline{s}_k^{\mathrm{R}} \leq \overline{S}\tan\theta, \quad \sum_{k \in X_+} \overline{s}_k^{\mathrm{R}} \leq \overline{S}(1 + \tan\theta). \quad (2.46)$$

We then solve two separate 2-KP problems (one for each quadrant) to find subsets of demands that satisfy the individual guessed total projections. But since 2-KP is generally NP-hard, we need to round-up the demands to get a problem that can be solved efficiently by *dynamic programming*. We show that the violation of the optimal solution to the rounded problem with respect to to the original problem is small in ϵ.

Next, we describe the rounding in detail. First, we define $L \triangleq \frac{\epsilon \overline{S}}{n(P(n)+1)}$, such that the new rounded-up demands $\hat{s_k}$ are defined by:

$$\hat{s}_k = \hat{s}_k^{\mathrm{R}} + \mathbf{i}\hat{s}_k^{\mathrm{I}} \triangleq \begin{cases} \left\lceil \frac{\overline{s}_k^{\mathrm{R}}}{L} \right\rceil \cdot L + \mathbf{i} \left\lceil \frac{\overline{s}_k^{\mathrm{I}}}{L} \right\rceil \cdot L, & \text{if } d_k^{\mathrm{R}} \geq 0, \\ \left\lfloor \frac{\overline{s}_k^{\mathrm{R}}}{L} \right\rfloor \cdot L + \mathbf{i} \left\lceil \frac{\overline{s}_k^{\mathrm{I}}}{L} \right\rceil \cdot L, & \text{otherwise.} \end{cases} \tag{2.47}$$

Let ξ_+ (and ξ_-), ζ_+ (and ζ_-) be respectively the guessed real and imaginary absolute total projections of the rounded demands in $X_+^\star$ (and $X_-^\star$). Then the possible values of ξ_+, ξ_-, ζ_+ and ζ_- are integer mutiples of L:

$$\xi_+ \in \mathcal{A}_+ \triangleq \left\{ 0, L, 2L, \dots, \left\lceil \frac{\overline{S}(1+P(n))}{L} \right\rceil \cdot L \right\}, \tag{2.48}$$

$$\xi_- \in \mathcal{A}_- \triangleq \left\{ 0, L, 2L, \dots, \left\lceil \frac{\overline{S} \cdot P(n)}{L} \right\rceil \cdot L \right\}, \tag{2.49}$$

$$\zeta_+, \zeta_- \in \mathcal{B} \triangleq \left\{ 0, L, 2L, \dots, \left\lceil \frac{\overline{S}}{L} \right\rceil \cdot L. \right\} \tag{2.50}$$

The next step is to solve the rounded instance exactly. Assume an arbitrary order on $\mathcal{I} = \{1, ..., n\}$. We use recursion to define a 3D table, with each entry $U(k, c_1, c_2)$ as the maximum utility obtained from a subset of users $\{1, 2, \dots, k\} \subseteq \mathcal{I}$ with demands $\{\hat{s_1}, \hat{s_2}, ..., \hat{s_k}\}$ that can fit exactly (i.e., satisfies the capacity constraint as an equation) within capacity c_1 on the real axis and c_2 on the imaginary axis. We denote by `2-KP-DP`$[\cdot]$ the algorithm for solving 2-KP by dynamic programming.

Theorem 2.5. For any fixed $\epsilon > 0$, Algorithm `CKP-bFPTAS` returns a $(1, 1+3\epsilon)$-approximation for $\mathrm{CKP}[0, \pi - \varepsilon]$ in time polynomial in both n and $\frac{1}{\epsilon}$.

Algorithm 4 `CKP-bFPTAS` $([(u_k, \overline{s}_k)_{k \in \mathcal{I}}, \overline{S}, \epsilon]$

$\hat{X} \leftarrow \varnothing$.
for all $\overline{s}_k$ and $k \in \mathcal{I}$ **do**
  Set $\hat{s}_k \leftarrow \hat{s}_k^{\mathrm{R}} + \mathbf{i}\hat{s}_k^{\mathrm{I}}$ as defined by Eqn. (2.47)
end for
for all $\xi_+ \in \mathcal{A}_+, \xi_- \in \mathcal{A}_-, \zeta_+, \zeta_- \in \mathcal{B}$ **do**
  if $(\xi_+ - \xi_-)^2 + (\zeta_+ + \zeta_-)^2 \le (1+2\epsilon)^2 \overline{S}^2$ **then**
    $F_+ \leftarrow$ `2-KP-DP`$(\{u_k, \frac{\hat{s}_k}{L}\}_{k \in \mathcal{I}_+}, \frac{\xi_+}{L}, \frac{\zeta_+}{L})$
    $F_- \leftarrow$ `2-KP-DP`$(\{u_k, \frac{-\hat{s}_k}{L}\}_{k \in \mathcal{I}_-}, \frac{\xi_-}{L}, \frac{\zeta_-}{L})$
    if $u(F_+ \cup F_-) > u(\hat{X})$ **then**
      $\hat{X} \leftarrow \{F_+ \cup F_-\}$
    end if
  end if
end for
return $\hat{X}$

Proof. First, the running time is proportional to the number of guesses, upper bounded by $O(\frac{1}{\epsilon^4} n^4 P(n)^6)$. For each guess, `2-KP-DP` constructs a table of size at most $O(\frac{1}{\epsilon^2} n^3 P(n)^4)$. Since we assumed $P(n)$ is polynomial in n, the total running time is polynomial in n and $\frac{1}{\epsilon}$.

To show the approximation ratio of 1, we note `CKP-bFPTAS` enumerates over all possible rounded projections subject to the capacity constraint in CKP and that `2-KP-DP` returns the exact optimal solution for each rounded problem. In particular, by Lemma 2.6 one of the choices would be the rounded projections for the optimum solution $X^\star$. It remains to show that the violation of the returned solution is small in ϵ. This is given in Lemma 2.7 below, which shows that the solution $\hat{X}$ to the rounded problem violates the capacity constraint by only a factor at most $(1+3\epsilon)$. $\square$

For any set $X \subseteq \mathcal{I}$, we write

$$D_+(X) \triangleq \sum_{k \in X_+} \overline{s}_k^{\mathrm{R}}, \; D_-(X) \triangleq \sum_{k \in X_-} -\overline{s}_k^{\mathrm{R}}, \; D_{\mathrm{I}}(X) \triangleq \sum_{k \in X} \overline{s}_k^{\mathrm{I}},$$
$$\hat{D}_+(X) \triangleq \sum_{k \in X_+} \hat{s}_k^{\mathrm{R}}, \; \hat{D}_-(X) \triangleq \sum_{k \in X_-} -\hat{s}_k^{\mathrm{R}}, \; \hat{D}_{\mathrm{I}}(X) \triangleq \sum_{k \in X} \hat{s}_k^{\mathrm{I}}$$

Algorithm 5 `2-KP-DP` $[(u_k, \hat{s}_k)_{k \in \mathcal{I}'}, \overline{S}^1, \overline{S}^2]$

1: Create a 3D table of size $|\mathcal{I}'| \times (\overline{S}^1 + 1) \times (\overline{S}^2 + 1)$, with each entry $U(k, c_1, c_2)$ according to:

$$U(1, c_1, c_2) \triangleq \begin{cases} u_1 & \text{if } \hat{s}_1^{\mathrm{R}} = c_1 \text{ and } \hat{s}_1^{\mathrm{I}} = c_2 \\ -\infty & \text{otherwise} \end{cases} \tag{2.51}$$

$$U(k, 0, 0) \triangleq 0 \quad \text{and} \quad U(k, c_1', c_2') \triangleq -\infty \text{ for all } c_1' < 0 \text{ or } c_2' < 0 \tag{2.52}$$

$$U(k, c_1, c_2) \triangleq \max\Big\{u_k + U(k-1, c_1 - \hat{s}_k^{\mathrm{R}}, c_2 - \hat{s}_k^{\mathrm{I}}), U(k-1, c_1, c_2)\Big\} \tag{2.53}$$

2: Create a 3D table of size $|\mathcal{I}'| \times \overline{S}^1 \times \overline{S}^2$, with each each entry $\mathcal{I}(k, c_1, c_2)$ according to:

$$\mathcal{I}(1, c_1, c_2) \triangleq \begin{cases} \{1\} & \text{if } U(1, c_1, c_2) = u_1 \\ \varnothing & \text{otherwise} \end{cases} \tag{2.54}$$

$$\mathcal{I}(k, c_1, c_2) \triangleq \begin{cases} \mathcal{I}(k-1, c_1, c_2), \\ \qquad \text{if } U(k, c_1, c_2) = U(k-1, c_1, c_2) \\ \mathcal{I}(k-1, c_1 - \hat{s}_k^{\mathrm{R}}, c_2 - \hat{s}_k^{\mathrm{I}}) \cup \{k\}, \\ \qquad \text{if } U(k, c_1, c_2) = u_k + U(k-1, c_1 - \hat{s}_k^{\mathrm{R}}, c_2 - \hat{s}_k^{\mathrm{I}}) \end{cases} \tag{2.55}$$

3: **return** $\mathcal{I}(n, \overline{S}^1, \overline{S}^2)$.

Then by (2.47) and the fact that $x \le t\lceil \frac{x}{t} \rceil \le x + t$ for any x, t such that $t > 0$, we have

$$\begin{aligned} \max\{\hat{D}_+(X) - L|X|, 0\} &\le D_+(X) \le \hat{D}_+(X), \\ \max\{\hat{D}_-(X) - L|X|, 0\} &\le D_-(X) \le \hat{D}_-(X), \\ \max\{\hat{D}_{\mathrm{I}}(X) - L|X|, 0\} &\le D_{\mathrm{I}}(X) \le \hat{D}_{\mathrm{I}}(X). \end{aligned} \tag{2.56}$$

Lemma 2.6. For any optimal solution $X^\star$ to CKP$[0, \pi - \varepsilon]$, $L \triangleq \frac{\epsilon \overline{S}}{nP(n)}$

and $\epsilon > 0$, we have

$$\Big(\sum_{k\in X^\star} \hat{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in X^\star} \hat{s}_k^{\mathrm{I}}\Big)^2 \le \overline{S}^2(1+2\epsilon)^2. \tag{2.57}$$

Proof. Using (2.56),

$$\begin{aligned}
&\Big(\sum_{k\in X^\star} \hat{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in X^\star} \hat{s}_k^{\mathrm{I}}\Big)^2 \qquad (2.58)\\
=&\Big(\hat{D}_+(X^\star) - \hat{D}_-(X^\star)\Big)^2 + \hat{D}_{\mathrm{I}}^2(X^\star)\\
=&\hat{D}_+^2(X^\star) + \hat{D}_-^2(X^\star) - 2\hat{D}_+(X^\star)\hat{D}_-(X^\star) + \hat{D}_{\mathrm{I}}^2(X^\star)\\
\le&(D_+(X^\star)+nL)^2 + (D_-(X^\star)+nL)^2 - 2D_+(X^\star)D_-(X^\star) + (D_{\mathrm{I}}(X^\star)+nL)^2\\
=&(D_+(X^\star)-D_-(X^\star))^2 + D_{\mathrm{I}}^2(X^\star) + 2nL(D_+(X^\star)+D_-(X^\star)+D_{\mathrm{I}}(X^\star))\\
&\quad + 3n^2L^2\\
=&\Big(\sum_{k\in X^\star} \overline{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in X^\star} \overline{s}_k^{\mathrm{I}}\Big)^2 + 2nL\Big(\sum_{k\in X^\star} |\overline{s}_k^{\mathrm{R}}| + \sum_{k\in X^\star} \overline{s}_k^{\mathrm{I}}\Big) + 3n^2L^2\\
\le&\overline{S}^2 + 4nL(P(n)+1)\overline{S} + 3n^2L^2 = \overline{S}^2 + 4\epsilon\overline{S}^2 + \frac{3\epsilon^2\overline{S}^2}{1+P(n))^2}\\
\le&\overline{S}^2(1+4\epsilon+\epsilon^2) \le \overline{S}^2(1+2\epsilon)^2. \qquad (2.59)
\end{aligned}$$

□

Lemma 2.7. Let $\hat{X}$ be the solution returned by `CKP-bFPTAS`. Then $|\sum_{k\in\hat{X}} \overline{s}_k| \le (1+3\epsilon)\overline{S}$.

Proof. As in (2.59),

$$\begin{aligned}
&\Big(\sum_{k\in\hat{X}} \overline{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in\hat{X}} \overline{s}_k^{\mathrm{I}}\Big)^2\\
=&\Big(D_+(\hat{X}) - D_-(\hat{X})\Big)^2 + D_{\mathrm{I}}^2(\hat{X})\\
=&D_+^2(\hat{X}) + D_-^2(\hat{X}) - 2D_+(\hat{X})D_-(\hat{X}) + D_{\mathrm{I}}^2(\hat{X}). \qquad (2.60)
\end{aligned}$$

If both $\hat{D}_+(\hat{X})$ and $\hat{D}_-(\hat{X})$ are less than nL, then the right-hand side of (2.60) can be bounded by

$$\begin{aligned}
&\hat{D}_+^2(\hat{X}) + \hat{D}_-^2(\hat{X}) + \hat{D}_{\mathrm{I}}^2(\hat{X})\\
\le&\hat{D}_+^2(\hat{X}) + \hat{D}_-^2(\hat{X}) - 2\hat{D}_+(\hat{X})\hat{D}_-(\hat{X}) + 2n^2L^2 + \hat{D}_{\mathrm{I}}^2(\hat{X})\\
=&(\hat{D}_+(\hat{X}) - \hat{D}_-(\hat{X}))^2 + \hat{D}_{\mathrm{I}}^2(\hat{X}) + 2n^2L^2. \qquad (2.61)
\end{aligned}$$

Otherwise, we bound the right-hand side of (2.60) by

$$\begin{aligned}
&\hat{D}_+^2(\hat{X}) + \hat{D}_-^2(\hat{X}) - 2(\hat{D}_+(\hat{X}) - nL)(\hat{D}_-(\hat{X}) - nL) + \hat{D}_{\mathrm{I}}^2(\hat{X})\\
=&(\hat{D}_+(\hat{X}) - \hat{D}_-(\hat{X}))^2 + \hat{D}_{\mathrm{I}}^2(\hat{X}) + 2nL(\hat{D}_+(\hat{X}) + \hat{D}_-(\hat{X})) - 2n^2L^2. \qquad (2.62)
\end{aligned}$$

Since $\hat{X} = F_+ \cup F_-$ is obtained from feasible solutions F_+ and F_- to `2-KP-DP`$(\{u_k, \frac{\hat{s}_k}{L}\}_{k\in\mathcal{I}_+}, \frac{\xi_+}{L}, \frac{\zeta_+}{L})$ and `2DKP`$(\{u_k, \frac{-\hat{s}_k}{L}\}_{k\in\mathcal{I}_-}, \frac{\xi_-}{L}, \frac{\zeta_-}{L})$, respectively, and $\xi_+, \xi_-, \zeta_+, \zeta_-$ satisfy the condition in Step 6, it follows from (2.60)-(2.62) that

$$\begin{aligned}
&\Big(\sum_{k\in\hat{X}} \overline{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in\hat{X}} \overline{s}_k^{\mathrm{I}}\Big)^2\\
\le&\Big(\sum_{k\in\hat{X}} \hat{s}_k^{\mathrm{R}}\Big)^2 + \Big(\sum_{k\in\hat{X}} \hat{s}_k^{\mathrm{I}}\Big)^2 + 2nL\sum_{k\in\hat{X}_-} |\hat{s}_k^{\mathrm{R}}| + 2n^2L^2\\
=&(\xi_+ - \xi_-)^2 + (\zeta_+ + \zeta_-)^2 + 2nL\xi_- + 2n^2L^2\\
\le&\Big((1+2\epsilon)^2\overline{S}^2 + 2n\frac{\epsilon}{n(P(n)+1)}\overline{S}^2 + 2n^2\frac{\epsilon^2}{n^2(P(n)+1)^2}\Big)\overline{S}^2\\
\le&\Big((1+2\epsilon)^2 + \epsilon + \frac{\epsilon^2}{2}\Big)\overline{S}^2 \le (1+3\epsilon)^2\overline{S}^2.
\end{aligned}$$

□

Many techniques for maximizing CKP$[0, \pi - \varepsilon]$ can be applied to minimizing CKP$_{\min}[0, \pi - \varepsilon]$ by slight modifications. By a similar approach, we can obtain the following result.

Theorem 2.8. There is an algorithm `CKP-bFPTAS`$_{\min}$ that, for any fixed $\epsilon > 0$, returns a $(1, 1 + O(\epsilon))$-approximation for CKP$_{\min}[0, \pi - \varepsilon]$ in time polynomial in both n and $\frac{1}{\epsilon}$.

2.5 Notes

Classical knapsack problems have been well studied in the literature. See Kellerer *et al.* (2010) for a comprehensive treatment of knapsack problems. Complex-demand Knapsack Problem (CKP) is a non-trivial generalization of the classical knapsack problem. CKP was initially studied in Yu and Chau (2013) in the context of smart grid, although it was also considered in Woeginger (2000) called 2-weight knapsack problem in a different context. Yu and Chau (2013) presented a $\frac{1}{2}$-approximation algorithm. The greedy algorithm in Section 2.2 was given in Karapetyan *et al.* (2018). Chau *et al.* (2014) and Chau *et al.* (2016) gave the PTAS and resource-augmented FPTAS for CKP. However, the PTAS presented in Chau *et al.* (2014) and Chau *et al.* (2016) relies on a complicated geometrical argument that allows also for implementing *truthful mechanisms* for CKP. The PTAS based on linear programming approach presented here is due to Elbassioni and Nguyen (2015). It was shown in Woeginger (2000) and Yu and Chau (2013), independently, that there exists no FPTAS for CKP$[0, \frac{\pi}{2}]$. Details of this will be given in Chapter 5.

3

Constant-Sized AC Electric Power Networks

Based on the results on single-capacitated AC electric power systems presented in the previous chapter, we next extend to the more general settings of constant-sized radial AC electric power networks. We consider the number of links in the distribution network (i.e., $|\mathcal{V}^+| = |\mathcal{E}| = m$) to be a *constant*, but we allow the number of users n to be a scalable parameter which is a part of the input to the problem. The running time of our algorithms will be polynomial in n.

3.1 Preliminaries of OPF

3.1.1 Assumptions

OPF with combinatorial constraints is hard to solve. Hence, we need to make some (natural) assumptions to facilitate our algorithms:

- A0: $f_0(-s_0^{\mathrm{R}})$ (resp., $h_0(-s_0^{\mathrm{R}})$) is non-increasing (resp., non-decreasing) in $-s_0^{\mathrm{R}} \in \mathbb{R}^+$.
- A1: $z_e \geq 0, \forall e \in \mathcal{E}$, which naturally hold in distribution networks.
- A2: $v_0 < \overline{v}_j, \forall j \in \mathcal{V}^+$, which is also assumed in Huang *et al.* (2017). Typically in a distribution network, $v_0 = 1$ (per unit), $\underline{v}_j = (.95)^2$

and $\overline{v}_j = (1.05)^2$; in other words, 5% deviation from the nominal voltage is allowed.

A3: $\mathrm{Re}(z_e^* s_k) \geq 0, \forall s_k \in \mathcal{S}_k,\ k \in \mathcal{I}, e \in \mathcal{E}$. Intuitively, A3 requires that the phase angle difference between any z_e and $s_k \in \mathcal{S}_k$ for $k \in \mathcal{I}$ is at most $\frac{\pi}{2}$. This assumption holds, if the discrete demands do not have large negative reactive power.

A4: $\left|\angle s_k - \angle s_{k'}\right| \leq \frac{\pi}{2}$ for any $s_k \in \mathcal{S}_k, s_{k'} \in \mathcal{S}_{k'}, k, k' \in \mathcal{N}$. Intuitively, A4 requires that the demands have "similar“ power factors. A4 can also be stated as $\mathrm{Re}(s_k^* s_{k'}) \geq 0$.

Assumptions A3 and A4 are motivated, from a theoretical point of view, by the inapproximability results which will be presented in Chapter 5 (if either assumption does not hold, then the problem cannot be approximated within any polynomial factor unless P=NP). Assumption A3 also holds in reasonable practical settings, see e.g., Huang *et al.* (2017). As we will see in Section 3.1.3, by performing an axis rotation, we may assume by A4 that $s_k \geq 0$. Clearly, under this and assumption A1, the reverse power constraint in (1.28) is implied by the forward power constraint ($|S_e| \leq \overline{S}_e$). It will also be observed below that under assumptions A1, A2 and A3, the voltage upper bounds in (1.27) can be dropped.

3.1.2 Tree Formulation for OPF

For convenience, we give another formulation of OPF, based the recursive “unfolding” of Eqns. (1.24)-(1.26). We start with the following simple lemma.

Lemma 3.1. Let $F \triangleq (s_0, s, v, \ell, S)$ be a vector satisfying (1.24)-(1.26). Then

$$S_{i,j} = \sum_{k \in \mathcal{N}_j} s_k + \sum_{e \in \mathcal{E}_j \cup \{(i,j)\}} z_e \ell_e, \qquad \forall (i,j) \in \mathcal{E}. \tag{3.1}$$

$$v_j = v_0 - 2\sum_{k\in\mathcal{N}} \mathrm{Re}\Big(\sum_{(h,t)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,t} s_k\Big) - \Big(2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}\big(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\ell_e\big) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t}\Big), \quad \forall j\in\mathcal{V}^+. \tag{3.2}$$

Proof. The first equation is obtained by rewriting $S_{i,j}$, recursively using Eqn. (1.24), substituting from the leaves. To see the second equation, rewrite Cons. (1.26) by recursively substituting v_j, for j, moving away from the root, and then substituting for $S_{h,t}$ using (3.1):

$$\begin{aligned}
v_j &= v_0 - 2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}(z^*_{h,t}S_{h,t}) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t}\\
&= v_0 - 2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}\Big(z^*_{h,t}\big(\sum_{k\in\mathcal{N}_t} s_k + \sum_{e\in\mathcal{E}_t\cup\{(h,t)\}} z_e\ell_e\big)\Big) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t},\\
&= v_0 - 2\sum_{k\in\mathcal{N}} \mathrm{Re}\Big(\sum_{(h,t)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,t}s_k\Big) - 2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}\Big(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\ell_e\Big)\\
&\quad - 2\sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t} + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t},
\end{aligned} \tag{3.3}$$

where the last statement follows from exchanging the summation operators, and using $z^*_e z_e = |z_e|^2$. □

It follows from Lemma 3.1, that we may equivalently formulate OPF as

$$(\textsc{t}\mathrm{OPF}) \quad \min_{s_0,s,v,\ell,S} \; f(s_0, s),$$
$$\text{subject to (1.23), (3.1), (1.25), (3.2), (1.27)} - \text{(1.31)}.$$

We shall refer to this as the *tree formulation* of OPF.

Lemma 3.2. Formulations OPF and tOPF are equivalent.

Proof. Given a feasible solution $F \triangleq (s_0, s, v, \ell, S)$ of OPF, Lemma 3.1 shows that F is also feasible for tOPF. Conversely, let $F \triangleq (s_0, s, v, \ell, S)$ be a feasible solution of tOPF, we show by that F satisfies (1.24) and (1.26). Consider first (1.24). Note by (3.1) that, for $(i, j) \in \mathcal{E}$,

$$
\begin{aligned}
S_{i,j} - \sum_{l:(j,l)\in\mathcal{E}} S_{j,l} &= \sum_{k\in\mathcal{N}_j} s_k + \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e\ell_e \\
&\quad - \left(\sum_{l:(j,l)\in\mathcal{E}} \sum_{k\in\mathcal{N}_l} s_k + \sum_{e\in\mathcal{E}_l\cup\{(j,l)\}} z_e\ell_e \right) \\
&= \sum_{k\in\mathcal{U}_j} s_k + z_{i,j}\ell_{i,j}.
\end{aligned}
$$

Consider next (1.26). Then (3.2) implies that, for $(i,j)\in\mathcal{E}$,

$$
\begin{aligned}
v_j - v_i &= 2\sum_{k\in\mathcal{N}} \mathrm{Re}\Big(\sum_{(h,t)\in\mathcal{P}_k\cap\mathcal{P}_i} z^*_{h,t}s_k \Big) - 2\sum_{k\in\mathcal{N}} \mathrm{Re}\Big(\sum_{(h,t)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,t} \Big) s_k \\
&\quad + \Big(2\sum_{(h,t)\in\mathcal{P}_i} \mathrm{Re}(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\ell_e) + \sum_{(h,t)\in\mathcal{P}_i} |z_{h,t}|^2\ell_{h,t}\Big) \\
&\quad - \Big(2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\ell_e) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\ell_{h,t}\Big) \\
&= -2\sum_{k\in\mathcal{N}_j} \mathrm{Re}\Big(\sum_{(h,t)\in\mathcal{P}_k\cap(\mathcal{P}_j\setminus\mathcal{P}_i)} z^*_{h,t}s_k \Big) \\
&\quad - \Big(2\sum_{(h,t)\in\mathcal{P}_j\setminus\mathcal{P}_i} \mathrm{Re}(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\ell_e) + \sum_{(h,t)\in\mathcal{P}_j\setminus\mathcal{P}_i} |z_{h,t}|^2\ell_{h,t}\Big) \\
&= -2\sum_{k\in\mathcal{N}_j} \mathrm{Re}(z^*_{i,j}s_k) - \Big(2\mathrm{Re}(z^*_{i,j}\sum_{e\in\mathcal{E}_j} z_e\ell_e) + |z_{i,j}|^2\ell_{i,j}\Big) \\
&= -2\sum_{k\in\mathcal{N}_j} \mathrm{Re}(z^*_{i,j}s_k) - 2\Big(\mathrm{Re}(z^*_{i,j}\sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e\ell_e)\Big) + |z_{i,j}|^2\ell_{i,j} \\
&= -2\mathrm{Re}\Big(z^*_{i,j}(\sum_{k\in\mathcal{N}_j} s_k + \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e\ell_e)\Big) + |z_{i,j}|^2\ell_{i,j} \\
&= -2\mathrm{Re}(z^*_{i,j}S_{i,j}) + |z_{i,j}|^2\ell_{i,j}.
\end{aligned}
$$

where the last equality follows from (3.1). □

3.1.3 Rotational Invariance of OPF

We note that if we rotate all complex quantities in the OPF problem (namely, $z_e, \mathcal{S}_k$) by a fixed angle ϕ, then the problem structure remains the same. For a set $\mathcal{S} \subseteq \mathbb{C}$, we write $\mathcal{S}_k e^{\mathbf{i}\phi} \triangleq \{s_k e^{\mathbf{i}\phi} : \ s_k \in \mathcal{S}\}$. Define the objective function in the rotated space by $f^{\phi}(s_0, s) \triangleq f(s_0 e^{\mathbf{i}\phi}, s)$.

Formally, rotated OPF is defined as follows.

$$(\mathrm{OPF}^{\phi}) \quad \max_{s_0, s, S, v, \ell} \ f^{\phi}(s_0, s)$$

$$\text{subject to} \quad (1.23), (1.25), (1.27) - (1.31)$$

$$S_{i,j} = \sum_{k \in \mathcal{U}_j} s_k e^{\mathbf{i}\phi} + \sum_{l:(j,l) \in \mathcal{E}} S_{j,l} + z_{i,j} e^{\mathbf{i}\phi} \ell_{i,j}, \ \forall (i,j) \in \mathcal{E} \tag{3.4}$$

$$v_j = v_i + |z_{i,j}|^2 \ell_{i,j} - 2\mathrm{Re}(z_{i,j}^* e^{-\mathbf{i}\phi} S_{i,j}), \ \forall (i,j) \in \mathcal{E} \tag{3.5}$$

$$s_k \in \mathcal{S}_k e^{\mathbf{i}\phi}, \quad \forall k \in \mathcal{N}. \tag{3.6}$$

Lemma 3.3. Problems OPF^{ϕ} and OPF are equivalent.

Proof. One can easily show that a feasible solution $F = (s_0, s, v, \ell, S)$ of (OPF^{ϕ}) can be mapped to a feasible solution $F' = (s_0', s', v, \ell, S')$ to OPF, such that $S_{i,j}' \triangleq S_{i,j} e^{-\mathbf{i}\phi}, s_0' \triangleq s_0 e^{-\mathbf{i}\phi}$ are rotated by ϕ, and vise versa. Moreover, the two objective functions are equal. □

Lemma 3.3 allows us to replace assumptions A0 and A4 by the following assumptions:

- A0′: $f_0(-s_0^{\mathrm{R}} \cos\phi - s_0^{\mathrm{I}} \sin\phi)$ is non-increasing in $s_0^{\mathrm{R}}, s_0^{\mathrm{I}}$.
- A4′: $s_k \geq 0$ for all $s_k \in \mathcal{S}_k, \ k \in \mathcal{N}$. This is because all demand sets satisfying A4 are now in the first quadrant after the rotation by ϕ.

Note that assumption A1 continues to hold for OPF^{ϕ}, assuming the original OPF problem satisfies A3: $z_e e^{\mathbf{i}\phi} \geq 0, \forall e \in \mathcal{E}$. This is because of A3, namely, $\mathrm{Re}(z_e^* \overline{s}_k) \geq 0, \forall k \in \mathcal{N}, \ e \in \mathcal{E}$, such that the phase angle difference between z_e and $\overline{s}_k$ is at most $\frac{\pi}{2}$. Note also that A1 and A4′ already imply A3.

From now on, we consider for convenience the rotated problem OPF^{ϕ}, which we simply denote by OPT, where

$$\phi \triangleq \max\Big\{\max_{k \in \mathcal{I}}\{-\angle \overline{s}_k\}, 0\Big\} \in [0, \frac{\pi}{2}]$$

Namely, ϕ is the minimum angle needed to rotate all the demand sets from the fourth quadrant to the first quadrant (see Fig. 3.1). We also simply drop ϕ from the definitions of $f^{\phi}(\cdot)$ and $h^{\phi}(\cdot)$.

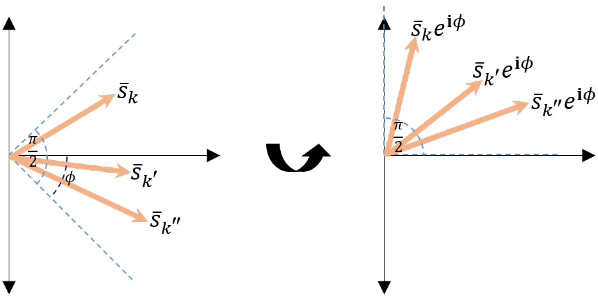


Figure 3.1: Angle ϕ is the minimum angle of rotation to rotate $(\bar{s}_k)_{k\in\mathcal{I}}$ into the first quadrant.

3.2 SOCP Relaxation of OPF

This section presents a brief review of SOCP relaxations of OPF. The idea of relaxing OPF to a convex optimization problem can significantly improve the solvability of OPF. Convex optimization problems can be solved (almost exactly) efficiently by polynomial-time algorithms. Under certain conditions, convex relaxations can be shown to obtain an optimal solution for OPF. A SOCP relaxation of OPF is obtained by replacing Cons. (1.23) by $\ell_{i,j} \geq \frac{|S_{i,j}|^2}{v_i}$, and replacing the discrete constraints in (1.30) by $s_k \in \widetilde{\mathcal{S}}_k \triangleq \text{conv.hull}\{\mathcal{S}_k\}$ for all $k \in \mathcal{N}$:

$$\text{(cOPF)} \quad \max_{s_0,s,S,v,\ell} \ f(s_0, s),$$

$$\text{subject to} \quad (1.24) - (1.29), (1.31)$$

$$\ell_{i,j} \geq \frac{|S_{i,j}|^2}{v_i}, \ \forall (i,j) \in \mathcal{E}, \tag{3.7}$$

$$s_k \in \widetilde{\mathcal{S}}_k, \quad \forall k \in \mathcal{N}. \tag{3.8}$$

To see that this is indeed an SOCP relaxation, note that Cons. (3.7) can be written as

$$\left\| \begin{pmatrix} 2S^{\mathrm{R}}_{i,j} \\ 2S^{\mathrm{I}}_{i,j} \\ \ell_{i,j} - v_i \end{pmatrix} \right\|_2 \leq \ell_{i,j} + v_i$$

Note that the constraint $s_k \in \widetilde{\mathcal{S}}_k$, for $k \in \mathcal{I}$, can be written as a subsystem of linear constraints:

$$s_k = \sum_{q \in \mathcal{S}_k} x_{k,q} q, \tag{3.9}$$

$$\sum_{q \in \mathcal{S}_k} x_{k,q} q = 1, \quad \forall k \in \mathcal{I}, \tag{3.10}$$

$$x_{k,q} \geq 0, \quad \forall q \in \mathcal{S}_k, \quad \forall k \in \mathcal{I}. \tag{3.11}$$

Using (3.9)-(3.11), we can rewrite cOPF as follows.

$$\begin{aligned} (\text{cOPF}) \quad & \max_{s_0, s, S, v, \ell} \ f_0(s_0) + \sum_{k \in \mathcal{F}} f_k(s_k) + \sum_{k \in \mathcal{I}} \sum_{q \in \mathcal{S}_k} x_{k,q} f_k(q), \\ \text{subject to} \quad & (1.24) - (1.29), (1.31), (3.7), (3.10) - (3.11) \\ & s_k \in \mathcal{S}_k, \quad \forall k \in \mathcal{F}. \end{aligned} \tag{3.12}$$

For a given $\hat{s} \in \mathbb{C}^n$ such that $s_k \in \widetilde{\mathcal{S}}_k$, we denote by OPF$[\hat{s}]$ (resp., cOPF$[\hat{s}]$ the restriction of OPF (resp., cOPF) where we set $s = \hat{s}$. The convex relaxation cOPF$[\hat{s}]$ is called (efficiently) *exact*, if every optimal solution $F^\star$ of cOPF$[\hat{s}]$ can be converted to an optimal solution of OPF$[\hat{s}]$, in a *polynomial* number of steps. This definition is adopted from Huang *et al.* (2017), Gan *et al.* (2015), Low (2014a), and Low (2014b), but also with an emphasis on *efficient computation*.

There are several sufficient conditions of exactness[1], which are imposed on a given (optimal) solution F of cOPF:

C1: The solution $F = (s_0, s, v, \ell, S)$ of (cOPF) satisfies the following linear system (in $(\widehat{S}_{i,j})_{(i,j) \in \mathcal{E}}$) and $(\widehat{v}_j)_{j \in \mathcal{V}^+}$):

$$\widehat{S}_{i,j} = \sum_{k \in \mathcal{U}_j} s_k + \sum_{l:(j,l) \in \mathcal{E}} \widehat{S}_{j,l} \quad \forall (i,j) \in \mathcal{E}, \tag{3.13}$$

$$\widehat{v}_i - \widehat{v}_j = 2\text{Re}(z^*_{i,j} \widehat{S}_{i,j}) \quad \forall (i,j) \in \mathcal{E}, \tag{3.14}$$

$$\text{Re}(z^*_{h,l} \widehat{S}_{i,j}) \geq 0, \quad \forall (i,j) \in \mathcal{E}, (h,l) \in \mathcal{E}_j, \tag{3.15}$$

$$\widehat{v}_j \leq \overline{v}_j \quad \forall j \in \mathcal{V}^+. \tag{3.16}$$

Note that Eqns. (3.13) and (3.14) correspond to Eqns. (1.16) and (1.17) in the DistFlow model.

[1] It should be noted that another sufficient condition for exactness was given in **gan2015exact**, but we will not consider here.

C2: The solution $F = (s_0, s, v, \ell, S)$ of (cOPF) satisfies

$$\sum_{k \in \mathcal{N}_j} \mathrm{Re}(z_{h,l}^* s_k) \geq 0 \quad \forall j \in \mathcal{V}^+, (h,l) \in \mathcal{E}_j \cup \{(i,j)\}. \tag{3.17}$$

Note that (3.17) is equivalent to (3.13) and the following slightly modified version of (3.15):

$$\mathrm{Re}(z_{h,l}^* \widehat{S}_{i,j}) \geq 0, \quad \forall (i,j) \in \mathcal{E}, (h,l) \in \mathcal{E}_j \cup \{(i,j)\}. \tag{3.18}$$

In Huang *et al.* (2017), it is shown that C1 is a sufficient condition for exactness of OPF considering *uni-directional* power capacity constraints from a leaf to the root[2]. In order to attain exactness of OPF with *bi-directional* power capacity constraints, a stronger condition ought to be considered. In addition to (3.15), it is also required that $\mathrm{Re}(z_{i,j}^* \widehat{S}_{i,j}) \geq 0$, which gives (3.18). Note that by (3.18) and A2, and the recursive substitution of $\widehat{v}_j$ from the root in (3.14) (as in Lemma 3.2), Cons. (3.16) is already satisfied as

$$\widehat{v}_j = v_0 - 2 \sum_{e \in \mathcal{P}_j} \mathrm{Re}(z_e^* \widehat{S}_e) \leq v_0 < \overline{v}_j,$$

where $\mathcal{P}_j$ denotes the unique path from the root 0 to node j.

Note also that A3 implies C2 when $\mathcal{N} = \mathcal{I}$ (as A3 applies to only discrete demands, whereas C2 applies to all the demands and edges within a subtree).

In proving the sufficient condition for exactness of the SOCP relaxation of OPF, we will make use of the following lemma.

Lemma 3.4. Let $F \triangleq (s_0, s, v, \ell, S)$ and $F' \triangleq (s_0', s', v', \ell', S')$ be two vectors satisfying (1.24) (or equivalently(3.1)), such that $s = s'$ and $\ell \leq \ell'$ (component wise). Suppose also that F satisfies C2. Then under assumption A1, $S_{i,j} \leq S_{i,j}'$ and $|S_{i,j}| \leq |S_{i,j}'|$ for all $(i,j) \in \mathcal{E}$, and $v_j \geq v_j'$ for all $j \in \mathcal{V}^+$.

Proof. Write $\Delta \ell_e \triangleq \ell_e - \ell_e' \leq 0$, $\Delta S_e \triangleq S_e - S_e'$, and $\Delta |S_e|^2 \triangleq |S_e|^2 - |S_e'|^2$, for $e \in \mathcal{E}$. Let $S_j \triangleq \sum_{k \in \mathcal{N}_j} s_k$, $L_{i,j} \triangleq \sum_{e \in \mathcal{E}_j \cup \{(i,j)\}} z_e \ell_e$, and

[2] The sufficient condition in Huang *et al.* (2017) is stated in a slightly different way, because their problem formulation adopts an opposite flow orientation.

$L'_{i,j} \triangleq \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e\ell'_e$. Note by (3.1) that $\tilde{S}_{i,j} = S_j + L_{i,j}$ and, similarly, $S'_{i,j} = S_j + L'_{i,j}$. It follows that, for all $(i,j) \in \mathcal{E}$,

$$\Delta S_{i,j} = L_{i,j} - L'_{i,j} = \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e \Delta\ell_e \leq 0, \tag{3.19}$$

where the inequality follows by assumption A1. This implies that $S_{i,j} \leq S'_{i,j}$. Furthermore,

$$\begin{aligned}
\Delta|S_{i,j}|^2 &= |S_{i,j}|^2 - |S'_{i,j}|^2 && (3.20)\\
&= (S^{\mathrm{R}}_{i,j})^2 - (S'^{\mathrm{R}}_{i,j})^2 + (S^{\mathrm{I}}_{i,j})^2 - (S'^{\mathrm{I}}_{i,j})^2 && (3.21)\\
&= \Delta S^{\mathrm{R}}_{i,j}(S^{\mathrm{R}}_{i,j} + S'^{\mathrm{R}}_{i,j}) + \Delta S^{\mathrm{I}}_{i,j}(S^{\mathrm{I}}_{i,j} + S'^{\mathrm{I}}_{i,j}) && (3.22)\\
&= \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z^{\mathrm{R}}_e \Delta\ell_e \big(2S^{\mathrm{R}}_j + L^{\mathrm{R}}_{i,j} + L'^{\mathrm{R}}_{i,j}\big) \\
&\quad + \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z^{\mathrm{I}}_e \Delta\ell_e \big(2S^{\mathrm{I}}_j + L^{\mathrm{I}}_{i,j} + L'^{\mathrm{I}}_{i,j}\big) && (3.23)\\
&= \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} 2\Delta\ell_e \mathrm{Re}(z^*_e S_j) + \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} \Delta\ell_e \mathrm{Re}(z^*_e L_{i,j}) \\
&\quad + \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} \Delta\ell_e \mathrm{Re}(z^*_e L'_{i,j}) \leq 0, && (3.24)
\end{aligned}$$

where the Inequality follows by A1, C2 and $\Delta\ell_e \leq 0$. Therefore $|S_{i,j}| \leq |S'_{i,j}|$. Finally, using (3.2), we get by A1 that

$$v_j - v'_j = -\Big(2\sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}\big(z^*_{h,t}\sum_{e\in\mathcal{E}_t} z_e\Delta\ell_e\big) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2\Delta\ell_{h,t}\Big) \geq 0.$$

□

Corollary 3.5. Let $F \triangleq \big(s_0, s, v, \ell, S\big)$ be a vector satisfying (1.24), (1.25) and (1.26) (or equivalently(3.1), (1.25) and (3.2)), and $|S_{i,j}| \leq \overline{S}_{i.j}$, for all $(i,j) \in \mathcal{E}$. Suppose also that F satisfies C2. Then under assumptions A1 and A2, F also satisfies $v_j \leq \overline{v}_j$, for all $j \in \mathcal{V}^+$, and $|S_{i,j} - z_{i,j}\ell_{i,j}| \leq \overline{S}_{i,j}$, for all $(i,j) \in \mathcal{E}$.

Proof. The first claim is immediate from (3.2) and assumptions A1, A2 and C2 as

$$v_j = v_0 - 2 \sum_{(h,t)\in\mathcal{P}_j} \sum_{k\in\mathcal{N}_t} \mathrm{Re}(z_{h,t}^* s_k) -$$
$$\Big(2 \sum_{(h,t)\in\mathcal{P}_j} \mathrm{Re}\big(z_{h,t}^* \sum_{e\in\mathcal{E}_t} z_e \ell_e\big) + \sum_{(h,t)\in\mathcal{P}_j} |z_{h,t}|^2 \ell_{h,t}\Big). \tag{3.25}$$

The second claim follows from Lemma 3.4 as $F' \triangleq (s_0, s, v, \ell', S')$ with

$$\ell'_{t,h} \triangleq \begin{cases} 0 & \text{if } (t,h) = (i,j) \\ \ell_{t,h} & \text{otherwise,} \end{cases}$$

satisfies $S'_{i,j} = S_{i,j} - z_{i,j}\ell_{i,j}$. □

Theorem 3.6. Let $F'' = (s''_0, s', v', \ell'', S'')$ be a feasible solution of $\mathrm{cOPF}[s']$ satisfying C2. Under assumptions A0, A1, and A2, there a feasible solution $F' = (s'_0, s', v', \ell', S')$ of $\mathrm{cOPF}[s']$ that satisfies $\ell_{i,j} = \frac{|S'_{i,j}|^2}{v'_i}$ for all $(i,j) \in \mathcal{E}$, and $f(F') \geq f(F'')$. Given F'', then such a solution F' can be found in polynomial time.

Proof. The proof follows essentially the same lines as in Gan *et al.* (2015), Low (2014a), Low (2014b), and Huang *et al.* (2017). We consider the following convex program[3]:

$$(\mathrm{cOPF}'[s']) \quad \min_{s_0,s,v,\ell,S} \sum_{e\in\mathcal{E}} \ell_e,$$
$$\text{subject to } (1.24) - (1.27), (1.31), (3.7)$$
$$\ell_{i,j} \leq \ell''_{i,j}, \quad \forall (i,j) \in \mathcal{E} \tag{3.26}$$
$$s_k = s'_k, \quad \forall k \in \mathcal{N}. \tag{3.27}$$

Clearly, $\mathrm{cOPF}'[s']$ is feasible as F'' satisfies all its constraints. Hence, it has an optimal solution $F' = (s'_0, s', v', \ell', S')$, which we claim to satisfy the statement of the theorem. First, we observe that F' is an optimal

[3]As mentioned in Section 1.3.7, typical convex programming solvers return a solution that is feasible within an absolute error $\epsilon > 0$, where the running time depends on $\log \frac{1}{\epsilon}$. For simplicity, we assume that the convex program can be solved exactly.

solution for cOPF$[s']$ (minimizing $\sum_{e\in\mathcal{E}} \ell_e$ among all such solutions). Indeed, Ineq. (3.26) implies that

$$\ell'_e \leq \ell''_e \leq \overline{\ell}_e \text{ for all } e \in \mathcal{E}. \tag{3.28}$$

It follows by Lemma 3.4 and the feasibility of F'' for cOPF$[s']$ that, for all $(i,j) \in \mathcal{E}$,

$$S'_{i,j} \leq S''_{i,j} \text{ and } |S'_{i,j}| \leq |S''_{i,j}| \leq \overline{S}_{i,j}. \tag{3.29}$$

In particular, for $(i,j) = (0,1)$, we obtain

$$-s'^{\mathrm{R}}_0 = S'^{\mathrm{R}}_{0,1} \leq S''^{\mathrm{R}}_{0,1} = -s''^{\mathrm{R}}_0, \tag{3.30}$$

implying by A0 that $f_0(-s'^{\mathrm{R}}_0) \geq f_0(-s''^{\mathrm{R}}_0)$ and hence by (3.27), $f(s'_0, s') \geq f(s''_0, s')$.

Note also that, since F' satisfies (1.24)-(1.26), we have by Corollary 3.5 that $|S'_{i,j} - z_{i,j}\ell'_{i,j}| \leq \overline{S}_{i,j}$, for all $(i,j) \in \mathcal{E}$. We conclude the feasibility of F' for cOPF$[s']$.

Next, suppose, for the sake of contradiction, that there exists an edge (h,t) such that $\ell'_{h,t} > \frac{|S'_{h,t}|^2}{v'_h}$. We construct a feasible solution $\tilde{F} = (\tilde{s}_0, s', \tilde{v}, \tilde{\ell}, \tilde{S})$ for cOPF$'[s']$ such that $\sum_{e\in\mathcal{E}} \tilde{\ell}_e < \sum_{e\in\mathcal{E}} \ell'_e$, leading to a contradiction to the optimality of F' for cOPF$'[s']$.

To obtain $\tilde{F} = (\tilde{s}_0, s', \tilde{v}, \tilde{\ell}, \tilde{S})$ from F', we set $\tilde{\ell}_{i,j} \leftarrow \frac{|S'_{i,j}|^2}{v'_i}$, then obtain $\tilde{S}$ and $\tilde{v}$ by substituting $\ell \leftarrow \tilde{\ell}$ in Eqns. (3.1) and (3.2). To complete the proof, we show the feasibility of $\tilde{F}$.

By the way we constructed $\tilde{F}$, all equality constraints of cOPF$'[s']$ are satisfied (via Lemma 3.2), and by the feasibility of F' for cOPF$[s']$ (in particular, Ineq. (3.7)), we also have

$$\tilde{\ell}_{i,j} = \frac{|S'_{i,j}|^2}{v'_i} \leq \ell'_{i,j} \leq \ell''_{i,j} \text{ for all } (i,j) \in \mathcal{E}. \tag{3.31}$$

It follows by Lemma 3.4 and the feasibility of F' that

$$\tilde{S}_{i,j} \leq S'_{i,j} \text{ and } |\tilde{S}_{i,j}| \leq |S'_{i,j}|, \qquad \forall (i,j) \in \mathcal{E}, \tag{3.32}$$

$$\tilde{v}_j \geq v'_j \geq \underline{v}_j, \qquad \forall j \in \mathcal{V}^+. \tag{3.33}$$

Note also that, since $\tilde{F}$ satisfies (1.24)-(1.26) and C2, we have by Corollary 3.5 that $\tilde{v}_j \le \overline{v}_j$ for all $j \in \mathcal{V}^+$. Moreover, by Ineqs. (3.32) and (3.33), $\tilde{\ell}_{i,j} = \frac{|S'_{i,j}|^2}{v'_i} \ge \frac{|\tilde{S}_{i,j}|^2}{\tilde{v}_i}$, hence, $\tilde{F}$ is feasible for cOPF$'[s']$.

Finally by the first inequality in (3.31) and the fact that $\ell'_{h,t} > \frac{|S_{h,t}|^2}{v_h} = \tilde{\ell}_{h,t}$, we have $\sum_{e\in\mathcal{E}} \tilde{\ell}_e < \sum_{e\in\mathcal{E}} \ell'_e$, contradicting the optimality of F' for cOPF$'[s']$. □

Corollary 3.7. Let $F' \triangleq (s'_0, s', v', \ell', S')$ be a feasible solution to cOPF and $\mathcal{I}' \subseteq \mathcal{I}$, $\hat{s} \in (\bigcup_{k\in\mathcal{I}} \mathcal{S}_k)^{\mathcal{I}'}$ be a given subset and a vector of discrete demands such that

$$\sum_{k\in\mathcal{I}'} f_k(\hat{s}_k) \ge \sum_{k\in\mathcal{I}} f_k(s'_k) - \epsilon f(s'_0, s'), \quad \text{for some } \epsilon > 0, \tag{3.34}$$

$$\sum_{k\in\mathcal{I}'} \mathrm{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,l}\hat{s}_k\Big) \le \sum_{k\in\mathcal{N}} \mathrm{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,l}s'_k\Big),\ \forall j \in \mathcal{V}^+, \tag{3.35}$$

$$\sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \hat{s}_k \le \sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} s'_k,\ \forall j \in \mathcal{V}^+. \tag{3.36}$$

Then under assumptions A0$'$, A1, A2, A3 and A4$'$, we can find in polynomial time a feasible solution $\hat{F} = (\hat{s}_0, \hat{s}, \hat{x}, \hat{S}, \hat{v}, \hat{\ell}_e)$ to OPF such that $f(\hat{F})(1-\epsilon) \ge f(F')$.

Proof. For $k \in \mathcal{N} \setminus \mathcal{I}'$, set $\hat{s}_k = s'_k$. We construct a feasible solution $\hat{F}' = (\hat{s}'_0, \hat{s}, \hat{S}', \hat{v}', \hat{\ell}')$ for cOPF$[\hat{s}]$ by setting $\hat{\ell}' = \ell'$ and substituting $\hat{s}$, $\hat{\ell}'$ in (3.1) and (3.2) to obtain $\hat{S}'$ and $\hat{v}'$, respectively.

We next show the feasibility of $\hat{F}'$. Clearly, by the definition of $\hat{\ell}'$, $\hat{\ell}'_{i,j} \le \overline{\ell}_{i,j}$. Write $\Delta S_e \triangleq \hat{S}'_e - S'_e$, and $\Delta|S_e|^2 \triangleq |\hat{S}'_e|^2 - |S'_e|^2$, for $e \in \mathcal{E}$. Let $S'_j \triangleq \sum_{k\in\mathcal{N}_j} s'_k$, $\hat{S}'_j \triangleq \sum_{k\in\mathcal{N}_j} \hat{s}'_k$, and $L'_{i,j} \triangleq \sum_{e\in\mathcal{E}_j\cup\{(i,j)\}} z_e\ell'_e$. Note by (3.1) that $S'_{i,j} = S'_j + L'_{i,j}$ and, $\hat{S}'_{i,j} = \hat{S}'_j + L'_{i,j}$. It follows by (3.36) that

$$\Delta S_{i,j} = \hat{S}'_j - S'_i \le 0. \tag{3.37}$$

In particular, for $(i,j) = (0,1)$, we obtain

$$-\hat{s}'^{\mathrm{R}}_0 = \hat{S}'^{\mathrm{R}}_{0,1} \le S'^{\mathrm{R}}_{0,1} = -s'^{\mathrm{R}}_0, \tag{3.38}$$

implying by A0 that $f_0(-\hat{s}_0'^{\mathrm{R}}) \geq f_0(-s_0'^{\mathrm{R}})$ and hence, $f(\hat{s}_0', \hat{s}') \geq (1-\epsilon) f(s_0', s')$. Furthermore, since $\hat{S}_{i,j}', S_{i,j}' \geq \mathbf{0}$ by A1, A4' and (3.1), it follows that

$$\Delta |S_{i,j}|^2 = |\hat{S}_{i,j}'|^2 - |S_{i,j}'|^2 \tag{3.39}$$
$$= (\hat{S}_{i,j}'^{\mathrm{R}})^2 - (S_{i,j}'^{\mathrm{R}})^2 + (\hat{S}_{i,j}'^{\mathrm{I}})^2 - (S_{i,j}'^{\mathrm{I}})^2 \tag{3.40}$$
$$= \Delta S_{i,j}^{\mathrm{R}}(\hat{S}_{i,j}'^{\mathrm{R}} + S_{i,j}'^{\mathrm{R}}) + \Delta S_{i,j}^{\mathrm{I}}(\hat{S}_{i,j}^{\mathrm{I}} + S_{i,j}'^{\mathrm{I}}) \leq 0. \tag{3.41}$$

Therefore, by the feasibility of S',

$$|\hat{S}_{i,j}'| \leq |S_{i,j}'| \leq \overline{S}_{i,j}. \tag{3.42}$$

Note also that, since $\hat{F}'$ satisfies (3.1) and (3.2) (or equivalently, (1.24) and (1.25)), we have by Corollary 3.5 that $|S_{i,j}' - z_{i,j}\ell_{i,j}'| \leq \overline{S}_{i,j}$, for all $(i,j) \in \mathcal{E}$ and $\hat{v}_j' \leq \overline{v}_j$, for all $j \in \mathcal{V}^+$. Moreover, (3.2) and (3.35) imply that

$$\hat{v}_j' \geq v_j' \geq \underline{v}_j. \tag{3.43}$$

By (3.42), (3.43), one obtains $\hat{\ell}_{i,j}' = \ell_{i,j}' \geq \frac{|S_{i,j}'|^2}{v_i'} \geq \frac{|\hat{S}_{i,j}'|^2}{\hat{v}_i'}$, hence $\hat{\ell}_{i,j}$ satisfies Cons. (3.7).

Finally we invoke Theorem 3.6 to convert $\hat{F}' = (\hat{s}_0', \hat{s}, \hat{S}', \hat{v}', \hat{\ell}')$ to a feasible solution $\hat{F}' = (\hat{s}_0, \hat{s}, \hat{S}, \hat{v}, \hat{\ell})$ of OPF$[\hat{s}]$ satisfying $f(\hat{F}) \geq f(\hat{F}') \geq (1-\epsilon) f(F')$. $\square$

Remark 3.1. Similar results as in Theorem 3.6 and Corollary 3.7 can be obtained for the minimization version $\mathrm{OPF}_{\min}$, where the objective function f is replaced by h and the assumption on h_0 in A0 is used.

3.3 PTAS

Based on SOCP relaxation of OPF, this section presents a $(1+\epsilon, 1)$-approximation algorithm (PTAS) for OPF. Note that we consider the number of links in the distribution network (i.e., $|\mathcal{V}^+| = |\mathcal{E}| = m$) is a constant.

3.3.1 Maximizing OPF

We will need here that assumptions A0, A1, A2, A3 and A4 hold. By Lemma 3.3, we may assume after rotation with an appropriate angle ϕ that A0′, A1, A2, A3 and A4′ hold instead. As mentioned earlier, we will denote for convenience the rotated problem by OPF.

The basic steps of PTAS are illustrated in Fig. 3.2. After convex relaxation and rotation, we enumerate possible partial guesses for configuring the control variables of a small subset of discrete demands. For each guess, we solve the remaining subproblem by relaxing the other discrete control variables to be continuous control variables, and then rounding the continuous control variables to obtain a feasible solution. This algorithm can attain a parameterized approximation ratio by carefully adjusting the number of partial guesses and rounding.

A formal description of the PTAS algorithm (called PTAS-cOPF) is presented as follows.

1. First, define a *partial guess* by $I_1 \subseteq \mathcal{I}$ and vector $\hat{s} \in (\bigcup_{k\in\mathcal{I}} \mathcal{S}_k)^{I_1}$. For each guess, we set $s_k = \hat{s}_k, \forall k \in I_1$ and $s_k = \mathbf{0}, \forall k \in I_0$.

2. Define a variant of cOPF with partially pre-configured and partially relaxed discrete control variables, denoted by $\text{P1}[I_1, \hat{s}]$, as follows.

$$
\begin{aligned}
(\text{P1}[I_1, \hat{s}]) \quad & \max_{s_0,s,S,v,\ell} \ f(s_0, s) \\
\text{subject to} \quad & (1.24) - (1.29), (1.31), (3.7), \\
& s_k = \hat{s}_k, \ \forall k \in I_1, && (3.44) \\
& s_k \in \widetilde{\mathcal{S}}'_k \triangleq \text{cvxhull}(\mathcal{S}'_k), \ \forall k \in \mathcal{I}', && (3.45)
\end{aligned}
$$

where

$$
\mathcal{S}'_k \triangleq \mathcal{S}_k \setminus \{q \in \mathcal{S}_k : \ f_k(q) \le \min_{k'\in I_1} \{f_{k'}(\hat{s}_{k'})\}\}, \tag{3.46}
$$

and $\mathcal{I}' \triangleq \mathcal{I} \setminus I_1$. Note that $\text{P1}[I_1, \hat{s}]$ is an SOCP (and hence is solvable in polynomial time). We then solve this relaxation to obtain an optimal solution $F' = (s'_0, s', S', v', \ell')$. Note that F' may not satisfy the discrete demand constraints (1.30) in cOPF. Next, F' will be rounded to obtain a feasible solution to cOPF.

3. Define P2$[F',\mathcal{I}']$ as follows.

$$(\text{P2}[F',\mathcal{I}']) \max_{x_{k,q}\in[0,1],k\in\mathcal{I}',q\in\mathcal{S}_k} \sum_{k\in\mathcal{I}'}\sum_{q\in\mathcal{S}_k} x_{k,q} f_k(q) \tag{3.47}$$

subject to

$$\sum_{k\in\mathcal{I}'} \text{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z_{h,l}^* s_k\Big) \le \sum_{k\in\mathcal{I}'} \text{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z_{h,l}^* s'_k\Big), \quad \forall j \in \mathcal{V}^+, \tag{3.48}$$

$$\sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \text{Re}(s_k) \le \sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \text{Re}(s'_k), \quad \forall j\in\mathcal{V}^+, \tag{3.49}$$

$$\sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \text{Im}(s_k) \le \sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \text{Im}(s'_k), \quad \forall j\in\mathcal{V}^+, \tag{3.50}$$

$$s_k = \sum_{q\in\mathcal{S}_k} x_{k,q}\cdot q, \quad \forall k\in\mathcal{I}', \tag{3.51}$$

$$\sum_{q\in\mathcal{S}_k} x_{k,q} = 1, \quad \forall k\in\mathcal{I}', \tag{3.52}$$

$$x_{k,q}\ge 0, \quad \forall q\in\mathcal{S}_k, \forall k\in\mathcal{I}'. \tag{3.53}$$

Note that P2$[F',\mathcal{I}']$ is an LP.

4. Suppose $(x''_{k,q})_{k\in\mathcal{I}',q\in\mathcal{S}_k}$ is an optimal BFS of P2$[F',\mathcal{I}']$. We define an integral solution $\hat{s}_k$, for $k\in\mathcal{I}'$, as follows

$$\hat{s}_k = \begin{cases} q, & \text{if there is a } q\in\mathcal{S}_k \text{ such that } x_{k,q}=1 \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.54}$$

Note that we use here the assumption that $\mathbf{0}\in\mathcal{S}_k$, for all $k\in\mathcal{I}$.

5. Then, obtain the corresponding $\hat{s}_0, \hat{s}, \hat{S}, \hat{\ell}, \hat{v}$ by invoking Corollary 3.7 with $s' = ((\hat{s}_k)_{k\in\mathcal{I}'}, (s'_k)_{k\in\mathcal{N}\setminus\mathcal{I}'})$.

6. The output solution will be the one having the maximal objective value among all guesses.

The pseudo-codes of PTAS-cOPF are given in Algorithm 6.

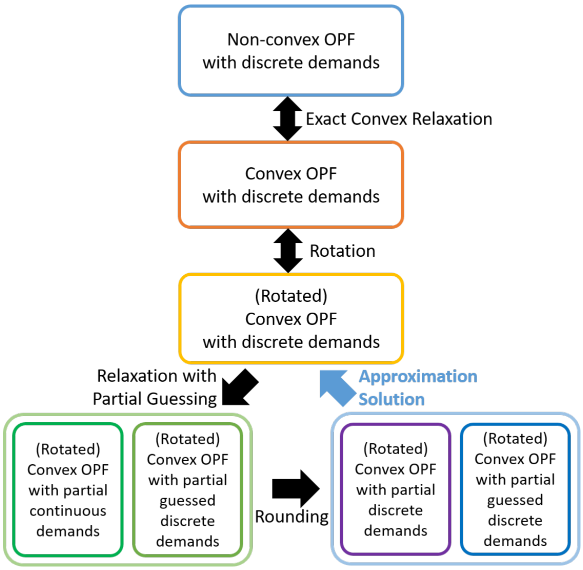


Figure 3.2: Basic steps of PTAS for OPF.

3.3.2 Analysis of Approximation Ratio

In this section, the approximation ratio of PTAS-cOPF will be shown to be $(1-\epsilon)$, if one sets the size of partial guesses of satisfiable discrete demands as $|I_1| \le \frac{6m}{\epsilon}$, where m is the number of edges in distribution network. Therefore, one can adjust the approximation ratio by limiting the size of I_0 in partial guessing.

Remark 3.2. To (practically) speed up PTAS-cOPF, one can first compute the optimal objective value (denoted by $\underline{f}$) of P1 by taking $I_1 = \varnothing$, which naturally is a lower bound to that of cOPF. Then we may run PTAS-cOPF and check the gap between the objective value of the produced solution and $\underline{f}$; when this gap is sufficiently small, we can stop. This may allow us to skip a lot of partial enumeration if $\underline{f}$ is already close to the solution of PTAS-cOPF (which is often observed as the case in the experimental evaluation in Chapter 6).

We will use the following corollary of Lemma 1.1 in our derivation of the approximation ratio of PTAS-cOPF.

Algorithm 6 PTAS-cOPF

Require: $\epsilon, v_0, (\underline{v}_j, \overline{v}_j)_{j\in\mathcal{V}^+}, (\overline{S}_e, \overline{\ell}_e, z_e)_{e\in\mathcal{E}}, (\underline{s}_k, \overline{s}_k)_{k\in\mathcal{N}}$

Ensure: Solution $\hat{F} = (\hat{s}_0, \hat{s}, \hat{x}, \hat{S}, \hat{v}, \hat{\ell})$ to cOPF

$f_{\max} \leftarrow 0$
for each set $I_1 \subseteq \mathcal{I}$ and $\hat{s} \in (\bigcup_{k\in\mathcal{I}} \mathcal{S}_k)^{I_1}$ such that $|I_1| \le \frac{6m}{\epsilon}$ **do**
    $\mathcal{I}' \leftarrow \mathcal{I}\backslash I_1$
    $\mathcal{S}'_k \triangleq \mathcal{S}_k \setminus \{q \in \mathcal{S}_k : \ f_k(q) \le \min_{k'\in I_1}\{f_{k'}(\hat{s}_{k'})\}\}$, for $k \in \mathcal{I}'$
    if P1$[I_1, \hat{s}]$ is feasible **then**
        $F' \leftarrow$ Optimal solution of P1$[I_1, \hat{s}]$
        $(x''_{k,q})_{k\in\mathcal{I}',q\in\mathcal{S}_k} \leftarrow$ Optimal BFS of P2$[F', \mathcal{I}']$
        $(\hat{s}_k)_{k\in\mathcal{I}'} \leftarrow$ rounded solution according to (3.54).
        $(\hat{s}_0, \hat{s}, \hat{S}, \hat{v}, \hat{\ell}) \leftarrow$ solution returned by Corollary 3.7
        if $f_{\max} < f(\hat{s}_0, \hat{s})$ **then**
            $\hat{F} \leftarrow (\hat{s}_0, \hat{s}, \hat{S}, \hat{v}, \hat{\ell})$
            $f_{\max} \leftarrow f(\hat{s}_0, \hat{s})$
        end if
    end if
end for
return $\hat{F}$

Lemma 3.8. Let x be a BFS of (P2$[F', \mathcal{I}']$). Then x has at most $6m$ fractional components: $|\{k \in \mathcal{I}',\ q \in \mathcal{S}_k \mid x_{k,q} \in (0,1)\}| \le 6m$.

Proof. The proof essentially follows an argument from Patt-Shamir and Rawitz (2010). First, for all $k \in \mathcal{I}'$, substitute s_k from (3.51) in (3.48)-(3.50), and consider the resulting linear system. Let $h = |\mathcal{I}'|$. By Lemma 1.1, the number of strictly positive components in x is at most $3m + h$. Furthermore, Cons. (3.52) impose that for each $k \in \mathcal{I}$ among those h users, there is a $q \in \mathcal{S}_k$ such that $x_{k,q} > 0$. The remaining (at most) $3m$ positive variables can appear in at most $3m$ of Cons. (3.52), implying that at least $\max\{h - 3m, 0\}$ variables are set to 1. It follows that the total number variables taking non-integral values is at most $3m + h - \max\{h - 3m, 0\} \le 6m$. □

Theorem 3.9. With assumptions A1',A2,A3,A4', for any fixed $\epsilon > 0$, PTAS-cOPF provides a $(1 - \epsilon, 1)$-approximation solution for cOPF$^{\phi}$, in time polynomial in n.

Proof. It is easy see that the running time of PTAS-cOPF is polynomial in n, for any fixed $\epsilon > 0$. Next, we show that the output solution $\hat{F}$ is $(1-\epsilon, 1)$-approximation for cOPF.

Let $F^\star = (s_0^\star, s^\star, S^\star, v^\star, \ell^\star)$ be an optimal solution of cOPF. Define $I_1^\star \triangleq \{k \in \mathcal{I} \mid s_k^\star \neq \mathbf{0}\}$. There are two cases:

1. If $|I_1^\star| \leq \frac{6m}{\epsilon}$, then there exists a partial guess I_1, such that $I_1 = I_1^\star$ and $\hat{s}_k = s_k^\star$ for $k \in I_1^\star$. Thus, PTAS-cOPF can find an optimal solution $F^\star$ of cOPF by enumerating all possible I_1 and $\hat{s}_k$ such that $|I_1| \leq \frac{6m}{\epsilon}$.

2. Otherwise, $|I_1^\star| > \frac{6m}{\epsilon}$, then PTAS-cOPF can still find some I_1, which is a subset of satisfiable discrete demands in $I_1^\star$ with a number of $\lfloor \frac{6m}{\epsilon} \rfloor$ highest $f_k(\cdot)$:
$$I_1 \subseteq I_1^\star \text{ and } |I_1| = \lfloor \tfrac{6m}{\epsilon} \rfloor \text{ and } \min_{k\in I_1, q\in\mathcal{S}_k} f_k(q) > \max_{k\in I_1^\star\setminus I_1, q\in\mathcal{S}_k} f_k(q) \tag{3.55}$$
Next, we assume I_1 satisfying (3.55) and $\mathcal{S}_k'$ satisfying (3.46), for $k \in \mathcal{I}\backslash I_1$.

Then, we focus on case 2. Let us consider an optimal solution $F' = (s_0', s', S', v', \ell')$ of P1$[I_1, \hat{s}]$, where I_1 satisfies (3.55) and $\hat{s}_k = s_k^\star$ for $k \in I_1$. Since $F^\star$ is feasible for P1$[I_1, \hat{s}]$, it follows that
$$f(s_0', s') \geq f(s_0^\star, s^\star). \tag{3.56}$$

Next, let us consider an optimal BFS $(x''_{k,q})_{k\in\mathcal{I}', q\in\mathcal{S}_k}$ of P2$[F', \mathcal{I}']$. Note that s' can be extended to a feasible solution to P2$[F', \mathcal{I}']$ (where Cons. (3.48)-(3.50) are tight) by defining x' in the obvious way. It follows that
$$\sum_{k\in\mathcal{I}'} \sum_{q\in\mathcal{S}_k} x''_{k,q} f_k(q) \geq \sum_{k\in\mathcal{I}'} f_k(s_k'). \tag{3.57}$$

By Lemma 3.8, at most $6m$ components in $(x''_{k,q})_{k\in\mathcal{I}', q\in\mathcal{S}_k}$ are fractional. For each fractional component, say $x''_{k,q}$, for $l \in \mathcal{I}'$ and $q \in \mathcal{S}_k$, one has by (3.46)
$$f_k(q) \leq \min_{k'\in I_1} f_{k'}(\hat{s}_{k'}) \leq \frac{1}{|I_1|} \sum_{k'\in I_1} f_{k'}(\hat{s}_{k'}). \tag{3.58}$$

Therefore, by our rounding Step (3.54), we set $x''_{k,q}$ to 0 for at most $6m$ components. Thus, by (3.58) and (3.57) and the non-negativity of $f_0(\cdot),\ f_k(\cdot)$,

$$\begin{aligned}\sum_{k\in\mathcal{I}'} f_k(\hat{s}_k) &\geq \sum_{k\in\mathcal{I}'}\sum_{q\in\mathcal{S}_k} x''_{k,q} f_k(q) - \frac{6m}{|I_1|}\sum_{k\in I_1} f_k(\hat{s}_k) \\ &\geq \sum_{k\in\mathcal{I}'}\sum_{q\in\mathcal{S}_k} x''_{k,q} f_k(q) - \epsilon \sum_{k\in I_1} f_k(\hat{s}_k) \\ &\geq \sum_{k\in\mathcal{I}'} f_k(s'_k) - \epsilon f(s'_0, s'). \end{aligned} \tag{3.59}$$

Finally, by (3.59) and Corollary 3.7, one obtains that the solution returned in Step 9 satisfies

$$f(\hat{s}_0, \hat{s}) \geq (1-\epsilon) f(s'_0, s') \geq (1-\epsilon) f(s^\star_0, s^\star).$$

This completes the proof of the theorem. □

3.3.3 Minimizing OPF

Lastly, we consider $\text{OPF}_{\min}$, the minimization version of OPF (i.e., replacing $\max_{s_0,s,S,v,\ell} f(s_0,s)$ in OPF by $\min_{s_0,s,S,v,\ell} h(s_0,s)$.). The aforementioned techniques for the maximization version of OPF (i.e., PTAS-cOPF) can be adapted with minor modifications to $\text{OPF}_{\min}$.

Define $h^\phi(s_0,s) \triangleq h(s_0 e^{-\mathbf{i}\phi}, s)$, and $\text{cOPF}^\phi_{\min}$ as the minimizing version of cOPF^ϕ. As before, we simply drop "ϕ“ from these definitions. Define the minimizing versions of $\text{P1}[I_1,\hat{s}]$ and $\text{P2}[F',\mathcal{I}']$ as follows.

$$\begin{aligned} &(\text{P1}_{\min}[I_1,\hat{s}]) && \min_{s_0,s,S,v,\ell}\ h(s_0,s) \\ &\text{subject to} && (1.24)-(1.29),(1.31),(3.7) \\ & && s_k = \hat{s}_k,\ \forall k\in I_1, && (3.60) \\ & && s_k \in \widetilde{\mathcal{S}}'_k \triangleq \text{cvxhull}(\mathcal{S}'_k),\ \forall k\in\mathcal{I}', && (3.61) \end{aligned}$$

where

$$\mathcal{S}'_k \triangleq \mathcal{S}_k \setminus \{q\in\mathcal{S}_k:\ h_k(q) \leq \min_{k'\in I_1}\{h_{k'}(\hat{s}_{k'})\}\}, \tag{3.62}$$

and $\mathcal{I}' \triangleq \mathcal{I}\setminus I_1$.

$$(\mathrm{P2}_{\min}[F', \mathcal{I}']) \min_{x_{k,q}\in[0,1], k\in\mathcal{I}', q\in\mathcal{S}_k} \sum_{k\in\mathcal{I}'} \sum_{q\in\mathcal{S}_k} x_{k,q} h_k(q)$$

subject to

$$\sum_{k\in\mathcal{I}'} \mathrm{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,l} s_k\Big) \le \sum_{k\in\mathcal{I}'} \mathrm{Re}\Big(\sum_{(h,l)\in\mathcal{P}_k\cap\mathcal{P}_j} z^*_{h,l} s'_k\Big), \forall j \in \mathcal{V}^+, \tag{3.63}$$

$$\sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \mathrm{Re}(s_k) \le \sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \mathrm{Re}(s'_k), \quad \forall j \in \mathcal{V}^+, \tag{3.64}$$

$$\sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \mathrm{Im}(s_k) \le \sum_{k\in\mathcal{N}_j\cap\mathcal{I}'} \mathrm{Im}(s'_k), \quad \forall j \in \mathcal{V}^+, \tag{3.65}$$

$$s_k = \sum_{q\in\mathcal{S}_k} x_{k,q} \cdot q, \quad \forall k \in \mathcal{I}', \tag{3.66}$$

$$\sum_{q\in\mathcal{S}_k} x_{k,q} = 1, \quad \forall k \in \mathcal{I}', \tag{3.67}$$

$$x_{k,q} \ge 0, \quad \forall q \in \mathcal{S}_k, \quad \forall k \in \mathcal{I}'. \tag{3.68}$$

By a similar approach, we have the following result.

Theorem 3.10. For any fixed $\epsilon > 0$, PTAS-cOPF$_{\min}$ returns a $(1+\epsilon, 1)$-approximation for $\mathrm{OPF}_{\min}[0, \pi - \varepsilon]$ in time polynomial in both n and $\frac{1}{\epsilon}$, assuming A0, A1, A2, A3, A4.

3.4 Greedy Approximation Algorithm

Complementing with the PTAS in previous section, this section presents an alternate approximation algorithm for maximizing OPF, under DistFlow model, considering all discrete demands. This approximation algorithm is based on a greedy approximation algorithm for the unsplittable flow problem. We then analyze the approximation ratio of our approximation algorithm. Next, we adapt the algorithm as heuristic for OPF considering a mix of discrete and continuous demands.

Algorithm 7 PTAS-cOPF$_{\min}$

Require: $\epsilon, v_0, (\underline{v}_j, \overline{v}_j)_{j\in\mathcal{V}^+}, (\overline{S}_e, \overline{\ell}_e, z_e)_{e\in\mathcal{E}}, (\underline{s}_k, \overline{s}_k)_{k\in\mathcal{N}}$

Ensure: Solution $\hat{F} = (\hat{s}_0, \hat{s}, \hat{x}, \hat{S}, \hat{v}, \hat{\ell})$ to cOPF$_{\min}$

$h_{\min} \leftarrow \infty$

for each set $I_1 \subseteq \mathcal{I}$ and $\hat{s} \in (\bigcup_{k\in\mathcal{I}} \mathcal{S}_k)^{I_1}$ such that $|I_1| \le \frac{6m}{\epsilon}$ **do**

  $\mathcal{I}' \leftarrow \mathcal{I}\backslash I_1$

  $\mathcal{S}'_k \triangleq \mathcal{S}_k \setminus \{q \in \mathcal{S}_k : \ h_k(q) \le \min_{k'\in I_1}\{h_{k'}(\hat{s}_{k'})\}\}$, for $k \in \mathcal{I}'$

  if P1$_{\min}[I_1, \hat{s}]$ is feasible **then**

    $F' \leftarrow$ Optimal solution of P1$_{\min}[I_1, \hat{s}]$

    $(x''_{k,q})_{k\in\mathcal{I}', q\in\mathcal{S}_k} \leftarrow$ Optimal BFS of P2$_{\min}[F', \mathcal{I}']$

    $(\hat{s}_k)_{k\in\mathcal{I}'} \leftarrow$ rounded solution according to (3.54).

    $(\hat{s}_0, \hat{s}, \hat{S}, \hat{v}, \hat{\ell}) \leftarrow$ solution returned by Corollary 3.7

    if $h_{\min} > h(\hat{s}_0, \hat{s})$ **then**

      $\hat{F} \leftarrow (\hat{s}_0, \hat{s}, \hat{x}, \hat{S}, \hat{v}, \hat{\ell})$

      $h_{\min} \leftarrow h(\hat{s}_0, \hat{s})$

    end if

  end if

end for

return $\hat{F}$

3.4.1 Preliminaries of the Unsplittable Flow Problem

When the demands are real-valued, our problem is related to the well-known *unsplittable flow problem* (UPF): given a tree $\mathcal{T} = (V, \mathcal{E})$ with edge capacities c_e for $e \in \mathcal{E}$, and a set of paths $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ where each path $\mathcal{P}_k$ is associated with a demand $d_k \in \mathbb{R}^+$ and a utility u_k, the objective is to choose the maximum-utility subset of paths whose total demand on each edge does not exceed the capacity. This can be formally defined as follows.

$$\text{(UFP)} \quad \max_x \sum_{k\in\mathcal{I}} u_k x_k,$$

$$\text{subject to} \quad \sum_{k:\ e\in\mathcal{P}_k} d_k x_k \le c_e, \qquad \forall\, e \in \mathcal{E}, \tag{3.69}$$

$$x_k \in \{0, 1\}, \qquad \forall k \in \mathcal{N}. \tag{3.70}$$

In UPF, each demand is associated with an arbitrary path from a source to a sink, while in our problem all demands share a single source (or sink). A special case of UPF is when all demands and edge capacities are uniform is the classical maximum edge-disjoint path problem (MEDP). Note that MEDP is one of the original Karp's NP-complete problems (Karp, 1972).

3.4.2 Simplified OPF

Recall that $P_e \subseteq \mathcal{E}$ denotes the path from edge e to the root, and by $P_k \subseteq \mathcal{E}$ the path from node k to the root. Note that $\ell_{i,j} = \frac{|S_{i,j}|^2}{v_i}$ is positive. We assume that $\ell_e \le \bar{\ell}_e$ for a constant $\bar{\ell}_e$ independent of solution $(x_k)_{k \in \mathcal{N}}$.

We rewrite Cons. (1.24) in OPF by Eqn. (3.1) in the tree formulation. Note that

$$|S_{i,j}| \le \Big| \sum_{k \in \mathcal{N}_j} \bar{s}_k x_k \Big| + \Big| \sum_{e \in \mathcal{E}_i} z_e \ell_e \Big| \le \Big| \sum_{k \in \mathcal{N}_j} \bar{s}_k x_k \Big| + \Big| \sum_{e \in \mathcal{E}_i} z_e \bar{\ell}_e \Big|$$

Let $\widehat{L}_e \triangleq \big| \sum_{e \in \mathcal{E}_i} z_e \bar{\ell}_e \big|$. Thus, the power capacity constraint Cons. (1.26) can be implied by the following constraint:

$$\Big| \sum_{k \in \mathcal{N}_j} \bar{s}_k x_k \Big| = \Big| \sum_{k : e \in P_k} \bar{s}_k x_k \Big| \le \overline{\mathsf{S}}_e \triangleq \overline{S}_e - \widehat{L}_e$$

Also, we rewrite Cons. (1.26) by Eqn. (3.2). Let $\underline{V_e} \triangleq \frac{1}{2}(v_0 - \overline{v})$ and $\overline{V_e} \triangleq \frac{1}{2}(v_0 - \underline{v}) - \Big(\sum_{(h,t) \in \mathcal{P}_j} \mathrm{Re}\big(z^*_{h,t} \sum_{e \in \mathcal{E}_t} z_e \bar{\ell}_e\big) + \sum_{(h,t) \in \mathcal{P}_j} |z_{h,t}|^2 \bar{\ell}_{h,t} \Big)$.

Thus, voltage constraint Cons. (1.26) can be implied by the following constraints:

$$\underline{V_e} \le \sum_{k \in \mathcal{N}} \Big(\sum_{e' \in P_k \cap P_e} z^{\mathrm{R}}_{e'} \bar{s}^{\mathrm{R}}_k + z^{\mathrm{I}}_{e'} \bar{s}^{\mathrm{I}}_k \Big) x_k \le \overline{V_e},$$

We define a simplified version of maximizing OPF (denoted by sOPF), such that a feasible solution to sOPF is a feasible solution to OPF.

$$\text{(sOPF)} \quad \max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k$$

$$\text{subject to} \quad \Big| \sum_{k: e \in P_k} \overline{s}_k x_k \Big| \le \overline{\mathsf{S}}_e, \qquad \forall e \in \mathcal{E}, \tag{3.71}$$

$$\underline{V_e} \le \sum_{k \in \mathcal{N}} \Big(\sum_{e' \in P_k \cap P_e} z^{\mathrm{R}}_{e'} \overline{s}^{\mathrm{R}}_k + z^{\mathrm{I}}_{e'} \overline{s}^{\mathrm{I}}_k \Big) x_k \le \overline{V_e}, \qquad \forall e \in \mathcal{E}, \tag{3.72}$$

$$x_k \in \{0, 1\}, \qquad \forall k \in \mathcal{I},$$

$$x_k \in [0, 1], \qquad \forall k \in \mathcal{F}.$$

We denote by sOPF_{C} when sOPF considers only power capacity constraints Cons. (3.71) without Cons. (3.72), whereas by sOPF_{V} when sOPF considers only voltage constraints Cons. (3.72) without Cons. (3.71).

3.4.3 Approximation Algorithm for sOPF

In this section, we provide an approximation algorithm (`GreedyDisDm`) to sOPF considering only discrete demands ($\mathcal{I}$). This algorithm is inspired by an $O(\log n)$-approximation algorithm for the unsplittable flow problem in Chekuri *et al.* (2006).

`GreedyDisDm` (Algorithm 8) first normalizes the customers' utilities by $\bar{u}_k \triangleq \lfloor \frac{u_k}{L} \rfloor$. Then it partitions customers into groups $(\widehat{\mathcal{I}}_1, ..., \widehat{\mathcal{I}}_{\lceil 2 \log n \rceil + 1})$ according to the ranges of normalized utilities, such that the utilities of the i-th group are within $[2^i, 2^{i+1})$. For each group, it next calls `GreedysOPF` to return a feasible solution for the group of customers. Finally, `GreedyDisDm` returns the output solution as the group from `GreedysOPF` with the maximum utility.

`GreedysOPF` (Algorithm 9) first sorts the customers in a non-decreasing order according to the magnitudes of their demands. Then, it packs their demands greedily sequentially in that order, if the power capacity constraints or voltage constraints are not violated. The customers who can be satisfied are placed in the set M.

Algorithm 8 `GreedyDisDm`$[(u_k, \overline{s}_k)_{k\in\mathcal{I}}]$

Require: customers' utilities (u_k) and discrete demands $(\overline{s}_k)$

Let $L \triangleq \frac{u_{\max}}{n^2}$ and $u_{\max} = \max_{k\in\mathcal{I}} u_k$

Let $\bar{u}_k \triangleq \lfloor \frac{u_k}{L} \rfloor$ for each $k \in \mathcal{I}$

▷ Group customers according to the range of their utilities

$\widehat{\mathcal{I}}_1 \leftarrow \{k \in \mathcal{I} \mid \bar{u}_k \in [0, 2)\}$

for $i = 2, ..., \lceil 2\log n \rceil$ **do**

  $\widehat{\mathcal{I}}_i \leftarrow \{k \in \mathcal{I} \mid \bar{u}_k \in [2^i, 2^{i+1})\}$

end for

▷ Call `GreedysOPF` to solve sub-problems with $\widehat{\mathcal{I}}_i$

for $i = 1, ..., \lceil 2\log n \rceil$ **do**

  $M_i \leftarrow$ `GreedysOPF`$[(\overline{s}_k)_{k\in\widehat{\mathcal{I}}_i}]$

end for

▷ Return the group with maximum utility

return M such that $u(M) = \max_{i=1,...,\lceil 2\log n\rceil} u(M_i)$

Algorithm 9 `GreedysOPF`$[(\overline{s}_k)_{k\in\widehat{\mathcal{I}}}]$

Require: customers' discrete demands $(\overline{s}_k)$

Sort customers in $\widehat{\mathcal{I}}$ according to the magnitudes of demands:

$$|\overline{s}_1| \le |\overline{s}_2| \le ... \le |\overline{s}_{|\widehat{\mathcal{I}}|}|$$

$M \leftarrow \varnothing$

for each $k \in \widehat{\mathcal{I}}$ **do**

  if $\left| \sum_{k' \in M\cup\{k\} : e \in P_{k'}} \overline{s}_{k'} \right| \le \overline{\mathsf{S}}_e,\ \forall e \in \mathcal{E}$ **and**

$$\underline{V_e} \le \sum_{e' \in P_e} \sum_{k' \in M\cup\{k\} : e' \in P_{k'}} z^{\mathrm{R}}_{e'} \overline{s}^{\mathrm{R}}_{k'} + z^{\mathrm{I}}_{e'} \overline{s}^{\mathrm{I}}_{k'} \le \overline{V_e},\ \forall e \in \mathcal{E}$$

  then $M \leftarrow M \cup \{k\}$

end for

return M

Evidently, both `GreedyDisDm` and `GreedysOPF` have polynomial running time in n.

3.4.4 Analysis of Approximation Ratio

We first present some intuition for `GreedyDisDm` and `GreedysOPF`.

`GreedyDisDm` groups the customers with similar utilities, whereas `GreedysOPF` finds a solution that maximizes the number of satisfied customers greedily. If `GreedysOPF` can find a solution that is close to the optimal solution, when all customers have the same utility, then `GreedyDisDm` can find a group that approximates the optimal solution in general.

We denote `GreedysOPF` by $\texttt{GreedysOPF}_{\mathsf{C}}$ when solving sOPF_{C} (i.e., $\underline{V_e} \to -\infty$ and $\overline{V_e} \to \infty$), and by $\texttt{GreedysOPF}_{\mathsf{V}}$ when solving sOPF_{V} (i.e., $\overline{\mathsf{S}}_e \to \infty$ for all $e \in \mathcal{E}$).

In the supplementary materials, we also show that sOPF_{V} with both upper and lower voltage constraints are also inapproximable by any efficient algorithm for any approximation gap and any violation bound polynomial in n. Hence, we drop the lower voltage constraints ($\underline{V_e}$), when we analyzing the approximation ratio of `GreedysOPF`.

Analysis of `GreedysOPF`

Although `GreedysOPF` resembles an $O(\log n)^{-1}$-approximation algorithm for unsplittable flow problem provided in Chekuri *et al.* (2006), our proof for the approximation ratio is substantially more involved than those in Chekuri *et al.* (2006), because of the presence of complex-valued demands makes `GreedysOPF` behave very differently.

To analyze the approximation ratio of `GreedysOPF`, we first consider a simple setting where all utilities are identical (i.e., $u_k = 1$ for all $k \in \mathcal{I}$). The objective of sOPF then becomes to maximize the number of satisfied customers.

We will define the following notations:

- A demand path is a path from a customer to the root. Let $\eta \triangleq \max_{e\in\mathcal{E}} |\mathcal{P}_e|$ be the maximum length of any demand path.

- Let $\phi \triangleq \max_{k,k'\in\mathcal{I}} |\angle \bar{s}_k - \angle \bar{s}_{k'}|$ be the maximum angle difference between any pair of demands

- Let $\phi_{\rm zs} \triangleq \max_{k\in\mathcal{I}, e\in P_k} |\angle \overline{s}_k - \angle z_e|$ be the maximum angle difference between demands and line impedance along any path to the root. We assume $0 \le \phi_{\rm zs} < \frac{\pi}{2}$.

- Let $\rho_{\tilde{e}} \triangleq \max_{e,e'\in P_{\tilde{e}}} \frac{|z_e|}{|z_{e'}|}$ be the maximum ratio of impedance magnitude between any pair of edges along the path $P_{\tilde{e}}$.

- Let $\rho \triangleq \max_{\tilde{e}\in\mathcal{E}} \rho_{\tilde{e}}$ be the maximum of all ratios.

Since $0 \le \phi_{\rm zs} < \frac{\pi}{2}$, it necessarily holds that $z_e^{\rm R}\overline{s}_k^{\rm R} + z_e^{\rm I}\overline{s}_k^{\rm I} \ge 0$, for all $k \in \mathcal{I}$ and $e \in \mathcal{E}$. It follows that the left-hand side of Cons. (3.72) on edge e is at least as large as when $e \in \mathcal{L}$ is a leaf edge, where $\mathcal{L}$ is the set of all leaf edges defined by:

$$\mathcal{L} \triangleq \{(i,j) \in \mathcal{E} \mid \nexists k \in \mathcal{V} \text{ such that } (j,k) \in \mathcal{E}\}$$

Therefore, it suffices to consider Cons. (3.72) for each $e \in \mathcal{L}$:

$$\sum_{k\in\mathcal{I}} \Big(\sum_{e'\in P_k\cap P_e} z_{e'}^{\rm R}\overline{s}_k^{\rm R} + z_{e'}^{\rm I}\overline{s}_k^{\rm I} \Big) x_k \le \overline{V_e}, \quad \forall e \in \mathcal{L} \tag{3.73}$$

Theorem 3.11. Consider $u_k = 1$ for all $k \in \mathcal{I}$.

1. `GreedysOPF`$_{\sf C}$ is α-approximation for sOPF$_{\sf C}$, where
$$\alpha = \Big(\big\lfloor \sec\phi \cdot \sec\tfrac{\phi}{2} \big\rfloor + 1 \Big)^{-1}$$

2. `GreedysOPF`$_{\sf V}$ is α-approximation for sOPF$_{\sf V}$, where
$$\alpha = \Big(\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + 1 \Big)^{-1}$$

3. `GreedysOPF` is α-approximation for sOPF, where
$$\alpha = \Big(\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \big\lfloor \sec\phi \cdot \sec\tfrac{\phi}{2} \big\rfloor + 2 \Big)^{-1}$$

Proof. We first present the basic idea as follows. `GreedysOPF` first sorts customers in $\widehat{\mathcal{I}}$ in a non-decreasing order according to the magnitudes of their demands:

$$\overbrace{|\overline{s}_1| \le |\overline{s}_2| \le ...}^{A_1} \le \overbrace{|\overline{s}_t| \le ...}^{B_1} \le \overbrace{|\overline{s}_p| \le ...}^{A_2} \le ... \le \overbrace{|\overline{s}_z| \le ...}^{B_m}$$

We index the customers in such an order by a sequence $\widehat{\mathcal{I}} = (k_1, k_2, ..., k_r)$. `GreedysOPF` attempts to pack their demands greedily sequentially (by placing the satisfied customers into M), if the power capacity constraints or voltage constraints are not violated. Let the sets of customers who can be satisfied consecutively be $(A_1, ..., A_m)$ and the sets of customers who violate Con. (3.71) or (3.73) be $(B_1, ..., B_m)$ (where B_m may be empty).

Denote the optimal solution by $R^* \subseteq \widehat{\mathcal{I}}$, which is the maximal set of satisfied customers whose inelastic demands are satisfied in the optimal allocation. We follow an exchange argument by constructing two sequences of sets $(W_1, ..., W_r)$ and $(R_1, ..., R_r)$, such that the following conditions hold:

1. $W_j \subseteq R^* \cap \bigcup_{i=1}^m B_i$ for each $j = 1, ..., r$.
2. Set $R_j \triangleq (R_{j-1} \backslash W_j) \cup \{k_j\}$ for each $j = 1, ..., r$.
3. Finally, we obtain $R_r = M$.

The size of each $|W_j|$ which will be used to derive the approximation ratio α.

Formally, we define

$$Q_k(e) \triangleq \sum_{e' \in P_k \cap P_e} (z_{e'}^{\mathrm{R}} \overline{s}_k^{\mathrm{R}} + z_{e'}^{\mathrm{I}} \overline{s}_k^{\mathrm{I}})$$

and Cons. (3.73) becomes

$$\sum_{k \in \mathcal{I}} Q_k(e) x_k \le \overline{V_e}, \quad \forall e \in \mathcal{L}$$

Let $R_1 \triangleq (R^* \backslash W_1) \cup \{k_1\}$, such that customer $k_1 \in M$ is added to R^*, and W_1 is removed. Recursively, define $R_j \triangleq (R_{j-1} \backslash W_j) \cup \{k_j\}$ for

$j = 1, ..., r$. For each step j, $W_j \subseteq R_{j-1} \cap \bigcup_{i=1}^m B_i$ is defined to be any *minimal* subset such that R_j is a feasible solution.

Define $W_j^{(1)} \subseteq W_j$ and $W_j^{(2)} \subseteq W_j$ as follows.

$$W_j^{(1)} \triangleq \Bigg\{ k \in W_j \;\Bigg|\; \exists e \in \mathcal{E}, \Bigg| \sum_{k' \in R_j \wedge e \in P_{k'}} \overline{s}_{k'} \Bigg| \leq \overline{\mathsf{S}}_e \text{ and } \Bigg| \sum_{k' \in R_j \cup \{k\} \wedge e \in P_{k'}} \overline{s}_{k'} \Bigg| > \overline{\mathsf{S}}_e \Bigg\} \tag{3.74}$$

$$W_j^{(2)} \triangleq \Bigg\{ k \in W_j \;\Bigg|\; \exists e \in \mathcal{L}, \sum_{k' \in R_j} Q_{k'}(e) \leq \overline{V_e} \text{ and } \sum_{k' \in R_j \cup \{k\}} Q_{k'}(e) > \overline{V_e} \Bigg\} \tag{3.75}$$

See Fig. 3.3 for an illustration.

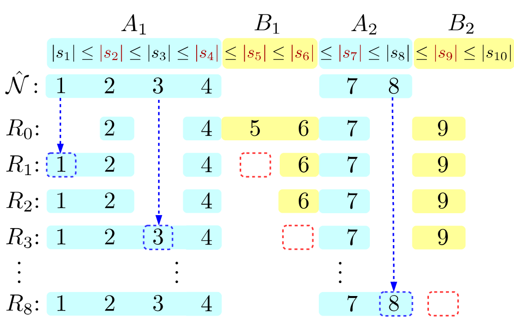


Figure 3.3: A pictorial illustration of the definition of R_j for an instance of 10 customers $\mathcal{I} = \{1, 2, ..., 10\}$.

The approximation ratio α is equivalent to the following bound:

$$|M| \geq \alpha |R^*|$$

The proof is completed by Lemmas 3.12, 3.13, 3.14. $\square$

Lemma 3.12.

1. Consider `GreedysOPF`$_\mathsf{C}$ for sOPF$_\mathsf{C}$.

$$|M| \geq \frac{|R^*|}{\lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2}$$

2. Consider `GreedysOPF`$_\mathsf{V}$ for sOPF$_\mathsf{V}$.

$$|M| \geq \frac{|R^*|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor}$$

3. Consider `GreedysOPF` for sOPF.

$$|M| \geq \frac{|R^*|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2}$$

Proof. By the definition of W_j, we observe the following:

(o1) $|W_j| \leq |W_j^{(1)}| + |W_j^{(2)}|$.

(o2) If $k_j \in R_{j-1}$, then $|W_j| = 0$.

(o3) By (o2), $(W_1, ..., W_r)$ form a partition over $R^*\backslash M$.

Consider `GreedysOPF` for sOPF. By Lemmas 3.13 and 3.14 and (o3), one can relate $|M|$ to the optimal $|R^*|$ as follows.

$$\begin{aligned}
|M| &= |M \cap R^*| + |M\backslash R^*| \\
&= |M \cap R^*| + \sum_{j\in M\backslash R^*} \frac{1}{|W_j|}|W_j| \\
&\geq |M \cap R^*| + \frac{\sum_{j\in M\backslash R^*} |W_j|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2} \\
&= |M \cap R^*| + \frac{|R^*\backslash M|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2} \quad \text{(by (o3))} \\
&\geq \frac{|M \cap R^*| + |R^*\backslash M|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2} \\
&= \frac{|R^*|}{\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2} \rfloor + 2}
\end{aligned}$$

The cases of `GreedysOPF`$_\mathsf{C}$ and `GreedysOPF`$_\mathsf{V}$ are special cases of $W_j = W_j^{(1)}$ and $W_j = W_j^{(2)}$ respectively. □

Lemma 3.13. Define $W_j^{(1)}$ as in Eqn. (3.74). We obtain

$$|W_j^{(1)}| \leq \lfloor \sec\phi \cdot \sec\tfrac{\phi}{2} \rfloor + 1 \tag{3.76}$$

Proof. Assume $|W_j^{(1)}| > 1$. We note that based on the tree topology, all demand paths share a single source (i.e., the root). When adding demand $\overline{s}_{k_j}$ to R_{j-1}, Eqn. (3.74) implies that each element of $W_j^{(1)}$ if added to R_j must cause violation at some (possibly more than one) edges. These violations occur only along the path P_{k_j}. Denote by $E \subseteq P_{k_j}$ the set of edges at which violations occur (after adding some $k \in W_j^{(1)}$ to R_j). Define $e^\circ \in E$ to be the closest edge to the root satisfies $e^\circ \in P_k$ for all $k \in W_j^{(1)}$ because all demands share the same source (see Fig. 3.4(a)). This property allows us to bound $|W_j^{(1)}|$.

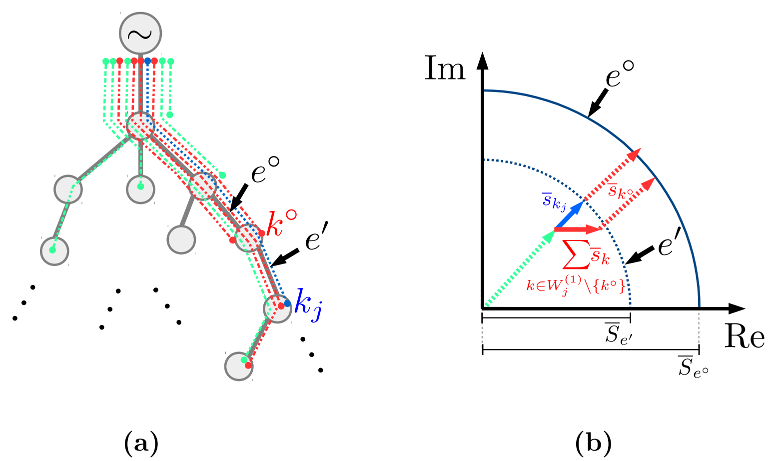


Figure 3.4: (a) The dotted lines illustrate the demand paths in $R_j\setminus\{k_j\}$ (green), $W_j^{(1)}$ (red), and k_j (blue). The paths of $W_j^{(1)}$ and k_j intersect at edge e°. Adding any demand from $W_j^{(1)}$ (red) to R_j (green and blue) will violate the pwoer capacity constraints at $\overline{S}_{e^\circ}$ and $\overline{S}_{e'}$. (b) The horizontal and vertical axes correspond to the real and imaginary components of complex-valued demands respectively. The circular lines visualize Cons. (3.71) at edges e° and e' respectively. The dotted green arrow corresponds to $\sum_{k\in R_j\setminus\{k_j\}} \overline{s}_k$, the solid blue is $\overline{s}_{k_j}$, the solid red is $\sum_{k\in W_j^{(1)}\setminus\{k^\circ\}} \overline{s}_k$, and the dotted red arrows is $\overline{s}_{k^\circ}$ (which is replicated to illustrate the violation).

More specifically, define customer $k^\circ \in W_j^{(1)}$, such that

$$\Big| \sum_{k:k\in R_j, e^\circ \in P_k} \overline{s}_k + \overline{s}_{k^\circ} \Big| > \overline{S}_{e^\circ}$$

See Fig. 3.4(b) for an illustration. Note that

$$\Big| \sum_{k:k\in R_j \wedge e^\circ \in P_k} \overline{s}_k + \overline{s}_{k^\circ} \Big| \;>\; \overline{S}_{e^\circ} \;\geq\; \Big| \sum_{\substack{k:k\in R_j \cup W_j^{(1)} \backslash \{k_j\} \\ \wedge e^\circ \in P_k}} \overline{s}_k \Big| \tag{3.77}$$

Claim 1. Given $k_j \in A_i$ for some i, then $W_j \cap \bigcup_{l=1}^{i-1} B_l = \varnothing$. Thus, for any $k_j \in M$, we have

$$|\overline{s}_{k_j}| \leq |\overline{s}_k| \quad \text{for all} \quad k \in W_j \tag{3.78}$$

We prove Claim 1 as follows. First, since $R_{j-1} \supseteq \cup_{l=1}^{i-1} A_l$, it follows that $R_{j-1} \cap B_l = \varnothing$ for all $l \leq i-1$, because R_{j-1} is a feasible solution. Therefore, $W_j \subseteq \bigcup_{l=i}^{m} B_l$ and $W_j \cap \bigcup_{l=1}^{i-1} B_l = \varnothing$. Then, Eqn. (3.78) follows from the non-decreasing order of demands $|\overline{s}_1| \leq |\overline{s}_2| \leq \dots$.

Hence, we obtain

$$|\overline{s}_{k_j}| \leq \frac{\sum_{k\in W_j^{(1)} \backslash \{k^\circ\}} |\overline{s}_k|}{|W_j^{(1)}| - 1} \tag{3.79}$$

Rearranging Eqn. (3.79) and using the fact that $|W_j^{(1)}|$ is an integer, we apply Lemma 2.2 to obtain

$$\begin{aligned} |W_j^{(1)}| &\leq \left\lfloor \frac{\sum_{k\in W_j^{(1)} \backslash \{k^\circ\}} |\overline{s}_k|}{|\overline{s}_{k_j}|} + 1 \right\rfloor \\ &\leq \left\lfloor \frac{\sec\frac{\phi}{2} \cdot \left| \sum_{k\in W_j^{(1)} \backslash \{k^\circ\}} \overline{s}_k \right|}{|\overline{s}_{k_j}|} \right\rfloor + 1 \end{aligned}$$

Let

$$d_0 = \sum_{k\in R_j \backslash \{k_j\}, e^\circ \in P_k} \overline{s}_k + \overline{s}_{k^\circ}, \quad d_1 = \overline{s}_{k_j}, \quad d_2 = \sum_{k\in W_j^{(1)} \backslash \{k^\circ\}} \overline{s}_k$$

By Eqn. (3.77), it follows that $|d_0 + d_1| > |d_0 + d_2|$.

Next, we apply Lemma 2.2 to bound $\left|\sum_{k \in W_j^{(1)} \setminus \{k^\circ\}} \overline{s}_k\right| / |\overline{s}_{k_j}| \leq \sec\phi$, and obtain

$$|W_j^{(1)}| \leq \lfloor \sec\phi \cdot \sec\tfrac{\phi}{2} \rfloor + 1$$

□

Lemma 3.14. Define $W_j^{(2)}$ as in Eqn. (3.75). We obtain

$$|W_j^{(2)}| \leq \lfloor \eta \cdot \rho \cdot \sec\phi_{\text{zs}} \rfloor + 1. \tag{3.80}$$

Proof. Assume $|W_j^{(2)}| > 1$. Let $k' \in W_j^{(2)}$ be an arbitrary customer. Considering Cons. (3.73), define edge $\tilde{e} \in \mathcal{L}$ such that

$$Q_{k_j}(\tilde{e}) \geq \sum_{k \in W_j^{(2)}} Q_k(\tilde{e}) - Q_{k'}(\tilde{e}). \tag{3.81}$$

Note that $\tilde{e}$ must exist, otherwise

$$Q_{k_j}(e) + Q_{k'}(e) < \sum_{k \in W_j} Q_k(e), \forall e \in \mathcal{L}$$

This implies that $W_j^{(2)}$ is not minimal, namely, $\{k'\} \cup R_j$ is a feasible solution, which contradicts the definition of $W_j^{(2)}$ in Eqn. (3.75). Let $e_{\max} \triangleq \arg\max_{e \in P_{k_j}} z_e^{\text{R}} \overline{s}_{k_j}^{\text{R}} + z_e^{\text{I}} \overline{s}_{k_j}^{\text{I}}$.

Then, for all $k \in W_j^{(2)} \setminus \{k'\}$ and $e \in P_k \cap P_{\tilde{e}}$, we obtain

$$\begin{aligned} z_{e_{\max}}^{\text{R}} \overline{s}_{k_j}^{\text{R}} + z_{e_{\max}}^{\text{I}} \overline{s}_{k_j}^{\text{I}} &= \frac{z_{e_{\max}}^{\text{R}} \overline{s}_{k_j}^{\text{R}} + z_{e_{\max}}^{\text{I}} \overline{s}_{k_j}^{\text{I}}}{z_e^{\text{R}} \overline{s}_k^{\text{R}} + z_e^{\text{I}} \overline{s}_k^{\text{I}}} (z_e^{\text{R}} \overline{s}_k^{\text{R}} + z_e^{\text{I}} \overline{s}_k^{\text{I}}) \\ &\leq \frac{|z_{e_{\max}}| \cdot |\overline{s}_{k_j}|}{|z_e| \cdot |\overline{s}_k| \cdot \cos\phi_{\text{zs}}} (z_e^{\text{R}} \overline{s}_k^{\text{R}} + z_e^{\text{I}} \overline{s}_k^{\text{I}}) \qquad (3.82) \\ &\leq \tfrac{\rho}{\cos\phi_{\text{zs}}} \cdot (z_e^{\text{R}} \overline{s}_k^{\text{R}} + z_e^{\text{I}} \overline{s}_k^{\text{I}}), \qquad (3.83) \end{aligned}$$

where Eqn. (3.82) follows by Cauchy-Schwarz inequality and $\phi_{\text{zs}} \in [0, \frac{\pi}{2})$, Eqn. (3.83) by $\frac{|\overline{s}_{k_j}|}{|\overline{s}_k|} \leq 1$ by Claim 1.

Summing all $k \in W_j^{(2)} \backslash \{k'\}$ and $e \in P_k \cap P_{\tilde{e}}$, we obtain

$$
\begin{aligned}
& z^{\rm R}_{e_{\max}} \overline{s}^{\rm R}_{k_j} + z^{\rm I}_{e_{\max}} \overline{s}^{\rm I}_{k_j} \\
& \le \left(\frac{\rho}{\cos\phi_{\rm zs}}\right) \cdot \frac{\sum_{k \in W_j^{(2)} \backslash \{k'\}} \sum_{e \in P_k \cap P_{\tilde{e}}} z^{\rm R}_e \overline{s}^{\rm R}_k + z^{\rm I}_e \overline{s}^{\rm I}_k}{\sum_{k \in W_j^{(2)} \backslash \{k'\}} |P_k \cap P_{\tilde{e}}|} \\
& \le \left(\frac{\rho}{\cos\phi_{\rm zs}}\right) \cdot \frac{\sum_{k \in W_j^{(2)} \backslash \{k'\}} Q_k(\tilde{e})}{|W_j^{(2)}| - 1}, \qquad (3.84)
\end{aligned}
$$

because of $|P_k \cap P_{\tilde{e}}| \ge 1$. By the definition of $e_{\max}$, we obtain

$$
\begin{aligned}
(z^{\rm R}_{e_{\max}} \overline{s}^{\rm R}_{k_j} + z^{\rm I}_{e_{\max}} \overline{s}^{\rm I}_{k_j}) & \ge \frac{1}{|P_{k_j} \cap P_{\tilde{e}}|} \sum_{e \in P_{k_j} \cap P_{\tilde{e}}} z^{\rm R}_e \overline{s}^{\rm R}_{k_j} + z^{\rm I}_e \overline{s}^{\rm I}_{k_j} \\
& \ge \frac{1}{\eta} \cdot Q_{k_j}(\tilde{e}) \qquad (3.85)
\end{aligned}
$$

By Eqns. (3.81), (3.84), (3.85) and $|W_j^{(2)}|$ as an integer, we obtain

$$
\begin{aligned}
|W_j^{(2)}| & \le \left\lfloor \eta \cdot \left(\frac{\rho}{\cos\phi_{\rm zs}}\right) \frac{\sum_{k \in W_j^{(2)} \backslash \{k'\}} Q_k(\tilde{e})}{Q_{k_j}(\tilde{e})} \right\rfloor + 1 \\
& \le \lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs} \rfloor + 1
\end{aligned}
$$

□

Analysis of `GreedyDisDm`

We complete the analysis of `GreedyDisDm` by the following theorem.

Theorem 3.15. Assume that ρ, $\sec\phi$ and $\sec\phi_{zs}$ are constants, and $\phi, \phi_{zs} < \frac{\pi}{2}$, then

1. $\texttt{GreedyDisDm}_{\mathsf{C}}$ is $\frac{1}{O(\log n)}$-approximation for $\mathrm{SOPF}_{\mathsf{C}}$.
2. $\texttt{GreedyDisDm}_{\mathsf{V}}$ is $\frac{1}{O(\eta \cdot \log n)}$-approximation for $\mathrm{SOPF}_{\mathsf{V}}$.
3. `GreedyDisDm` is $\frac{1}{O(\eta \cdot \log n)}$-approximation for SOPF.

Proof. By rounding utilities in `GreedyDisDm`, $\bar{u}_k \in \{0, ..., n^2\} = \{0, ..., 2^{2\log n}\}$ for all $k \in \mathcal{I}$. Therefore there are at most $2\log n + 1$ groups of users (denoted by $\widehat{\mathcal{I}}_1, ..., \widehat{\mathcal{I}}_{2\log n+1}$ respectively). Let $M_1, ..., M_{2\log n+1}$ be their respective unit-utility solutions, returned by Algorithm 8. Define $\textsc{Opt}$ to be an optimal solution value for sOPF (resp., sOPF_{C}, sOPF_{V}) and R_i^*, $i \in \{1, ..., 2\log n + 1\}$ be the subset of this optimal solution that belongs to group i. Clearly, $u_{\max} \leq \textsc{Opt}$, assuming each load can be individually served (those loads that cannot be individually served can be determined by checking the feasibility of the problem with exactly one load turned on). Define $u(N) \triangleq \sum_{k\in N} u_k$ for any $N \subseteq \mathcal{I}$. The solution returned by `GreedyDisDm` satisfies the following:

$$u(M) = \max_{i\in\{1,...,2\log n+1\}} u(M_i) \geq \frac{\sum_{i=1}^{2\log n+1} u(M_i)}{2\log n + 1} \tag{3.86}$$

Using the fact that $x \geq y\lfloor \frac{x}{y} \rfloor$ for any $x, y \in \mathbb{R}$ and $y \neq 0$, we obtain for $i \in \{1, ..., 2\log n + 1\}$,

$$u(M_i) = \sum_{k\in M_i} u_k \geq L \sum_{k\in M_i} \bar{u}_k \geq \alpha L \sum_{k\in R_i^*} \bar{u}_k \tag{3.87}$$

where α is the approximation ratio of `GreedysOPF` on unit-utility instances.

Next, we use $x \leq y\lfloor \frac{x}{y} \rfloor + y$ for any $x, y \in \mathbb{R}$ and $y \neq 0$ to obtain

$$\begin{aligned} \sum_{k\in R_i^*} u_k &\leq \sum_{k\in R_i^*} (L\bar{u}_k + L) = L \sum_{k\in R_i^*} \bar{u}_k + |R_i^*| L \\ \Rightarrow \quad L \sum_{k\in R_i^*} \bar{u}_k &\geq \sum_{k\in R_i^*} u_k - |R_i^*| L \end{aligned} \tag{3.88}$$

Finally, we complete the proof using Eqns. (3.86)-(3.88):

$$\begin{aligned} u(M) &\geq \frac{\alpha}{2\log n + 1} \sum_{i=1}^{2\log n+1} \Big(\sum_{k\in R_i^*} u_k - |R_i^*| L \Big) \\ &= \frac{\alpha}{2\log n + 1} \Big(\textsc{Opt} - \sum_{i=1}^{2\log n+1} |R_i^*| L \Big) \\ &\geq \frac{\alpha}{2\log n + 1} \Big(\textsc{Opt} - \frac{n \cdot u_{\max}}{n^2}\Big) \\ &\geq \frac{\alpha}{2\log n + 1} \cdot (1 - \tfrac{1}{n}) \cdot \textsc{Opt} \end{aligned}$$

Therefore, we obtain

1. $\texttt{GreedyDisDm}_{\mathsf{C}}$ is $\bar{\alpha}$-approximation for $\mathrm{SOPF}_{\mathsf{C}}$, where

$$\bar{\alpha} = \frac{\Big(\lfloor \sec\phi \cdot \sec\frac{\phi}{2}\rfloor + 1\Big)^{-1}}{2\log n + 1} \cdot (1 - \tfrac{1}{n}) = \frac{1}{O(\log n)}$$

2. $\texttt{GreedyDisDm}_{\mathsf{V}}$ is $\bar{\alpha}$-approximation for $\mathrm{SOPF}_{\mathsf{V}}$, where

$$\bar{\alpha} = \frac{\Big(\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs}\rfloor + 1\Big)^{-1}}{2\log n + 1} \cdot (1 - \tfrac{1}{n}) = \frac{1}{O(\eta \cdot \log n)}$$

3. $\texttt{GreedyDisDm}$ is $\bar{\alpha}$-approximation for SOPF, where

$$\begin{aligned}\bar{\alpha} &= \frac{\Big(\lfloor \eta \cdot \rho \cdot \sec\phi_{\rm zs}\rfloor + \lfloor \sec\phi \cdot \sec\frac{\phi}{2}\rfloor + 2\Big)^{-1}}{2\log n + 1} \cdot (1 - \tfrac{1}{n}) \\ &= \frac{1}{O(\eta \cdot \log n)}\end{aligned}$$

□

3.4.5 Greedy Approximation Algorithm for OPF

To incorporate continuous demands, we first solve the relaxed problem rOPF by relaxing all discrete demands to be continuous as follows.

$$\begin{aligned} (\mathrm{rOPF}) \quad & \max_{x,v,\ell,S} \sum_{k\in\mathcal{I}} u_k x_k, \\ \text{subject to} \quad & \text{Cons. (1.23)-(1.29)}, \\ & x_k \in [0,1], \ \forall k \in \mathcal{I}. \end{aligned}$$

We consider convex relaxation by relaxing Cons. (1.23) to be $\ell_{i,j} \geq \frac{|S_{i,j}|^2}{v_i}$. Let the solution be $\tilde{x} = \Big((\tilde{x}_k)_{k\in\mathcal{F}}, (\tilde{x}_k)_{k\in\mathcal{I}}\Big)$.

We next define a simplified residual problem $\mathrm{SIOPF}_\delta[\tilde{x}]$ by assuming the continuous demands are set according to $(\tilde{x}_k)_{k\in\mathcal{F}}$, and the links capacity are reduced by a factor of $(1-\delta)$ for a given $\delta \in (0,1)$:

$$
\begin{aligned}
(\textsc{siOPF}_\delta[\tilde{x}]) \quad & \max_{x} \sum_{k\in\mathcal{I}} u_k x_k \\
\text{subject to} \quad & \Big| \sum_{k:e\in P_k} \overline{s}_k x_k \Big| \le (1-\delta)\cdot \overline{S}_e, && \forall e\in\mathcal{E}, \quad (3.89)\\
& \sum_{k\in\mathcal{I}} \Big(\sum_{e'\in P_k\cap P_e} z^{\mathrm{R}}_{e'} \overline{s}^{\mathrm{R}}_k + z^{\mathrm{I}}_{e'} \overline{s}^{\mathrm{I}}_k \Big) x_k \le \overline{V_e}, && \forall e\in\mathcal{E}, \quad (3.90)\\
& x_k \in \{0,1\}, && \forall k\in\mathcal{I}, \quad (3.91)\\
& x_k = \tilde{x}_k, && \forall k\in\mathcal{F}. \quad (3.92)
\end{aligned}
$$

Then, we solve $\textsc{siOPF}_\delta[\tilde{x}]$ by `GreedyDisDm`. To verify the feasibility of a solution $\bar{x}$ by `GreedyDisDm`, we consider the following problem with given demands $\bar{x}$:

$$
\begin{aligned}
(\mathrm{OPF}[\bar{x}]) \quad & \min_{v,\ell,S} \sum_{e\in\mathcal{E}} |z_e|\cdot \ell_e, \\
\text{subject to} \quad & \text{Cons. (1.24)-(1.29)}, \\
& \ell_{i,j} \ge \frac{|S_{i,j}|^2}{v_i}, && \forall (i,j)\in\mathcal{E}, \\
& x_k = \bar{x}_k, && \forall k\in\mathcal{I}.
\end{aligned}
$$

Algorithm 10 `GreedyOPF`$[(u_k, \overline{s}_k)_{k\in\mathcal{I}}, \epsilon]$

1: $\delta \leftarrow 0$
2: **repeat**
3: $\quad \tilde{x} \leftarrow$ Solution of $\textsc{rOPF}$
4: $\quad M \leftarrow$ `GreedyDisDm` on $\textsc{siOPF}_\delta[\tilde{x}]$
5: $\quad$ **for** $k \in \mathcal{I}$ **do**
6: $\quad\quad \bar{x}_k \triangleq \begin{cases} 1 & \text{if } k\in M \\ \tilde{x}_k & \text{if } k\in\mathcal{F} \\ 0 & \text{otherwise} \end{cases}$
7: $\quad$ **end for**
8: $\quad \delta \leftarrow \delta + \epsilon$
9: **until** $\mathrm{OPF}[\bar{x}]$ is feasible
10: **return** $\bar{x}$

`GreedyOPF` (Algorithm 10) is a heuristic to obtain a feasible solution to OPF with both discrete and continuous demands. `GreedyOPF` has polynomial running time in n, because OPF$[\bar{x}]$ can be solved efficiently via convex optimization. However, the theoretical approximation ratio of `GreedyOPF` with respect to OPF is not easy to obtain.

3.5 Notes

Traditionally, Optimal Power Flow (OPF) problems have been tackled by relying on heuristics or general numerical solver, which lack optimality guarantees or efficient running time. Recently, there has been a major progress on tackling OPF efficiently using convex relaxations (Jabr, 2006; Bai *et al.*, 2008; Bose *et al.*, 2015; Huang *et al.*, 2017; Gan *et al.*, 2015; Low, 2014a; Low, 2014b). These papers mainly consider *radial* (i.e., tree) networks and derive sufficient conditions under which the convex relaxation is exact (i.e., equivalent to the original non-relaxed problem); for example, relaxing the rank-1 constraint in the semidefinite programming (SDP) formulation (Bose *et al.*, 2015), or relaxing the equality constraints in the second order cone programming (SOCP) formulation (Huang *et al.*, 2017; Gan *et al.*, 2015; Low, 2014a; Low, 2014b). However, these results yield polynomial time algorithms for OPF with *continuous* power injection constrains, i.e., the control of power injection can be partially satisfied (precisely, as fractional control decision variables).

The algorithms for combinatorial optimization on OPF are studied in Karapetyan *et al.* (2018), Elbassioni *et al.* (2017), and Khonji *et al.* (2017). Karapetyan *et al.* (2018) presents a greedy algorithm for DistFlow model. Elbassioni *et al.* (2017) presents a QPTAS for line networks with scalable network size. Khonji *et al.* (2017) presents PTAS for constant-sized tree networks.

On the other hand, OPF with discrete demands is related to Unsplittable Flow Problem (UFP). UFP has also received considerable attention and approximation algorithms are known for different variants. In directed graphs, the best known approximation is $O(\min\{\sqrt{m}, n^{\frac{2}{3}} \log^{\frac{1}{3}} n\})$ (Kleinberg, 1996; Varadarajan and Venkataraman, 2004), while it is NP-Hard to approximate within $\Omega(n^{\frac{1}{2}-\epsilon})$ (Guruswami *et al.*, 2003), where n and m are the number of nodes and edges respectively. Azar and Regev (2001) shows that UFP in directed graphs is $\Omega(n^{1-\epsilon})$-hard unless P = NP. In undirected graphs, there is an $O(\sqrt{n})$-approximation (Chekuri *et al.*, 2006), and the best known hardness result is $\Omega(\log^{\frac{1}{2}-\epsilon} n)$ assuming NP $\not\subseteq$ ZPTIME$(n^{O(\text{polylog}(n))})$ (Andrews *et al.*, 2005). These hardness results suggest that the problem is difficult to solve in general graphs. For tree topology, the problem is APX-Hard (i.e., hard to approximate within a constant factor) even when demands are uniform (Chekuri *et al.*, 2007). Chekuri *et al.* (2009) obtained an $O(\log n)$-approximation. Recently, Anagnostopoulos *et al.* (2014) obtained a $2+\epsilon$-approximation for path topology.

4

Scheduling of AC Electric Power

We extend the basic setting of single-capacitated AC electric power systems to consider the scheduling problem of discrete demands over a constant time horizon. We consider a discrete time horizon denoted by $\mathcal{T} \triangleq \{1, ..., m\}$. At each time slot $t \in \mathcal{T}$, the generation capacity of the power grid is denoted by $\overline{S}_t \in \mathbb{R}^+$. Denote $\mathcal{N} \triangleq \{1, ..., n\}$ by the set of all users. Each user $k \in \mathcal{N}$ declares a set of demand preferences indexed by the set D_k. Each demand $j \in D_k$ is defined over a time interval $T_j \subseteq \mathcal{T}$, that is, $T_j = \{t_1, t_1 + 1, ..., t_2\}$ where $t_1, t_2 \in \mathcal{T}$ and $t_1 \leq t_2$. Demand j is also associated with a set of complex numbers $\{\overline{s}_{k,j}(t)\}_{t \in T_j}$ where $\overline{s}_{k,j}(t) \triangleq \overline{s}^{\mathrm{R}}_{k,j}(t) + \mathbf{i}\overline{s}^{\mathrm{I}}_{k,j}(t) \in \mathbb{C}$ is a complex power demand at time t. A positive utility $u_{k,j}$ is associated with each user demand (k, j) if satisfied.

4.1 Preliminaries of Multi-Choice Knapsack Problem

Our scheduling problem of AC power demands is related to a general setting of knapsack problem, known as *multi-choice* knapsack problem. Denote the set of feasible choices of each user k by D_k. Consider positive real demand $\hat{s}_{k,j}$ for all $k \in \mathcal{I}$, $j \in D_k$. We define multi-choice knapsack problem (MKP) as follows.

$$
\begin{aligned}
\text{(MKP)} \quad & \max_{x} \sum_{k\in\mathcal{I}} \sum_{j\in D_k} u_{k,j} x_{k,j} \\
\text{subject to} \quad & \sum_{k\in\mathcal{I}} \sum_{j\in D_k} \hat{s}_{k,j} x_{k,j} \le \overline{S}, && && (4.1)\\
& \sum_{j\in D_k} x_{k,j} \le 1, && \forall k \in \mathcal{I}, && (4.2)\\
& x_{k,j} \in \{0,1\}, && \forall k \in \mathcal{I}, && (4.3)
\end{aligned}
$$

where Cons. (4.2) is known as the multi-choice constraint that restricts the multiplicity of satisfiable selections of each user.

In the context of scheduling of AC power demands, D_k is defined with respect to the scheduling constraints of each user. Some user demands are inelastic, denoted by $\mathcal{I} \subseteq \mathcal{N} \times \bigcup_k D_k$, which are required to be either fully satisfied or fully dropped. An example is an appliance that should be either supplied with a fixed amount of power, or switched off. The rest of demands, denoted by $\mathcal{F} \subseteq \mathcal{N} \times \bigcup_k D_k$ such that $\mathcal{F} \cap \mathcal{I} = \varnothing$, are elastic demands, which can be partially satisfied. The goal is to decide a solution of control variables $(x_{k,j})_{k\in\mathcal{N}, j\in D_k} \in [0,1]^n$ that maximizes the total utility of satisfiable users subject to the generation capacity over time.

Extending MKP, we define the Complex-demand Scheduling Problem over m discrete time slots (m-CSP) by the following mixed integer programming problem.

$$
\begin{aligned}
(m\text{-CSP}) \quad & \max \sum_{k\in\mathcal{N}} \sum_{j\in D_k} u_{k,j} x_{k,j} \\
\text{subject to} \quad & \Big| \sum_{k\in\mathcal{N}} \sum_{j\in D_k : T_j \ni t} \overline{s}_{k,j}(t) \cdot x_{k,j} \Big| \le \overline{S}_t, && \forall t \in \mathcal{T}, && (4.4)\\
& \sum_{j\in D_k} x_{k,j} \le 1, && \forall k \in \mathcal{N}, && (4.5)\\
& x_{k,j} \in \{0,1\}, && \forall (k,j) \in \mathcal{I}, && (4.6)\\
& x_{k,j} \in [0,1], && \forall (k,j) \in \mathcal{F}, && (4.7)
\end{aligned}
$$

where $|\mu|$ denotes the magnitude of $\mu \in \mathbb{C}$. Cons. (4.4) captures the capacity constraint, and Cons. (4.5) forces at most one inelastic demand for every user to be selected.

We consider the following assumptions: for any user k,

(i) $|D_k| = 1$ if $(k, j) \in \mathcal{F}, j \in D_k$; and

(ii) all demands $\overline{s}_{k,j}(t)$, $j \in D_k$ reside in one quadrant of the complex plane.

Note that 1-CSP (i.e., $|\mathcal{T}| = 1$) is complex-demand knapsack problemCKP. (We drop subscripts t and j when $|\mathcal{T}| = 1$ and $|D_k| = 1$ for all $k \in \mathcal{N}$.) We write m-CSP$[\phi_1, \phi_2]$ for the restriction of problem m-CSP subject to $\phi_1 \le \max_{k\in\mathcal{N}} \arg(\overline{s}_k) \le \phi_2$, where $\arg(\overline{s}_k) \ge 0$ for all $k \in \mathcal{N}$.

4.2 PTAS

In this section we assume the number of time slots $|\mathcal{T}|$ is a constant. This assumption is practical in the realistic setting, where users declare their demands on hourly basis one day ahead in the electric market. We remark that the results in this section do not require T_j to be a continuous interval in $\mathcal{T}$.

4.2.1 PTAS for m-CSP$[0, \frac{\pi}{2}]$

Define a convex relaxation of m-CSP (denoted by RCSP), such that Cons. (4.6) are replaced by $x_{k,j} \in [0, 1]$ for all $(k, j) \in \mathcal{I} \cup \mathcal{F}$. We define another convex relaxation that will be used in the PTAS denoted by RCSP$[I_1, I_0]$ which is equivalent to RCSP, subject to partial substitution such that $x_{k,j} = 1$, for all $(k, j) \in I_1$ and $x_{k,j} = 0$, for all $(k, j) \in I_0$, where $I_1 \cap I_0 = \varnothing$.

The above relaxation can be solved approximately in polynomial time using standard convex optimization algorithms. In fact, such algorithms can find a feasible solution $x^{\rm cx}$ to the convex relaxation such that $u(x^{\rm cx}) \ge \textsc{Opt}^\star - \delta$, in time polynomial in the input size (including the bit complexity) and $\log \frac{1}{\delta}$, where $\textsc{Opt}^\star$ is the optimal objective value of RCSP$[I_1, I_0]$. Notice that $\textsc{Opt}^\star \ge \bar{u} \triangleq \max_{k,j} u_{k,j}$, setting δ to $\frac{\epsilon}{2} \cdot \bar{u}$ assures that $u(x^{\rm cx}) \ge (1 - \frac{\epsilon}{2}) \cdot \textsc{Opt}^\star$.

$$
\begin{aligned}
(\text{rCSP}[I_1, I_0]) \max_{x_{k,j}\in[0,1]} & \sum_{k\in\mathcal{N}} \sum_{k\in D_k} u_{k,j} x_{k,j} \\
\text{subject to } & \Big(\sum_{k\in\mathcal{N}} \sum_{j\in D_k: t\in T_j} \overline{s}^{\mathrm{R}}_{k,j}(t)\cdot x_{k,j} \Big)^2 \\
& + \Big(\sum_{k\in\mathcal{N}} \sum_{j\in D_k: t\in T_j} \overline{s}^{\mathrm{I}}_{k,j}(t)\cdot x_{k,j} \Big)^2 \le \overline{S}^2_t,\ \forall t\in\mathcal{T}, \quad (4.8) \\
& \sum_{j\in D_k} x_{k,j} \le 1,\ \forall k\in\mathcal{N}, \quad (4.9) \\
& x_{k,j} = 1,\ \forall (k,j)\in I_1, \quad (4.10) \\
& x_{k,j} = 0,\ \forall (k,j)\in I_0. \quad (4.11)
\end{aligned}
$$

We provide a $(1-\epsilon, 1)$-approximation for m-CSP$[0, \frac{\pi}{2}]$ in Algorithm 11, denoted by `CSP-PTAS`. The idea of `CSP-PTAS` is based on that proposed in Elbassioni and Nguyen (2015) with two extensions. First, we consider multiple demands per user. This in fact adds n extra constraints to that in Elbassioni and Nguyen (2015), and thus the rounding procedure requires further analysis. The second extension is the addition of elastic demands $\mathcal{F}$.

Given a feasible solution $\tilde{x}$ to rCSP$[I_1, I_0]$, a restricted set of demands $R \subseteq \mathcal{I} \cup \mathcal{F}$, and vectors $c^1, c^2 \in \mathbb{R}^{+m}$, we define the following relaxation, denoted by LP$[c^1, c^2, \tilde{x}, R]$.

`CSP-PTAS` proceeds as follows. We guess $I_1 \subseteq \mathcal{I}$ to be the set of largest-utility $\frac{8m}{\epsilon}$ inelastic demands in the optimal solution; this defines an excluded set of demands $I_0 \subseteq \mathcal{I} \setminus I_1$ whose utilities exceed one of the utilities in I_1 (Step 4). For each such I_1 and I_0, we solve the convex program rCSP$[I_1, I_0]$ and obtain a $(1-\frac{\epsilon}{2})$-approximation x^{cx} (note that the feasibility of the convex program is guaranteed by the conditions in Step 3). The real and imaginary projections over all time slots of solution x^{cx}, denoted by $L^{\mathrm{R}} \in \mathbb{R}^m_+$ and $L^{\mathrm{I}} \in \mathbb{R}^m_+$, are used to define the linear program LP$[L^{\mathrm{R}}, L^{\mathrm{I}}, x^{\text{cx}}, \mathcal{F} \cup I_1 \cup I_0]$ over the restricted set of demands $\mathcal{F} \cup I_1 \cup I_0$. We solve the linear program in Step 10, and then round down the solution corresponding to demands $(k,j) \in \mathcal{I}$ in Step 11. Finally, we return a solution $\hat{x}$ that attains the maximum utility among all obtained solutions.

$$
\begin{aligned}
(\mathrm{LP}[c^1, c^2, \tilde{x}, R]) \quad & \max_{x_{k,j} \in [0,1]} \sum_{k \in \mathcal{N}} \sum_{j \in D_k} u_{k,j} x_{k,j} && && (4.12)\\
\text{subject to} \quad & \sum_{k \in \mathcal{N}} \sum_{j \in D_k : t \in T_j} \overline{s}^{\mathrm{R}}_{k,j}(t) \cdot x_{k,j} \le c^1_t, && \forall t \in \mathcal{T}, && (4.13)\\
& \sum_{k \in \mathcal{N}} \sum_{j \in D_k : t \in T_j} \overline{s}^{\mathrm{I}}_{k,j}(t) \cdot x_{k,j} \le c^2_t, && \forall t \in \mathcal{T}, && (4.14)\\
& \sum_{j \in D_k} x_{k,j} \le 1, && \forall k \in \mathcal{N}, && (4.15)\\
& x_{k,j} = \tilde{x}_{k,j}, && \forall (k,j) \in R. && (4.16)
\end{aligned}
$$

Theorem 4.1. For any fixed ϵ, `CSP-PTAS` obtains a $(1-\epsilon, 1)$-approximation in polynomial time.

Proof. One can easily see that the running time of `CSP-PTAS` is polynomial in size of the input, for any given ϵ. We now argue that the solution $\hat{x}$ is $(1-\epsilon)$-approximation for m-CSP$[0, \frac{\pi}{2}]$. Let $x^\star$ be an optimal solution for m-CSP$[0, \frac{\pi}{2}]$ of utility $\textsc{Opt} \triangleq u(x^\star)$. Define $X^\star \triangleq \{(k,j) \in \mathcal{I} \mid x^\star_{k,j} = 1\}$. By the feasibility of $x^\star$, in Step 5 the algorithm obtains

$$
u(x^{\mathrm{cx}}) \ge (1 - \tfrac{\epsilon}{2}) \cdot \textsc{Opt}^\star \ge (1 - \tfrac{\epsilon}{2}) \cdot \textsc{Opt}, \tag{4.17}
$$

where $\textsc{Opt}^\star$ is the optimal value of rCSP$[I_1, I_0]$ for some I_1 equal to the highest $\frac{8m}{\epsilon}$ utility demands in $X^\star$, and $I_0 \cap X^\star = \varnothing$. If $|X^\star| \le \frac{8m}{\epsilon}$, then obviously $\hat{x} = x^{''} = x^{\mathrm{cx}} \ge (1 - \frac{\epsilon}{2})\textsc{Opt}$.

Now suppose $|X^\star| > \frac{8m}{\epsilon}$. Observe that x^{cx} (a solution of rCSP$[I_1, I_0]$) is also a feasible solution for LP$[L^{\mathrm{R}}, L^{\mathrm{I}}, x^{\mathrm{cx}}, \mathcal{F} \cup I_1 \cup I_0]$ (Cons. (4.13)-(4.16) are tight when x^{cx} is substituted). Therefore, the optimal solution $x^{''}$ of LP$[L^{\mathrm{R}}, L^{\mathrm{I}}, x^{\mathrm{cx}}, \mathcal{F} \cup I_1 \cup I_0]$ satisfies

$$
u(x^{''}) \ge u(x^{\mathrm{cx}}). \tag{4.18}
$$

By Lemma 4.2 below, LP$[L^{\mathrm{R}}, L^{\mathrm{I}}, x^{\mathrm{cx}}, \mathcal{F} \cup I_1 \cup I_0]$ has a basic feasible solution (BFS) with at most $4m$ fractional components, and for any fractional component (k,j), $u_{k,j} < \min_{(k',j') \in I_1} u_{k',j'} \le \frac{\sum_{(k',j') \in I_1} u_{k',j'}}{|I_1|}$.

Algorithm 11 CSP-PTAS$[(u_{k,j}, \{\overline{s}_{k,j}(t)\}_{t\in T_j})_{k\in\mathcal{N},j\in D_k}, (\overline{S}_t)_{t\in\mathcal{T}}, \epsilon]$

Ensure: $(1-\epsilon,1)$-solution $\hat{x}$ to m-CSP$[0,\frac{\pi}{2}]$

1: $\hat{x} \leftarrow \mathbf{0}$

2: **for** each set $I_1 \subseteq \mathcal{I}$ such that $|I_1| \le \frac{8m}{\epsilon}$ **do**

3: **if**

$$\Big|\sum_{(k,j)\in I_1 : t\in T_j} \overline{s}_{k,j}(t)\Big| \le \overline{S}_t, \forall t \in \mathcal{T} \text{ and } \sum_{j\in D_k} x_{k,j} \le 1, \forall k \in \mathcal{N}$$

then

4: $I_0 \leftarrow \{(k,j) \in \mathcal{I} \setminus I_1 \mid u_{k,j} > \min_{(k',j')\in I_1} u_{k',j'}\}$

5: $x^{\mathrm{cx}} \leftarrow$ Solution of rCSP$[I_1, I_0]$

▷ Obtain a $(1-\frac{\epsilon}{2})$-approximation

6: **for all** $t \in \mathcal{T}$ **do**

7: $L^{\mathrm{R}}_t \leftarrow \sum_{k\in\mathcal{N}}\sum_{j\in D_k : t\in T_j} \overline{s}^{\mathrm{R}}_{k,j}(t)\cdot x^{\mathrm{cx}}_{k,j}$

8: $L^{\mathrm{I}}_t \leftarrow \sum_{k\in\mathcal{N}}\sum_{j\in D_k : t\in T_j} \overline{s}^{\mathrm{I}}_{k,j}(t)\cdot x^{\mathrm{cx}}_{k,j}$

9: **end for**

10: $x'' \leftarrow$ Solution of LP$[L^{\mathrm{R}}, L^{\mathrm{I}}, x^{\mathrm{cx}}, \mathcal{F}\cup I_1 \cup I_0]$

▷ Round down the LP solution

11: $\bar{x} \leftarrow \{(\bar{x}_{k,j})_{k\in\mathcal{N},j\in D_k} \mid \bar{x}_{k,j} = \lfloor x''_{k,j} \rfloor$ for $(k,j)\in\mathcal{I}$, and $\bar{x}_{k,j} = x''_{k,j}$ for $(k,j)\in\mathcal{F}\}$

12: **if** $u(\bar{x}) > u(\hat{x})$ **then**

13: $\hat{x} \leftarrow \bar{x}$

14: **end if**

15: **end if**

16: **end for**

17: **return** $\hat{x}$

Therefore, rounding down x'' in Step 11 obtains,

$$u(\hat{x}) \ge u(x'') - 4m\frac{\sum_{(k,j)\in I_1} u_{k,j}}{|I_1|} \ge (1-\tfrac{\epsilon}{2})u(x'')$$
$$\ge (1-\tfrac{\epsilon}{2})^2\cdot \textsc{Opt} \ge (1-\epsilon)\cdot\textsc{Opt},$$

where the last inequity follows by Eqns. (4.17)-(4.18). It remains to show that $\hat{x}$ is feasible. Since $\hat{x}$ is obtained by rounding down x'' (Step. 11),

$$\begin{aligned}
&\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{R}}_{k,j}(t)\cdot\hat{x}_{k,j}\Big)^2+\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{I}}_{k,j}(t)\cdot\hat{x}_{k,j}\Big)^2\\
\le&\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{R}}_{k,j}(t)\cdot x''_{k,j}\Big)^2+\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{I}}_{k,j}(t)\cdot x''_{k,j}\Big)^2\\
\le&(L^{\mathrm{R}}_t)^2+(L^{\mathrm{I}}_t)^2\\
=&\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{R}}_{k,j}(t)x^{\mathrm{cx}}_{k,j}\Big)^2+\Big(\sum_{k\in\mathcal{N}}\sum_{j\in D_k:t\in T_j}\overline{s}^{\mathrm{I}}_{k,j}(t)x^{\mathrm{cx}}_{k,j}\Big)^2\le\overline{S}^2_t,
\end{aligned}\tag{4.19}$$

where Eqn. (4.19) follows by the feasibility of x'' and x^{cx} respectively. Hence, Cons. (4.4) are satisfied. Finally, since some components of x'' in Step 11 are only rounded down, Cons. (4.5)-(4.6) are also satisfied. □

Lemma 4.2 (Patt-Shamir and Rawitz (2010)). Let x be a basic feasible solution (BFS) for LP$[c^1, c^2, x^\star, R]$. Then x has at most $4m$ non-integral components.

Proof. Let h be the number of users k such that $\sum_{j\in D_k} x_{k,j} = 1$. By the properties of a BFS (see, e.g., Grötschel *et al.*, 1988; Schrijver, 1986), the number of strictly positive components in x is at most $2m + h$. Furthermore, constraints (4.15) impose that for each $k \in \mathcal{N}$ among those h users, there is a $j \in D_k$ such that $x_{k,j} > 0$. The remaining $2m$ positive variables can belong to at most $2m$ of the constraints (4.15), implying that at least $\max\{h - 2m, 0\}$ variables are set to 1. It follows that the total number variables taking non-integral values is at most $2m + h - \max\{h - 2m, 0\} \le 4m$. □

4.3 Resource-augmented FPTAS

In the previous section, we have restricted our attention to the setting where all demands lie in the positive quadrant of the complex plane (i.e., m-CSP$[0, \frac{\pi}{2}]$). In this section, we extend our study to the second quadrant (m-CSP$[0, \pi - \varepsilon]$) for any arbitrary small constant $\varepsilon > 0$, that is, we assume $\arg(s_{k,j}(t)) \le \pi - \varepsilon$ for all $k \in \mathcal{N}, j \in D_k, t \in T_j$.

For convenience, we let $\theta = \max\{\phi - \frac{\pi}{2}, 0\}$ (see Fig. 4.1 for an illustration). We present a $(1, 1+\epsilon)$-approximation for m-CSP$[0, \pi - \varepsilon]$ in Algorithm 12, denoted by `CSP-bFPTAS`, that is polynomial in both $\frac{1}{\epsilon}$ and n (i.e., FPTAS). We assume that $\tan\theta$ is bounded by a polynomial in n; without this assumption, a resource-augmented FPTAS is unlikely to exist (see Sec. 5.2).

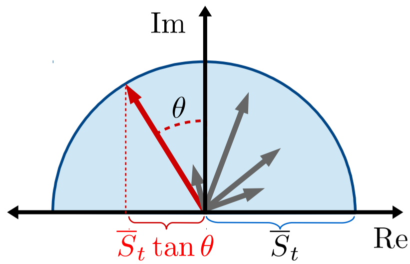


Figure 4.1: We measure $\theta = \phi - \frac{\pi}{2}$ from the imaginary axis.

Let $\mathcal{I}_+ \triangleq \{k \in \mathcal{I} \mid s^{\mathrm{R}}_{k,j}(t) \geq 0, \forall j \in D_k,\ t \in T_j\}$ and $\mathcal{I}_- \triangleq \{k \in \mathcal{I} \mid s^{\mathrm{R}}_{k,j}(t) < 0, \forall j \in D_k,\ t \in T_j\}$ be the subsets of users with demands in the first and second quadrants respectively. Note that $\mathcal{I}_+$ and $\mathcal{I}_-$ partition the set of users $\mathcal{I}$. Consider any solution $\widehat{x}$ to m-CSP$[0, \pi - \varepsilon]$. The basic idea of Algorithm `CSP-bFPTAS` is to enumerate the guessed total projections on real and imaginary axes of all time slots for $\sum_{k \in \mathcal{I}_+} \sum_{j \in D_k : t \in T_j} \widehat{x}_{k,j} \overline{s}_{k,j}(t)$ and $\sum_{k \in \mathcal{I}_-} \sum_{j \in D_k : t \in T_j} \widehat{x}_{k,j} \overline{s}_{k,j}(t)$ respectively. We can use $\tan\theta$ to upper bound the total projections for any feasible solution $\widehat{x}$ as follows, for all $t \in \mathcal{T}$:

$$\sum_{k \in \mathcal{I}} \sum_{j \in D_k : t \in T_j} \overline{s}^{\mathrm{I}}_{k,j}(t) \cdot \widehat{x}_{k,j} \leq \overline{S}_t,$$
$$\sum_{k \in \mathcal{I}_-} \sum_{j \in D_k : t \in T_j} -\overline{s}^{\mathrm{R}}_{k,j}(t) \cdot \widehat{x}_{k,j} \leq \overline{S}_t \tan\theta,$$
$$\sum_{k \in \mathcal{I}_+} \sum_{j \in D_k : t \in T_j} \overline{s}^{\mathrm{R}}_{k,j}(t) \cdot \widehat{x}_{k,j} \leq \overline{S}_t (1 + \tan\theta). \tag{4.20}$$

See Fig. 4.1 for a pictorial illustration.

We then solve two separate multi-dimensional knapsack problems of dimension $2m$ (denoted by $2m$DKP), to find subsets of demands that satisfy the individual guessed total projections. But since $2m$DKP is generally NP-hard, we need to round-up the demands to get a problem that can be solved efficiently by dynamic programming. We show that the violation of the optimal solution to the rounded problem with respect to to the original problem is small in ϵ.

Next, we describe the rounding in detail. First, we define $L_t \triangleq \frac{\epsilon \overline{S}_t}{n(\tan\theta+1)}$, for all $t \in \mathcal{T}$ such that the new rounded-up demands $\widehat{s}_{k,j}(t)$ are defined by:

$$\widehat{s}_{k,j}(t) = \widehat{s}^{\rm R}_{k,j}(t) + \mathbf{i}\widehat{s}^{\rm I}_{k,j}(t) \triangleq \begin{cases} \left\lceil \frac{\overline{s}^{\rm R}_{k,j}(t)}{L_t} \right\rceil \cdot L_t + \mathbf{i} \left\lceil \frac{\overline{s}^{\rm I}_{k,j}(t)}{L_t} \right\rceil \cdot L_t, & \text{if } s^{\rm R}_{k,j}(t) \geq 0, \\ \left\lfloor \frac{\overline{s}^{\rm R}_{k,j}(t)}{L_t} \right\rfloor \cdot L_t + \mathbf{i} \left\lceil \frac{\overline{s}^{\rm I}_{k,j}(t)}{L_t} \right\rceil \cdot L_t, & \text{otherwise.} \end{cases} \tag{4.21}$$

We also define $R \triangleq \frac{\epsilon}{3n(\tan\theta+1)^2}$, such that the values of any elastic $x_{k,j}$, $(k,j) \in \mathcal{F}$ are selected from the discrete set $\mathcal{R}$ of integer multiples of R defined by

$$\mathcal{R} \triangleq \left\{0, 1R, 2R, ..., (\lceil \tfrac{1}{R} \rceil - 1)R, 1\right\}.$$

Let $\xi_+ \in \mathbb{R}^m_+$ (and $\xi_- \in \mathbb{R}^m_+$), $\zeta_+ \in \mathbb{R}^m_+$ (and $\zeta_- \in \mathbb{R}^m_+$) be respectively the guessed real and imaginary absolute total projections of the rounded optimal solution.

Then the possible values of ξ_+, ξ_-, ζ_+ and ζ_- in each component t are integer mutiples of $(R \cdot L_t)$:

$$\begin{aligned} \xi_+(t) \in \mathcal{A}_+(t) &\triangleq \left\{0, (RL_t), 2(RL_t), \ldots, \left\lceil \frac{\overline{S}_t(1+\tan\theta)}{RL_t} \right\rceil \cdot (RL_t)\right\}, \\ \xi_-(t) \in \mathcal{A}_- &\triangleq \left\{0, (RL_t), 2(RL_t), \ldots, \left\lceil \frac{\overline{S}_t \cdot \tan\theta}{RL_t} \right\rceil \cdot (RL_t)\right\}, \\ \zeta_+(t), \zeta_-(t) \in \mathcal{B}(t) &\triangleq \left\{0, (RL_t), 2(RL_t), \ldots, \left\lceil \frac{\overline{S}_t}{RL_t} \right\rceil \cdot (RL_t)\right\}. \end{aligned} \tag{4.22}$$

The next step is to solve the rounded instance exactly. Assume an arbitrary order on $\mathcal{I} = \{1, ..., n\}$. We use recursion to define a table, with each entry $U(k, c^1, c^2)$, $c^1, c^2 \in \mathbb{R}^m_+$, as the maximum utility obtained from a subset of users $\{1, 2, \ldots, K\} \subseteq \mathcal{I}$ with demands $\{\widehat{s}_{k,j}(t)\}_{k \in \{1,...,K\}, j \in D_k, t \in T_j}$ that can fit exactly (i.e., satisfies the capacity constraints as equations) within capacities $\{c_t^1\}_{t=1,\ldots,m}$ on the real axis and $\{c_t^2\}_{t=1,\ldots,m}$ on the imaginary axis. We denote by $2m$DKP-Exact$[\cdot]$ the algorithm for solving exactly the rounded $2m$DKP by dynamic programming. We provide the detailed description of $2m$DKP-Exact$[\cdot]$ in Algorithm 13.

Algorithm 12 `CSP-bFPTAS`$[(u_{k,j}, \{\overline{s}_{k,j}(t)\}_{t\in T_j})_{k\in\mathcal{I},j\in D_k}, (\overline{S}_t)_{t\in\mathcal{T}}, \epsilon]$

Require: Users' utilities and demands $\{u_{k,j}, \{\overline{s}_{k,j}(t)\}_{t\in T_j}\}_{k\in\mathcal{I},j\in D_k}$; capacity over time $\overline{S}_t$; accuracy parameter ϵ

Ensure: $(1, 1+4\epsilon)$-solution $\widehat{x}$ to m-CSP$[0, \pi - \varepsilon]$

$\widehat{x} \leftarrow \mathbf{0}$
for all $\overline{s}_{k,j}(t)$, $k \in \mathcal{I}$, $j \in D_k$, and $t \in T_j$ **do**
  Set $\widehat{s}_{k,j}(t) \leftarrow \widehat{s}^{\mathrm{R}}_{k,j}(t) + \mathbf{i}\widehat{s}^{\mathrm{I}}_{k,j}(t)$ as defined by (4.21)
end for
for all $\xi_+ \in \prod_{t\in\mathcal{T}} \mathcal{A}_+(t), \xi_- \in \prod_{t\in\mathcal{T}} \mathcal{A}_-(t), \zeta_+, \zeta_- \in \prod_{t\in\mathcal{T}} \mathcal{B}(t)$ **do**
  if $\big(\xi_+(t) - \xi_-(t)\big)^2 + \big(\zeta_+(t) + \zeta_-(t)\big)^2 \le (1+2\epsilon)^2\overline{S}_t^2$ for all $t \in \mathcal{T}$ **then**
  $y_+ \leftarrow 2m\text{DKP-Exact}\Big(\{u_{k,j}, (\widehat{s}_{k,j}(t)/L_t)_t\}_{k\in\mathcal{I}_+,j\in D_k}, \big(\xi_+(t)/L_t\big)_t, \big(\zeta_+(t)/L_t\big)_t\Big)$
  $y_- \leftarrow 2m\text{DKP-Exact}\Big(\{u_{k,j}, (-\widehat{s}_{k,j}(t)/L_t)_t\}_{k\in\mathcal{I}_-,j\in D_k}, \big(\xi_-(t)/L_t\big)_t, \big(\zeta_-(t)/L_t\big)_t\Big)$
    if $u(y_+ + y_-) > u(\widehat{x})$ **then**
      $\widehat{x} \leftarrow y_+ + y_-$
    end if
  end if
end for
return $\widehat{x}$

Theorem 4.3. `CSP-bFPTAS` is a $(1, 1+4\epsilon)$-approximation for m-CSP$[0, \pi - \varepsilon]$ and its running time is polynomial in both n and $\frac{1}{\epsilon}$.

Proof. First, the running time is proportional to the number of guesses, upper bounded by $(\frac{1}{\epsilon}n(\tan\theta + 1))^{O(1)}$.

For each guess, $2m$DKP-Exact constructs a table of size at most $(\frac{1}{\epsilon}n(\tan\theta+1))^{O(1)}$. Since we assumed $\tan\theta$ is polynomial in n, the total running time is polynomial in n and $\frac{1}{\epsilon}$.

To show the approximation ratio of 1, we note `CSP-bFPTAS` enumerates over all possible rounded projections subject to the capacity constraints in m-CSP and that $2m$DKP-Exact returns the exact optimal solution for each rounded problem. In particular, by Lemma 4.4 below one of the choices would be rounded projection for the optimum solution x^*. It remains to show that the violation of the returned solution is small in ϵ. This is given in Lemma 4.5 below, which shows that the solution $\widehat{x}$ to the rounded problem violates the capacity constraint by only a factor at most $(1+4\epsilon)$. □

For any solution $x \in [0,1]^n$, let us write for brevity

$$P_{+,t}(x) \triangleq \sum_{k\in\mathcal{I}_+}\sum_{j\in D_k:t\in T_j} x_{k,j}\overline{s}^{\mathrm{R}}_{k,j}(t), \quad P_{-,t}(x) \triangleq \sum_{k\in\mathcal{I}_-}\sum_{j\in D_k:t\in T_j} -x_{k,j}\overline{s}^{\mathrm{R}}_{k,j}(t),$$
$$\widehat{P}_{+,t}(x) \triangleq \sum_{k\in\mathcal{I}_+}\sum_{j\in D_k:t\in T_j} x_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t), \quad \widehat{P}_{-,t}(x) \triangleq \sum_{k\in\mathcal{I}_-}\sum_{j\in D_k:t\in T_j} -x_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t),$$
$$P_{I,t}(x) \triangleq \sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j} x_{k,j}\overline{s}^{\mathrm{I}}_{k,j}(t), \quad \widehat{P}_{I,t}(x) \triangleq \sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j} x_{k,j}\widehat{s}^{\mathrm{I}}_{k,j}.$$

Consider the rounded solution $\widehat{x}$ of some x such that $\widehat{x}_{k,j} \in \mathcal{R}$, and $x_{k,j} \le \widehat{x}_{k,j} \le x_{k,j} + R$. By the fact that $\ell \le \tau\lceil\frac{\ell}{\tau}\rceil \le \ell+\tau$ for any ℓ,τ such that $\tau > 0$; and that each user k has at most one $\widehat{x}_{k,j} > 0$ by Algorithm 13 (and consequently one $x_{k,j} > 0$, since $\widehat{x}_{k,j} \ge x_{k,j}$) we have

$$\begin{aligned}
\widehat{P}_{+,t}(\widehat{x}) =& \sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j} \widehat{x}_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t) = \sum_{\substack{k\in\mathcal{I}\\ j:\widehat{x}_{k,j}>0}} \widehat{x}_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t)\\
\le& \sum_{\substack{k\in\mathcal{I}\\ j:\widehat{x}_{k,j}>0}} (x_{k,j}+R)(\overline{s}^{\mathrm{R}}_{k,j}(t)+L)\\
=& P_{+,t}(t) + \sum_{\substack{k\in\mathcal{I}\\ j:\widehat{x}_{k,j}>0}} \overline{s}^{\mathrm{R}}_{k,j}(t)R + nL + nL_tR\\
\le& P_{+,t}(t) + n\overline{S}_t(\tan\theta+1)R + nL + nL_tR \le P_{+,t}(t) + nL_t,
\end{aligned}$$

where the last inequality follows by Eqn. (4.21), and the definitions of R and L_t respectively. The same bound holds for $\widehat{P}_{-,t}$ and $\widehat{P}_{I,t}$:

$$\begin{aligned}\max\{\widehat{P}_{+,t}(\widehat{x}) - nL_t, 0\} &\le P_{+,t}(x) \le \widehat{P}_{+,t}(\widehat{x}),\\ \max\{\widehat{P}_{-,t}(\widehat{x}) - nL_t, 0\} &\le P_{-,t}(x) \le \widehat{P}_{-,t}(\widehat{x}),\\ \max\{\widehat{P}_{I,t}(\widehat{x}) - nL_t, 0\} &\le P_{I,t}(x) \le \widehat{P}_{I,t}(\widehat{x}).\end{aligned} \tag{4.23}$$

Lemma 4.4. For any feasible solution x to m-CSP $[0, \pi - \varepsilon]$, and the rounded $\widehat{x} \in \mathcal{R}^n$ such that $x_{k,j} \le \widehat{x}_{k,j} \le x_{k,j} + R$, we have

$$\Big|\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} \widehat{x}_{k,j}\widehat{s}_{k,j}(t)\Big| \le (1+2\epsilon)\overline{S}_t \qquad \text{for all } t \in \mathcal{T}.$$

Proof. Using (4.23) and (4.20), for all $t \in \mathcal{T}$,

$$\begin{aligned}
&\Big(\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} \widehat{x}_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t)\Big)^2 + \Big(\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} \widehat{x}_{k,j}\widehat{s}^{\mathrm{I}}_{k,j}(t)\Big)^2\\
=&\Big(\widehat{P}_{+,t}(\widehat{x}) - \widehat{P}_{-,t}(\widehat{x})\Big)^2 + \widehat{P}^2_{I,t}(\widehat{x})\\
=&\widehat{P}^2_{+,t}(\widehat{x}) + \widehat{P}^2_{-,t}(\widehat{x}) - 2\widehat{P}_{+,t}(\widehat{x})\widehat{P}_{-,t}(\widehat{x}) + \widehat{P}^2_{I,t}(\widehat{x})\\
\le&(P_{+,t}(x) + nL_t)^2 + (P_{-,t}(x) + nL_t)^2 - 2P_{+,t}(x)P_{-,t}(x) + (P_{I,t}(x) + nL_t)^2\\
=&(P_{+,t}(x) - P_{-,t}(x))^2 + P^2_{I,t}(x) + 2nL_t(P_{+,t}(x) + P_{-,t}(x) + P_{I,t}(x)) + 3n^2L_t^2\\
=&\Big(\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} x_{k,j}\overline{s}^{\mathrm{R}}_{k,j}(t)\Big)^2 + \Big(\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} x_{k,j}\overline{s}^{\mathrm{I}}_{k,j}(t)\Big)^2\\
&\quad + 2nL_t\Big(\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} x_{k,j}|\overline{s}^{\mathrm{R}}_{k,j}(t)| + \sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} x_{k,j}\overline{s}^{\mathrm{I}}_{k,j}(t)\Big) + 3n^2L_t^2\\
\le&\overline{S}^2_t + 4nL_t(\tan\theta + 1)\overline{S}_t + 3n^2L_t^2 = \overline{S}^2_t + 4\epsilon\overline{S}^2_t + 3\epsilon^2\overline{S}^2_t/(1+\tan\theta)^2\\
\le&\overline{S}^2_t(1 + 4\epsilon + 3\epsilon^2) \le \overline{S}^2_t(1+2\epsilon)^2.
\end{aligned}$$

□

Lemma 4.5. Let $\widehat{x}$ be the solution returned by `CSP-bFPTAS`. Then

$$\Big|\sum_{k\in\mathcal{I}} \sum_{j\in D_k: t\in T_j} \widehat{x}_{k,j}\overline{s}_{k,j}(t)\Big| \le (1+4\epsilon)\overline{S}_t \qquad \text{for all } t \in \mathcal{T}.$$

Proof. As in the proof of Lemma 4.4, for all $t \in \mathcal{T}$, we have

$$
\begin{aligned}
&\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\overline{s}^{\mathrm{R}}_{k,j}(t)\Big)^2+\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\overline{s}^{\mathrm{I}}_{k,j}(t)\Big)^2\\
=&\,(P_{+,t}(\widehat{x})-P_{-,t}(\widehat{x}))^2+P^2_{I,t}(\widehat{x})\\
=&P^2_{+,t}(\widehat{x})+P^2_{-,t}(\widehat{x})-2P_{+,t}(\widehat{x})P_{-,t}(\widehat{x})+P^2_{I,t}(\widehat{x}). \qquad (4.24)
\end{aligned}
$$

If both $\widehat{P}_{+,t}(\widehat{x})$ and $\widehat{P}_{-,t}(\widehat{x})$ are less than nL_t, then the right-hand side of (4.24) can be bounded by

$$
\begin{aligned}
&\widehat{P}^2_{+,t}(\widehat{x})+\widehat{P}^2_{-,t}(\widehat{x})+\widehat{P}^2_{I,t}(\widehat{x})\\
\le&\widehat{P}^2_{+,t}(\widehat{x})+\widehat{P}^2_{-,t}(\widehat{x})-2\widehat{P}_{+,t}(\widehat{x})\widehat{P}_{-,t}(\widehat{x})+2n^2L_t^2+\widehat{P}^2_{I,t}(\widehat{x})\\
=&(\widehat{P}_{+,t}(\widehat{x})-\widehat{P}_{-,t}(\widehat{x}))^2+\widehat{P}^2_{I,t}(\widehat{x})+2n^2L_t^2. \qquad (4.25)
\end{aligned}
$$

Otherwise, we bound the right-hand side of Eqn. (4.24) by

$$
\begin{aligned}
&\widehat{P}^2_{+,t}(\widehat{x})+\widehat{P}^2_{-,t}(\widehat{x})-2(\widehat{P}_{+,t}(\widehat{x})-nL_t)(\widehat{P}_{-,t}(\widehat{x})-nL_t)+\widehat{P}^2_{I,t}(\widehat{x})\\
=&(\widehat{P}_{+,t}(\widehat{x})-\widehat{P}_{-,t}(\widehat{x}))^2+\widehat{P}^2_{I,t}(\widehat{x})+2nL_t(\widehat{P}_{+,t}(\widehat{x})+\widehat{P}_{-,t}(\widehat{x}))-2n^2L_t^2. \qquad (4.26)
\end{aligned}
$$

Since $\hat{x}=y_++y_-$ is obtained from feasible solutions y_+ and y_- to

$$2m\text{DKP-E}\textsc{xact}\Big(\{u_{k,j},(\widehat{s}_{k,j}(t)/L_t)_t\}_{k\in\mathcal{I}_+,j\in D_k},\big(\xi_+(t)/L_t\big)_t,\big(\zeta_+(t)/L_t\big)_t\Big)$$

and

$$2m\text{DKP-E}\textsc{xact}\Big(\{u_{k,j},(-\widehat{s}_{k,j}(t)/L_t)_t\}_{k\in\mathcal{I}_-,j\in D_k},\big(\xi_-(t)/L_t\big)_t,\big(\zeta_-(t)/L_t\big)_t\Big),$$

respectively, and $\xi_+,\xi_-,\zeta_+,\zeta_-$ satisfy the condition in Step 6, it follows from (4.24)-(4.26) that

$$
\begin{aligned}
&\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\overline{s}^{\mathrm{R}}_{k,j}(t)\Big)^2+\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\overline{s}^{\mathrm{I}}_{k,j}(t)\Big)^2 && (4.27)\\
\le&\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t)\Big)^2+\Big(\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}\widehat{s}^{\mathrm{I}}_{k,j}(t)\Big)^2 && (4.28)\\
&\qquad+2nL_t\sum_{k\in\mathcal{I}}\sum_{j\in D_k:t\in T_j}\widehat{x}_{k,j}|\widehat{s}^{\mathrm{R}}_{k,j}(t)|+2n^2L_t^2\\
=&(\xi_+(t)-\xi_-(t)^2+(\zeta_+(t)+\zeta_-(t)^2+2nL_t(\xi_+(t)+\xi_-(t)+2n^2L_t^2 && (4.29)\\
\le&\Big((1+2\epsilon)^2\overline{S}_t^2+4n\frac{\epsilon\overline{S}_t}{n(\tan\theta+1)}(1+\tan\theta)\overline{S}_t+2n^2\frac{\epsilon^2\overline{S}_t^2}{n^2(\tan\theta+1)^2}\Big) && (4.30)\\
\le&\Big((1+2\epsilon)^2+4\epsilon+2\epsilon^2\Big)\overline{S}_t^2\le(1+4\epsilon)^2\overline{S}_t^2. && (4.31)
\end{aligned}
$$

□

4.4 Notes

Scheduling problems are a well-studied topic in computer science. However, most extant literature considers scheduling problems formulated by linear constraints. Khonji *et al.* (2016) and Khonji *et al.* (2018e) consider the Complex-demand Scheduling Problem (CSP), involving AC power demands subject to generation apparent power constraints. This monograph covers the case of constant number of time slots. For the case of a polynomial number of time slots, CSP is a generalization of the unsplittable flow problem on paths to accommodate complex-valued demands. Also, Khonji *et al.* (2016) and Khonji *et al.* (2018e) extend the greedy algorithm in Karapetyan *et al.* (2018) for the single time slot case to handle multiple demands per user keeping the same approximation ratio and running time. Recently, the joint problem of temporal scheduling and CSP has been studied in Khonji *et al.* (2018c), Khonji *et al.* (2018b), and Khonji *et al.* (2018d).

Algorithm 13 $2m\text{DKP-}\textsc{Exact}[\big(u_{k,j}, \{\widehat{s}_{k,j}(t)\}_{t\in T_j}\big)_{k\in\mathcal{W},j\in D_k}, (c^1_t)_{t\in\mathcal{T}}, (c^2_t)_{t\in\mathcal{T}}]$

Require: Utilities, and rounded demands of a restricted set of users $\mathcal{W} \subseteq \mathcal{I}$, $\{u_{k,j}, \{\overline{s}_{k,j}(t)\}_{t\in T_j}\}_{k\in\mathcal{I},j\in D_k}$; integer capacity vectors $(c^1_t)_{t\in\mathcal{T}}, (c^2_t)_{t\in\mathcal{T}}$

Ensure: A utility-maximizing optimal solution $y \in [0,1]^n$ subject to the capacity constraints defined by c^1_t, c^2_t

1: Create a table of size $|\mathcal{W}| \cdot \prod_t (c^1_t + 1)\cdot(c^2_t+1)$, with each entry $U(k, c^1, c^2)$ according to:

$$U(1,c^1,c^2) \triangleq \max \begin{Bmatrix} \max\limits_{(1,j)\in(\mathcal{I}\cap\mathcal{W})} \{u_{k,j} \mid \widehat{s}^{\mathrm{R}}_{k,j}(t) = c^1_t, \ \ \widehat{s}^{\mathrm{I}}_{k,j}(t) = c^2_t, \ \forall t\}, \\ \max\limits_{\substack{(1,j)\in(\mathcal{F}\cap\mathcal{W})\\ \widehat{x}_{k,j}\in\mathcal{R}}} \{\widehat{x}_{k,j}u_{k,j} \mid \widehat{x}_{k,j}\widehat{s}^{\mathrm{R}}_{k,j}(t) = c^1_t, \\ \qquad\qquad \widehat{x}_{k,j}\widehat{s}^{\mathrm{I}}_{k,j}(t) = c^2_t, \ \forall t\}, \\ -\infty \end{Bmatrix}$$

$$U(k,c^1,c^2) \triangleq \max \begin{Bmatrix} \max\limits_{(k,j)\in(\mathcal{I}\cap\mathcal{W})} \Big\{u_{k,j} + U\big(k-1, (c^1_t - \widehat{s}_{k,j}(t))_t, \\ \qquad\qquad (c^2_t - \widehat{s}_{k,j}(t))_t\big)\Big\}, \\ \max\limits_{\substack{(k,j)\in(\mathcal{F}\cap\mathcal{W})\\ \widehat{x}_{k,j}\in\mathcal{R}}} \Big\{\widehat{x}_{k,j}u_{k,j} + U\big(k-1, (c^1_t - \widehat{x}_{k,j}\widehat{s}_{k,j}(t))_t, \\ \qquad\qquad (c^2_t - \widehat{x}_{k,j}\widehat{s}_{k,j}(t))_t\big)\Big\}, \\ U(k-1, c^1, c^2) \end{Bmatrix}$$

$$U(k,c^1,c^2) \triangleq -\infty, \ \forall c^1, c^2 \notin \mathbb{R}^m_+$$

2: Create a table of size $|\mathcal{W}| \cdot \prod_t (c^1_t + 1)\cdot(c^2_t+1)$, with each entry $y(k,c^1,c^2) \in [0,1]^n$ according to:

$$y(k,c^1,c^2) \triangleq \mathbf{0} \text{ if } U(k,c^1,c^2) = -\infty$$
$$y(1,c^1,c^2) \triangleq \{(0,...,\widehat{x}_{kj},...,0) \mid \widehat{x}_{k,j}u_{k,j} = U(1,c^1,c^2)\}$$
$$y(k,c^1,c^2) \triangleq \{y(k-1,c^1,c^2) + (0,...,\widehat{x}_{kj},...,0) \mid \\ \widehat{x}_{k,j}u_{k,j} = U(1,c^1,c^2) - u\big(y(k-1,c^1,c^2)\big)\}$$

3: **return** $y(|\mathcal{W}|, c^1, c^2)$.

5

Hardness Results

In this section, we provide inapproximability results for CKP and OPF, which complement with our approximation solutions. Based on the hardness results, our approximation algorithms (PTAS and resource-augmented FPTAS) are the best achievable efficient algorithms in theory in the respective settings of CKP and OPF.

In the following, we show hardness results for special cases of OPF, which are sufficient for the hardness of general OPF. We denote by $\mathrm{OPF}_{\mathsf{V}}$ when OPF considers voltage Cons. (1.27) and thermal capacity Cons. (1.29) without Cons. (1.28), whereas by $\mathrm{OPF}_{\mathsf{S}}$ when OPF considers capacity Cons. (1.28)-(1.29) without Cons. (1.27).

In the hardness proofs, we often rely on the hardness results of well-known NP-Hard problems called the EQUIPARTITION problem and Subset Sum (SUBSUM) problem.

Definition 5.1. (EQUIPARTITION): Given a set of positive integers $\{w_k\}_{k\in\mathcal{I}}$, with $|\mathcal{I}| = n$ where n is even, we determine if there is a subset of items $X \subseteq \mathcal{I}$ such that

$$|X| = \frac{n}{2} \text{ and } \sum_{k\in X} w_k = \sum_{k\notin X} w_k \tag{5.1}$$

Definition 5.2 (SubSum). Given a set of positive integers $A \triangleq \{a_1, ..., a_m\}$ and a positive integer B, decide if there exists a subset of A that sums-up to exactly B.

Note that B is generally not polynomial in m. Otherwise, SubSum can be solved easily in polynomial time by dynamic programming.

5.1 Absence of FPTAS

First, we show that CKP$[0, \frac{\pi}{2}]$ does not admit an FPTAS (a PTAS but requires the running time to be polynomial in both input size and $1/\epsilon$), unless P = NP.

We remark that it is known that there is no FPTAS for 2-KP (see Kellerer *et al.* (2010)), which does not have direct implications for CKP. However, our proof is an extension of the basic idea in the proof for 2-KP. As in the reduction for 2-KP, we reduce the Equipartition problem to CKP$[0, \frac{\pi}{2}]$.

Theorem 5.1. There is no FPTAS for CKP$[0, \frac{\pi}{2}]$, unless P = NP.

Remark 5.1. By Theorems 5.1, a PTAS is among the best achievable efficient algorithm that can be attained for CKP and OPF with discrete demands, because FPTAS is not possible.

Proof. We define a decision version of CKP$[0, \frac{\pi}{2}]$ with a cardinality objective: given $\{w_k\}_{k\in\mathcal{I}}$, a capacity bound $\overline{S}$ and a cardinality bound M, we determine if there is a subset of items X such that

$$|X| \geq M, \text{ and } \Big|\sum_{k\in X} \overline{s}_k\Big| \leq \overline{S} \tag{5.2}$$

Now we map every instance of Equipartition to an instance of the CKP$[0, \frac{\pi}{2}]$ decision problem that always yields the same answer.

Given $\{w_k\}_{k\in\mathcal{I}}$ from Equipartition, define

$$M = n/2, \quad \overline{s}_k^{\mathrm{R}} = w_k, \quad \overline{s}_k^{\mathrm{I}} = \beta(w_{\max} - w_k), \tag{5.3}$$

$$\overline{S} = \sqrt{\Big(\frac{W}{2}\Big)^2 + \beta^2\Big(\frac{n w_{\max}}{2} - \frac{W}{2}\Big)^2} \tag{5.4}$$

where $W \triangleq \sum_{k=1}^{n} w_k$, $w_{\max} \triangleq \max\{w_k : k \in \mathcal{I}\}$. Note that in our reduction, $\overline{s}_k^{\mathrm{I}} \geq 0$.

As shown in Fig. 5.1, the feasible region $\mathcal{D}$ for CKP$[0, \frac{\pi}{2}]$ is the $\frac{1}{4}$ disk of radius $\overline{S}$ in the first quadrant. Since for any subset $X \subseteq \mathcal{I}$, $\sum_{k\in X} \overline{s}_k^{\mathrm{I}} = \beta(|X| \cdot w_{\max} - \sum_{k\in X} \overline{s}_k^{\mathrm{R}})$, the cardinality constraint $|X| \geq \frac{n}{2}$ imposes all solutions to have its sum vector in the half-plane H : $\nu^{\mathrm{I}} \geq \beta(\frac{nw_{\max}}{2} - \nu^{\mathrm{R}})$. The dividing line of H goes through point P : $\left(\frac{W}{2}, \beta(\frac{nw_{\max}}{2} - \frac{W}{2})\right)$. Our main idea is to set $\beta > 0$ such that the dividing line of H coincides with the tangent line at P. Thus we make the intersection of H and $\mathcal{D}$ exactly P, which implies $|X| = \frac{n}{2}$ and $\sum_{k\in X} w_j = \frac{W}{2}$ for any solution X to our reduced CKP$[0, \frac{\pi}{2}]$ decision problem instance.

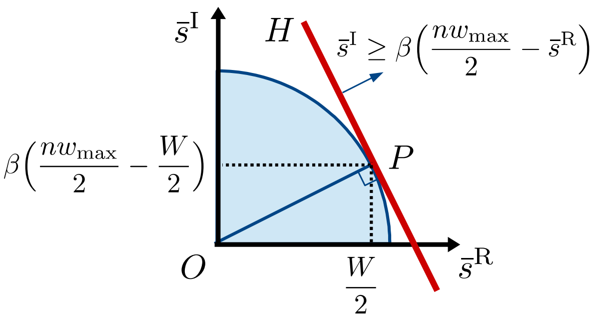


Figure 5.1: Reduction of inapproximability.

On the other hand, it is clear that each subset X satisfying conditions of Equipartition also satisfies conditions of the reduced CKP$[0, \frac{\pi}{2}]$ decision problem. Therefore, the solution of the reduced CKP$[0, \frac{\pi}{2}]$ decision problem is equivalent to the solution of Equipartition.

To determine a proper β, since the dividing line of half-plane H goes through P, it coincides with the tangent line at P if and only if they have the same slope, i.e.,

$$-\frac{\frac{W}{2}}{\beta(\frac{nw_{\max}}{2} - \frac{W}{2})} = -\beta. \tag{5.5}$$

Solving the above equation, we obtain

$$\beta = \sqrt{\frac{W}{nw_{\max} - W}}, \tag{5.6}$$

which is > 0 unless all weights are equal. In this case, we set $\beta = 0$, and it is trivially a "yes" instance for both EQUIPARTITION and our CKP$[0, \frac{\pi}{2}]$ decision problem.

So far we have shown the NP-hardness of the CKP$[0, \frac{\pi}{2}]$ decision problem. So its maximization version, where $|X| \geq M$ is replaced by $\max |X|$, is NP-hard. We use the standard technique to prove the inapproximability of the maximization version by FPTAS. Suppose that there exists an FPTAS for any $\epsilon > 0$ in time polynomial in n and $1/\epsilon$. Then we choose $\epsilon = \frac{1}{n+1}$. Let the optimal solution be $F^* > 0$ and that of the approximation solution produced by FPTAS be F^A. We obtain

$$F^A \geq (1 - \epsilon)F^* > F^* - \frac{F^*}{n} \geq F^* - 1 \tag{5.7}$$

because $F^* \leq n$. Moreover, since F^* is an integer, this implies that the FPTAS can solve the problem exactly in polynomial time, contradicting the NP-hardness of the problem.

Finally, since the maximization version of CKP$[0, \frac{\pi}{2}]$ decision problem is a special case of the original CKP$[0, \frac{\pi}{2}]$ with all $u_k = 1$, there is no FPTAS to CKP$[0, \frac{\pi}{2}]$. □

5.2 Hardness of CKP

In this section, we show the hardness of CKP. The hardness depends on the maximum angle ϕ at which the demands make with the positive real axis. When $\phi \in [\frac{\pi}{2} + \delta, \pi]$, we show that the problem is inapproximable within any polynomial factor if we do not allow a violation of Cons. (2.1). Moreover, when ϕ approaches π, there is no (α, β)-approximation, for any α and β with polynomial bit length. Our hardness results indicate that the approximability of the problem CKP differs depending on maximum argument of any demand ϕ. This insight suggests to study different techniques in the later sections to achieve the best approximation result possible for each case.

Theorem 5.2. Unless P=NP, for any $\delta > 0$ and $\delta' > 0$

(i) there is no $(\alpha, 1)$-approximation for CKP$[\frac{\pi}{2}+\delta, \pi]$ where α, δ have polynomial length.

(ii) there is no (α, β)-approximation for CKP$[\pi - \delta', \pi]$, where α and β have polynomial length, and δ' is exponentially small in n.

Remark 5.2. In fact, these hardness results hold even if we assume that all demands are on the real line, except one demand $\overline{s}_{m+1}$ such that $\arg(\overline{s}_{m+1}) = \frac{\pi}{2} + \theta$, for some $\theta \in [\delta, \frac{\pi}{2}]$ (see Fig. 5.2). Note that the trivial approximation of picking the user with the highest feasible value does not obtain $\frac{1}{n}$ approximation, because we allow demands to have both positive and negative real parts, which can cancel each other.

Remark 5.3. For CKP$[\pi - \delta', \pi]$, Theorem 5.2 shows that efficient approximation algorithm is not possible without violating the constraint. Hence, a resource-augmented FPTAS is the best achievable efficient algorithm.

Remark 5.4. Theorem 5.2 also shows that assumption A4 is necessary for PTAS of OPF with discrete demands.

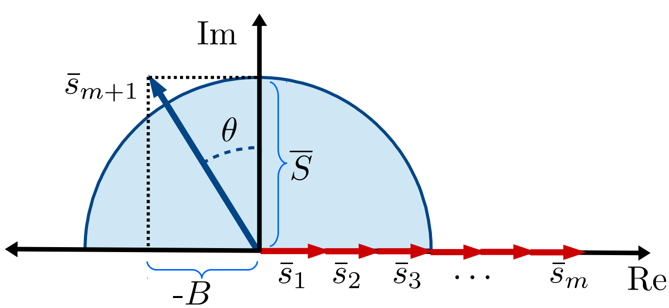


Figure 5.2: The set of demands $\{\overline{s}_k\}$ for the proof of Theorem 5.2.

Proof. We present a reduction from the (weakly) NP-hard Subset Sum (SubSum) problem: given an instance I, a set of positive integers $A \triangleq \{a_1, \ldots, a_m\}$ and a positive integer B, does there exist a subset of A

that sums-up to exactly B? We assume that B is not polynomial in m, otherwise the problem can be easily solved in polynomial time by dynamic programming.

We construct an instance I' of CKP$[\frac{\pi}{2}+\theta, \frac{\pi}{2}+\theta]$ for each instance I of SubSum such that if SubSum(I) is a "yes" instance then the optimum value of CKP$[\frac{\pi}{2}+\theta, \frac{\pi}{2}+\theta]$, denoted by Opt, is at least 1; and if SubSum(I) is a "no" instance, then Opt $< \alpha$ even when Cons. (2.1) can be violated by β.

We define $n \triangleq m+1$ demands: for each $a_k, k = 1, ..., m$, define a demand $\overline{s}_k \triangleq a_k$, and an additional demand

$$\overline{s}_{m+1} \triangleq -B + \mathbf{i}B\cot\theta.$$

For all $k = 1, ..., m$, let valuation $v_k \triangleq \frac{\alpha}{m+1}$, and $v_{m+1} \triangleq 1$. We let

$$\overline{S} \triangleq B\cot\theta.$$

We prove the first direction, assuming SubSum(I) is feasible. Namely, $\sum_{k=1}^m a_k\hat{x}_k = B$, where $\hat{x} \in \{0,1\}^m$ is a solution vector of SubSum. Construct a solution $x \in \{0,1\}^{m+1}$ of CKP such that

$$x_k = \begin{cases} \hat{x}_k & \text{if } k = 1, ..., m \\ 1 & \text{if } k = m+1. \end{cases}$$

In fact, this is a feasible solution that satisfies Cons. (2.1): using $\sum_{k=1}^m a_k\hat{x}_k - B = 0$, we get

$$\begin{aligned} &\left(\sum_{k=1}^m \overline{s}_k^{\mathrm{R}} x_k + \overline{s}_{m+1}^{\mathrm{R}}\right)^2 + \left(\sum_{k=1}^m \overline{s}_k^{\mathrm{I}} x_k + \overline{s}_{m+1}^{\mathrm{I}}\right)^2 \\ =&\left(\sum_{k=1}^m \overline{s}_k^{\mathrm{R}} x_k - B\right)^2 + B^2\cot^2\theta \\ =&B^2\cot^2\theta = \overline{S}^2. \end{aligned}$$

Since $v_{m+1} = 1$, the total value of such solution $v(x) \geq 1$, which implies that Opt is at least 1.

Conversely, assume that Opt $\geq \alpha$. Let $x^\star \in \{0,1\}^{m+1}$ be an optimal solution that may violate Cons. (2.1) by β. Since user $m+1$ has valuation $v_{m+1} = 1$, while the rest of users valuations total to less than

α: $\sum_{k=1}^{m} v_k < \alpha$, user $m+1$ must be included in the optimum. Therefore, substituting in Cons. (2.1) with violation at most β,

$$\left(\sum_{k=1}^{m} \overline{s}_k^{\mathrm{R}} x_k^\star - B\right)^2 + B^2 \cot^2\theta \le \beta^2 \overline{S}^2$$

gives

$$\begin{aligned}\left(\sum_{k=1}^{m} a_k x_k^\star - B\right)^2 &\le \beta^2 \overline{S}^2 - B^2 \cot^2\theta \\ &= B^2 \cot^2\theta(\beta^2 - 1).\end{aligned} \tag{5.8}$$

By the integrality of the a_i's,

$$\sum_{k=1}^{m} a_k x_k^\star = B \iff \left|\sum_{k=1}^{m} a_k x_k^\star - B\right| < 1 \tag{5.9}$$

In other words, SubSum is feasible if and only if the absolute difference $|\sum_{k=1}^{m} a_k x_k^\star - B| < 1$. This implies, SubSum$(I)$ is feasible when the right-hand side of Eqn. (5.8) is strictly less than 1. When $\beta = 1$, right-hand side of Eqn. (5.8) is zero, and we complete the second direction and hence, the proof of part (i) of the theorem.

For large enough θ, the right-hand side of Eqn. (5.8) is strictly less than 1:

$$B^2 \cot^2\theta(\beta^2 - 1) < 1$$

This implies, $\theta > \tan^{-1}\sqrt{B^2(\beta^2 - 1)}$. By Eqn. (5.9), SubSum is feasible which completes the second direction and establishes part (ii) of the theorem. □

5.3 Hardness of OPF with Voltage Constraint

Theorem 5.3. Unless P=NP, there is no (α, β)-approximation for OPF$_{\mathsf{V}}$ (even when $|\mathcal{E}| = 1$) by a polynomial-time algorithm in n, for any α and β have polynomial length in n.

Remark 5.5. Theorem 5.3 shows that assumption A3 is necessary for PTAS of OPF with discrete demands.

Proof. The basic idea is that we show a reduction from SubSum to OPF_{V}. Assume that there is an (α, β)-approximation for OPF_{V}. We construct an instance I' of OPF_{V} for each instance I of SubSum, such that SubSum(I) is a "yes" instance if and only if the (α, β)-approximation of $\text{OPF}_{\mathsf{V}}(I')$ gives a total utility at least α. Since SubSum is NP-hard, there exists no (α, β)-approximation for OPF_{V} in polynomial time. Otherwise, SubSum can be solved in polynomial time.

Given a SubSum instance $I = (A, B)$, where $A = \{a_1, ..., a_m\}$, we define OPF_{V} instance I' as follows.

- Consider a network with a single edge $e = (0, 1)$. Let $z_e \triangleq 1 + \mathbf{i}$.
- Fix some $\delta > 0$, set v_0 in instance I' by

  $$v_0 \triangleq \frac{1}{\frac{\delta}{2} - \epsilon} \Big| \sum_{k=1}^{m+1} \overline{s}_k \Big|^2, \quad \text{for arbitrarily small } \epsilon > 0.$$

  Set $\overline{v} = v_0 + \delta$ and $\underline{v} = v_0 - \delta$.
- Let $\mathcal{N} = \mathcal{I} = \{1, ..., m + 1\}$ be the set of customers attached to node 1 (see Fig. 5.3). Define

  $$\overline{\Lambda} \triangleq \max\{v_0 - \tfrac{1}{\beta}\underline{v}, \beta\overline{v} - v_0\}.$$

  For each $k \in \{1, ..., m + 1\}$, define the customers' demands and utilities as follows.

  $$\overline{s}_k \triangleq \overline{\Lambda} a_k, \quad u_k \triangleq \tfrac{\alpha}{m+1} \quad \overline{s}_{m+1} \triangleq -\mathbf{i}\overline{\Lambda} B, \quad u_{m+1} \triangleq 1$$

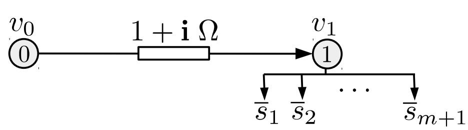


Figure 5.3: A gadget for reduction from SubSum to OPF_{V}.

First, we prove that if SubSum(I) is a "yes" instance, then the (α, β)-approximation of $\text{OPF}_{\mathsf{V}}(I')$ gives a total utility at least α. If

SUBSUM(I) is a "yes" instance, then $\sum_{k=1}^{m} a_k \hat{x}_k = B$, where $\hat{x} \in \{0,1\}^m$ is a solution of SUBSUM. We construct a solution $x \in \{0,1\}^{m+1}$ for $\text{OPF}_{\mathsf{V}}(I')$ by

$$x_k = \begin{cases} \hat{x}_k & \text{if } k = 1, ..., m \\ 1 & \text{if } k = m+1 \end{cases}$$

We formulate Cons. (1.23) as $S_e = \sum_{k \in \mathcal{N}} \overline{s}_k + z_e \ell_e$ and substitute it in Cons. (1.27) to obtain

$$\tfrac{1}{2}(v_0 - \overline{v}) \le \mathrm{Re}(z_e^* S_e) - \tfrac{1}{2}|z_e|^2 \ell_e \le \tfrac{1}{2}(v_0 - \underline{v})$$

Note that

$$\begin{aligned} \tfrac{1}{2}(v_0 - v_1) &= \mathrm{Re}(z_e^* S_e) - \tfrac{1}{2}|z_e|^2 \ell_e \\ &= \mathrm{Re}(\sum_{k \in \mathcal{N}} z_e^* \overline{s}_k x_k + |z_e|^2 \ell_e) - \tfrac{1}{2}|z_e|^2 \ell_e \\ &= \sum_{k=1}^{m+1} (z_e^{\mathrm{R}} s_k^{\mathrm{R}} + z_e^{\mathrm{I}} s_k^{\mathrm{I}}) x_k + \tfrac{1}{2}|z_e|^2 \ell_e \\ &= \overline{\Lambda}\Big(\sum_{k=1}^{m} a_k x_k - B x_{m+1}\Big) + \ell_e = \ell_e, \end{aligned}$$

where $\sum_{k=1}^{m} a_k \hat{x}_k - B = 0$ and $|z_e|^2 = 2$.

By Cons. (1.23) and the definition of v_0, we obtain

$$\ell_e = \frac{|S_e|^2}{v_0} = \frac{(\frac{\delta}{2} - \epsilon) \cdot \Big|\sum_{k=1}^{m+1} \overline{s}_k x_k\Big|^2}{\Big|\sum_{k=1}^{m+1} \overline{s}_k\Big|^2} \le \tfrac{\delta}{2} - \epsilon \tag{5.10}$$

Hence,

$$0 \le \frac{v_0 - v_1}{2} = \ell_e \le \tfrac{\delta}{2} - \epsilon < \frac{v_0 - \underline{v}}{2}$$

Therefore, $\underline{v} \le v_1 \le v_0 = \overline{v}$ and Cons. (1.27) is satisfied. Since $u_{m+1} = 1$, we have $u(x) \ge 1$, and OPT is also at least 1. By the feasibility of this solution, the (α, β)-approximation of $\text{OPF}_{\mathsf{V}}(I')$ gives a total utility at least α.

Conversely, assume that the (α, β)-approximation algorithm gives a solution $x \in \{0,1\}^{m+1}$ of total utility at least α. Customer $m+1$ must be satisfied in this solution. Then, Cons. (1.27) with violation β implies

$$\frac{v_0-\beta\overline{v}}{2} \le \sum_{k=1}^{m} \overline{\Lambda}(a_k x_k - B) + \ell_e \le \frac{v_0 - \frac{1}{\beta}\underline{v}}{2}$$

$$\Rightarrow -\frac{(\beta\overline{v}-v_0)}{2\overline{\Lambda}} - \frac{\ell_e}{\overline{\Lambda}} \le \sum_{k=1}^{m} a_k x_k - B \le \frac{v_0 - \frac{1}{\beta}\underline{v}}{2\overline{\Lambda}} - \frac{\ell_e}{\overline{\Lambda}}$$

The right-hand side can be bounded by

$$\frac{v_0 - \frac{1}{\beta}\underline{v}}{2\overline{\Lambda}} - \frac{\ell_e}{\overline{\Lambda}} \le \frac{1}{2} < 1$$

Using Eqn. (5.10), the left-hand side can be bounded by

$$-\frac{(\beta\overline{v}-v_0)}{2\overline{\Lambda}} - \frac{\ell_e}{\overline{\Lambda}} \ge -\frac{1}{2} - \frac{\frac{\delta}{2}-\epsilon}{\overline{\Lambda}} > -1$$

Since $|\sum_{k=1}^{m} a_k x_k - B| < 1$, and a_k, B are integers, this implies $\sum_{k=1}^{m} a_k x_k - B = 0$. Hence, SubSum($I$) is a "yes" instance. □

5.4 Hardness of OPF with Capacity Constraint

Theorem 5.4. Unless P=NP, there exists no (α, β)-approximation for OPF_C in general networks, for any α and β have polynomial length in n, even in DC electric networks (i.e., $\text{Im}(z_{i,j}) = 0$ for all $(i,j) \in \mathcal{E}$ and $\text{Im}(\overline{s}_k) = 0$ for all $k \in \mathcal{N}$).

Remark 5.6. Theorem 5.4 shows that the assumption of radial network is necessary for PTAS of OPF with discrete demands.

Proof. The basic idea is similar to that of Theorem 5.3. We consider a DC electric network that contains a cycle.

Given a SubSum instance $I = (A, B)$, where $A = \{a_1, ..., a_m\}$, we define a OPF_C instance I' as follows. Define the customers' demands and utilities by

$$\overline{s}_k \triangleq a_k, \quad u_k \triangleq \tfrac{\alpha}{m+1} \quad \overline{s}_{m+1} \triangleq B, \quad u_{m+1} \triangleq 1$$

Consider the network in Fig. 5.4 for OPF_C such that all power demands $\{\overline{s}_k\}_{k=1,...,m}$ are attached to node a, $\overline{s}_{m+1}$ is attached to b, and $z_{0,a} = z_{0,b} = z_{a,b} = 1$.

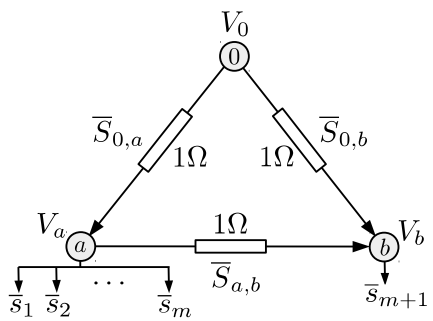


Figure 5.4: A gadget for reduction from SubSum to OPF$_\mathsf{C}$.

Denote the transmitted power, current, and resistance on edge (i,j) by $S_{i,j}$, $I_{i,j}$, $z_{i,j}$ respectively. Let $S_a \triangleq \sum_{k=1}^{m} \overline{s}_k x_k$ and $S_b \triangleq \overline{s}_{m+1} x_{m+1}$ be the total demand on node a and b respectively. Without loss of generality, assume $V_a \geq V_b$. By the power balance equations, we obtain

$$\begin{aligned} S_a &= S_{0,a} - I_{0,a}^2 z_{0,a} - S_{a,b}, \\ S_b &= S_{0,b} - I_{0,b}^2 z_{0,b} + S_{a,b} - I_{a,b}^2 z_{z,b}. \end{aligned}$$

Using Ohm's law $I_{i,j} = \frac{V_i - V_j}{z_{i,j}}$ and $S_{i,j} = V_i I_{i,j}$, we obtain

$$\begin{aligned} S_a &= \frac{V_0(V_0 - V_a)}{z_{0,a}} - \frac{(V_0 - V_a)^2}{z_{0,a}} - \frac{V_a(V_a - V_b)}{z_{a,b}}, \\ S_b &= \frac{V_0(V_0 - V_b)}{z_{0,b}} - \frac{(V_0 - V_b)^2}{z_{0,b}} + \frac{V_a(V_a - V_b)}{z_{a,b}} - \frac{(V_a - V_b)^2}{z_{a,b}} \end{aligned}$$

Note that the above equations can also obtained when $V_b \geq V_a$.

Since $z_{0,a} = z_{0,b} = z_{a,b} = 1$, we obtain

$$\begin{aligned} 2V_a^2 - (V_0 + V_b)V_a + S_a = 0 \\ 2V_b^2 - (V_0 + V_a)V_b + S_b = 0 \end{aligned}$$

It follows that

$$\begin{aligned} S_a - S_b &= 2V_b^2 - 2V_a^2 + V_0(V_a - V_b) && (5.11) \\ &= (V_b - V_a)(2V_b + 2V_a - V_0) && (5.12) \end{aligned}$$

Let $x = (x_1, \dots, x_{m+1})$ be a solution of the (α, β)-approximation to OPF_{C}, where x_k indicates if power demand $\overline{s}_k$ is satisfied for $k \in \{1, ..., m\}$, and x_{m+1} indicates if power demand $\overline{s}_{m+1}$ is satisfied. If the total utility of x is at least α, then we necessarily have $x_{m+1} = 1$. Considering the capacity constraints, we obtain

$$\begin{aligned}|V_a(V_a - V_b)| &\leq \beta \overline{S}_{a,b},\\ |V_0(V_0 - V_a)| &\leq \beta \overline{S}_{0,a},\\ |V_0(V_0 - V_b)| &\leq \beta \overline{S}_{0,b}\end{aligned}$$

Note that $V_0 > V_a$, because V_0 is attached to generation. Then, we obtain

$$\frac{V_0^2 - \beta \overline{S}_{0,a}}{V_0} \leq V_a \tag{5.13}$$

By substituting Eqn. (5.12), Eqn. (5.13) and considering $V_0 > V_a$ and $V_0 > V_b$, we obtain

$$\begin{aligned} & |V_a(V_a - V_b)| \leq \beta \overline{S}_{a,b} \\ \Rightarrow \qquad & |S_a - S_b| \leq \beta \overline{S}_{a,b} \left| \frac{2V_b + 2V_a - V_0}{V_a} \right| \\ & \leq \beta \overline{S}_{a,b} \frac{3V_0}{V_a} \leq \frac{3\beta \overline{S}_{a,b} V_0^2}{V_0^2 - \beta \overline{S}_{0,a}} \end{aligned}$$

Next, we set $\overline{S}_{a,b} < \frac{V_0^2 - \beta \overline{S}_{0,a}}{3\beta V_0^2}$, such that

$$|V_a(V_a - V_b)| \leq \beta \overline{S}_{a,b} \;\Rightarrow\; |S_a - S_b| = \left| B - \sum_{k=1}^{m} a_k x_k \right| < 1$$

Thus, $\textsc{SubSum}(I)$ is a “yes" instance.

Conversely, a feasible solution $x \in \{0, 1\}^{m+1}$ satisfying $\sum_{k=1}^{n} a_k x_k - B x_{m+1} = 0$, with $x_{m+1} = 1$, we can see that $S_a = S_b = B$. Next, we set $V_a = V_b = V'$ for some positive value $V' < V_0$ and $\overline{S}_{0,a} = \overline{S}_{0,b} = B + (V_0 - V')^2$. This is a feasible solution (with $\beta = 1$) to $\text{OPF}_{\mathsf{C}}(I')$, which has utility at least 1. Thus the (α, β)-approximation returns a solution of utility at least α. $\square$

5.5 Hardness of Simplified OPF

Recall that sOPF is simplified OPF under the DistFlow model. Note that since sOPF has a different model from OPF, its hardness results cannot be directly derived from those of OPF.

Definition 5.3. For $\alpha \in (0, 1]$ and $\beta \geq 1$, we define a bi-criteria (α, β)-approximation to sOPF as a solution $\hat{x} = \left((\hat{x}_k)_{k\in\mathcal{I}}, (\hat{x}_k)_{k\in\mathcal{F}}\right) \in \{0,1\}^{|\mathcal{I}|} \times [0,1]^{|\mathcal{F}|}$ satisfying

$$\left|\sum_{k:e\in P_k} \overline{s}_k x_k\right| \leq \beta \hat{C}_e, \qquad \forall e \in \mathcal{E} \tag{5.14}$$

$$\frac{1}{\beta}\underline{V_e} \leq \sum_{k\in\mathcal{N}} \Big(\sum_{e'\in P_k\cap P_e} z^{\mathrm{R}}_{e'}\overline{s}^{\mathrm{R}}_k + z^{\mathrm{I}}_{e'}\overline{s}^{\mathrm{I}}_k\Big) x_k \leq \beta\overline{V_e},\ \forall e \in \mathcal{E} \tag{5.15}$$

such that $u(\hat{x}) \geq \alpha\textsc{Opt}$.

Theorem 5.5. Unless P=NP, there is no (α, β)-approximation for $\mathrm{sOPF}_{\mathsf{V}}$ (even when $|\mathcal{E}| = 1$) by a polynomial-time algorithm in n, for any α and β have polynomial length in n.

Proof. We present a reduction from SubSum to $\mathrm{sOPF}_{\mathsf{V}}$. Assume that there is an (α, β)-approximation for $\mathrm{sOPF}_{\mathsf{V}}$. We construct an instance I' of $\mathrm{sOPF}_{\mathsf{V}}$ for each instance I of SubSum, such that $\textsc{SubSum}(I)$ is a "yes" instance if and only if the (α, β)-approximation of $\mathrm{sOPF}_{\mathsf{V}}(I')$ gives a total utility at least α.

We define the $\mathrm{sOPF}_{\mathsf{V}}$ instance I' as follows. Consider a graph with a single edge e. Let $z_e \triangleq 1 + \mathbf{i}$, $\underline{V} = \underline{V_e}$ and $\overline{V} = \overline{V_e}$. Define $\Lambda \triangleq \max\{-\frac{1}{\beta}\underline{V}, \beta\overline{V}\}$. Let $\mathcal{N} = \mathcal{I} = \{1, ..., m+1\}$ be the set of customers (i.e., all having inelastic demands). For each $k \in \{1, ..., m+1\}$, define the customers' demands and utilities by:

$$\overline{s}_k \triangleq 2\Lambda a_k, \quad u_k \triangleq \tfrac{\alpha}{m+1} \quad \overline{s}_{m+1} \triangleq -\mathbf{i}2\Lambda B, \quad u_{m+1} \triangleq 1$$

First, we prove that if $\textsc{SubSum}(I)$ is a "yes" instance, then the (α, β)-approximation of $\mathrm{sOPF}_{\mathsf{V}}(I')$ gives a total utility at least α. If $\textsc{SubSum}(I)$ is a "yes" instance, then $\sum_{k=1}^{m} a_k \hat{x}_k = B$, where $\hat{x} \in \{0,1\}^m$ is a solution vector of SubSum. Construct a solution $x \in \{0,1\}^{m+1}$ of

sOPF_V such that

$$x_k = \begin{cases} \hat{x}_k & \text{if } k = 1, ..., m \\ 1 & \text{if } k = m+1 \end{cases}$$

By $\sum_{k=1}^{m} a_k \hat{x}_k - B = 0$, we obtain

$$\begin{aligned} \sum_{k=1}^{m+1} (z_e^{\mathrm{R}} s_k^{\mathrm{R}} + z_e^{\mathrm{I}} s_k^{\mathrm{I}}) x_k &= \sum_{k=1}^{m} 2\Lambda a_k x_k - 2\Lambda B x_{m+1} \\ &= \sum_{k=1}^{m} 2\Lambda (a_k x_k - B x_{m+1}) = 0 \end{aligned}$$

Therefore, $(x_k)_{k \in \mathcal{N}}$ is a feasible solution of sOPF_V and satisfies Eqn. (3.72). Since $u_{m+1} = 1$, $u(x)$ is at least one which implies that Opt is also at least 1, and hence, by the feasibility of this solution, any (α, β)-approximation gives a total utility at least α.

Conversely, assume the (α, β)-approximation gives a solution $x \in \{0,1\}^{m+1}$ of total utility at least α. Since customer $m+1$ has valuation $v_{m+1} = 1$, while the rest of customers valuations total to less than α (i.e., $\sum_{k=1}^{m} u_k < \alpha$), customer $m+1$ must be satisfied in this solution. Therefore, we obtain

$$\frac{1}{\beta}\underline{V} \le \sum_{k=1}^{m} 2\Lambda(a_k x_k - B) \le \beta \overline{V} \tag{5.16}$$

$$\Rightarrow \quad \tfrac{\underline{V}}{2\beta\Lambda} \le \sum_{k=1}^{m} a_k x_k - B \le \tfrac{\beta \overline{V}}{2\Lambda} \tag{5.17}$$

Since $-\frac{\underline{V}}{2\beta\Lambda}, \frac{\beta\overline{V}}{2\Lambda} \le \frac{1}{2} < 1$, and a_k, B are integers, this implies $\sum_{k=1}^{m} a_k x_k - B = 0$. Hence, SubSum$(I)$ is a "yes" instance. □

5.6 Notes

The inapproximability results of CKP$[\frac{\pi}{2} + \delta, \pi]$ and CKP$[\pi - \delta', \pi]$ were obtained in Chau *et al.* (2016). It was shown in Woeginger (2000) and Yu and Chau (2013), independently, that there exists no FPTAS for CKP$[0, \frac{\pi}{2}]$. The hardness results of OPF with voltage constraint and discrete demands, and OPF with power capacity constraint and discrete demands were obtained in Khonji *et al.* (2018a). The hardness results of simplified OPF were also obtained in Khonji *et al.* (2018a). We note that there are other hardness results of OPF Lehmann *et al.* (2016), Verma (2009), and Bienstock and Verma (2015). But there are several differences compared to our hardness results. The results in Lehmann *et al.* (2016), Verma (2009), and Bienstock and Verma (2015) consider a different set of constraints, namely the phase angle difference on each link is bounded by some threshold, in addition to the voltage constraints. In our paper, we consider, either voltage constraints alone, or power capacity constraints alone (the latter can be related to phase angle constraints). The setting in Lehmann *et al.* (2016), Verma (2009), and Bienstock and Verma (2015) with no binary variables implies that checking feasibility is already NP-hard; on the other hand, since we allow binary variables associated with loads in our setting, the all-zero solution (and all voltages equal to $\underline{v}$ in case $\underline{v} > 0$) is trivially feasible. In this case, the non-trivial question is about optimization rather than decision. While the results in Lehmann *et al.* (2016), Verma (2009), and Bienstock and Verma (2015) show NP-hardness of (continuous) AC feasibility, we study the discrete problem. We show hardness of approximation, even if we allow the capacity/voltage constraints to be violated by some multiplicative parameter β.

6

Simulation Studies

We provided analysis on the approximations ratios of our algorithms in the previous chapters, which are the worst-case guarantees. In this chapter, we evaluate the empirical average-case ratios by simulations. We observe that our algorithms perform relatively well in several scenarios which are far below the theoretical worst-case values.

6.1 Simulation Settings

To evaluate the performance of the algorithms presented in earlier chapters, we consider a distribution network and over thousands of users. Each user has a specific power demand (including both active and reactive power) and a utility (or a cost) that is generated according to a probability preference model. In a micro grid (MG), the amount of generation is typically less than the amount of demand and thus, the users may suffer from a reduction of generation capacity occasionally. Various types of loads are considered including residential and industrial users ranging between 300KVA to 1MVA. We also assume that the distribution network is equipped with a two-way communication infrastructure capable of sending the optimal load management signals (determined by the centralized controller) and allowing for user demand and utility to be sent back to the centralized controller. The central controller is assumed to have full control over the on/off operations of its users.

We consider diverse case studies of various settings of power demands by taking into account the correlation between user demand and utility (resp. cost) considering various demand types. The following are the settings of power demands at the users:

(i) *Utility-demand correlation*:

  (a) *Correlated setting (C)*: The utility of each user is a function of the power demand:

$$u_k = a \cdot |\overline{s}_k|^2 + b \cdot |\overline{s}_k| + c\,, \tag{6.1}$$

  where $a > 0, b, c \geq 0$ are constants. For simplicity, $u_k = |\overline{s}_k|^2$ is considered in the simulation.

  (b) *Uncorrelated setting (U)*: The utility of each user is independent of the power demand and is generated randomly from $[0, |\overline{s}_{\max}(k)|]$. Here $\overline{s}_{\max}(k)$ depends on the user type (as defined below): if user k is an industrial user then $|\overline{s}_{\max}(k)| =$ 1MVA, otherwise $|\overline{s}_{\max}(k)| =$ 5KVA.

(ii) *User types*:

  (a) *Residential (R) users*: The users are comprised of residential users having small power demands ranging from 500VA to 5KVA.

  (b) *Mixed (M) users*: The users are comprised of a mix of industrial and residential users. Industrial users have big power demands ranging from 300KVA up to 1MVA and constitute no more than 20% of all users chosen at random.

In this chapter, the case studies will be represented by the aforementioned acronyms. For example, the case study named CM stands for the one with utilities-demand correlation and mixed users. The power factor for each user varies between 0.8 to 1 (to comply with

IEEE standards) and thus we restrict the phase angle θ of demands to be in the range of $[-36°, 36°]$. The algorithms were implemented using Python programming language with Scipy library for scientific computation. In order to quantify the performance of the proposed algorithms, Gurobi optimizer (Gurobi Optimization, 2017) is employed to obtain the close-to-optimal solutions numerically. The following parameters were set in Gurobi optimizer: (1) the total time expended for solving the problem was 200 seconds, (2) absolute mixed integer programming (MIP) optimality gap (i.e., the threshold of the absolute gap between the lower and upper objective bound) was set to zero, and (3) infeasibility tolerance was set to 10^{-9}. It is worth noting that there are no guarantees that given an integer programming problem the optimizer will return an optimal solution nor it will terminate in a reasonable time (i.e., within 200 seconds for each run). Whenever the optimizer exceeds the time limit, the current best solution is considered to be optimal.

6.2 Single-capacitated AC Electric Power Systems

In this section, we evaluate the performance of `GreedyRatio`, presented in Sec. 2.2. A micro grid (MG) with an overall capacity of 2MVA is considered. The simulations were evaluated using 2 Quad core Intel Xeon CPU E5607 2.27 GHz processors with 12 GB of RAM.

6.2.1 Optimality

In this section, we compare `GreedyRatio`, written as `GRA` for short, with two other greedy algorithms that follow from the conventional strategies:

1. *Greedy Utility Algorithm* (`GUA`): First, sort the users in $\mathcal{N} = \{1, ..., n\}$ by their utilities in a non-increasing order (with arbitrary tie-breaking), such that

$$u_1 \geq u_2 \geq ... \geq u_n \,. \tag{6.2}$$

   Then, select the satisfiable demands sequentially from the first user according to the order whenever feasible (i.e. $|\sum_k \overline{s}_k x_k| \leq \overline{S}$).

2. *Greedy Demand Algorithm* (GDA): Similar to GUA, but sort the users by the magnitudes of their demands in a non-decreasing order, such that

$$|\overline{s}_1| \leq |\overline{s}_2| \leq ... \leq |\overline{s}_n|. \tag{6.3}$$

Then, select the satisfiable demands sequentially from the first user according to the order whenever feasible (i.e. $|\sum_k \overline{s}_k x_k| \leq \overline{S}$).

In this subsection the proposed approaches are compared in terms of quality of solution. The optimal solutions computed by Gurobi, denoted by Opt, are considered to be the base case for the comparison. The algorithms are applied to various case studies where each case study is analyzed considering changes in the set of users. As an example, GRA is applied 30 times for each of the m number of users (where m varies between 100 to 1500 in steps of hundred) for case study CR (i.e., correlated, residential) considering random changes in demands and utilities of users. Thus, the total number of experiments for each case study is 450. In particular, Table 6.1 highlights the results obtained using GRA, GUA, and GDA for the various case studies. The results in Table 6.1 present the minimal ratio between the solutions obtained by the proposed algorithms and Gurobi. It is worthy to note that the closer this value is to 1, the closer is the solution to the optimum.

		C	**U**
GRA	**R**	0.999	0.883
	M	0.921	0.568
GUA	**R**	0.999	0.934
	M	0.921	0.403
GDA	**R**	0.541	0.839
	M	0.002	0.262

Table 6.1: The worst-case approximation ratios of GRA, GUA, and GDA with respect to the optimal solutions computed by Gurobi optimizer.

As can be observed from Table 6.1 when there is a utility-demand correlation for all users, GRA and GUA obtain solutions that are very close to the optimal Opt. Furthermore, for all the case studies GRA provides the highest approximation ratio in the worst-case (worst-case of GRA is 0.568, while that of GUA is 0.403) when compared to other candidate algorithms. Fig. 6.1a presents the maximized utility for the three algorithms at 95% confidence interval considering different user set cardinality. Similarly, each point presented in Fig. 6.1a represents the average objective value among 30 iterations. It was observed experimentally that 30 iterations was sufficient for convergence of the sample mean and variance.

It is observed that GDA performs the worst in terms of the quality of solution. In fact, GDA performance degrades significantly as the number of users increase when there is correlation between utilities and demands. The reason is that GDA heuristically selects the smallest demands first which on contrary obtain lower utility when considering a quadratic utility function. This situation does not occur when there is no such correlation (namely, in case studies UR and UM). For case study UR, the observed performance of GUA is slightly better than that of GRA when considering large user participation. This could be due to the fact that with increasing user participation the probability of having a user with a small utility but high utility to demand ratio increases. Unlike GUA, GRA selects those users first thus failing to consider the users with relatively large utilities. All algorithms can output optimal solutions when having only few residential users, because at this scale all users' demands are below the total capacity of 2MVA and hence can be all satisfied (see case studies CR and UR in Fig. 6.1a).

The observed approximation ratios are plotted in Fig. 6.1b against the number of users for each scenario. When a curve is close to the line $y = 1$, it indicates a close to optimal solution. The observed approximation ratios of GRA and GUA are approximately 1 for case studies CR and CM for scalable number of users. As highlighted earlier, the theoretical guarantees on the worst-case approximation ratio of GRA is 0.475, computed using $\theta = 36$ degrees (refer to Theorem 2.1). Nevertheless as can be observed from Table 6.1 GRA, for the majority of cases, can achieve almost twice higher approximation ratio compared to the theoretical bound.

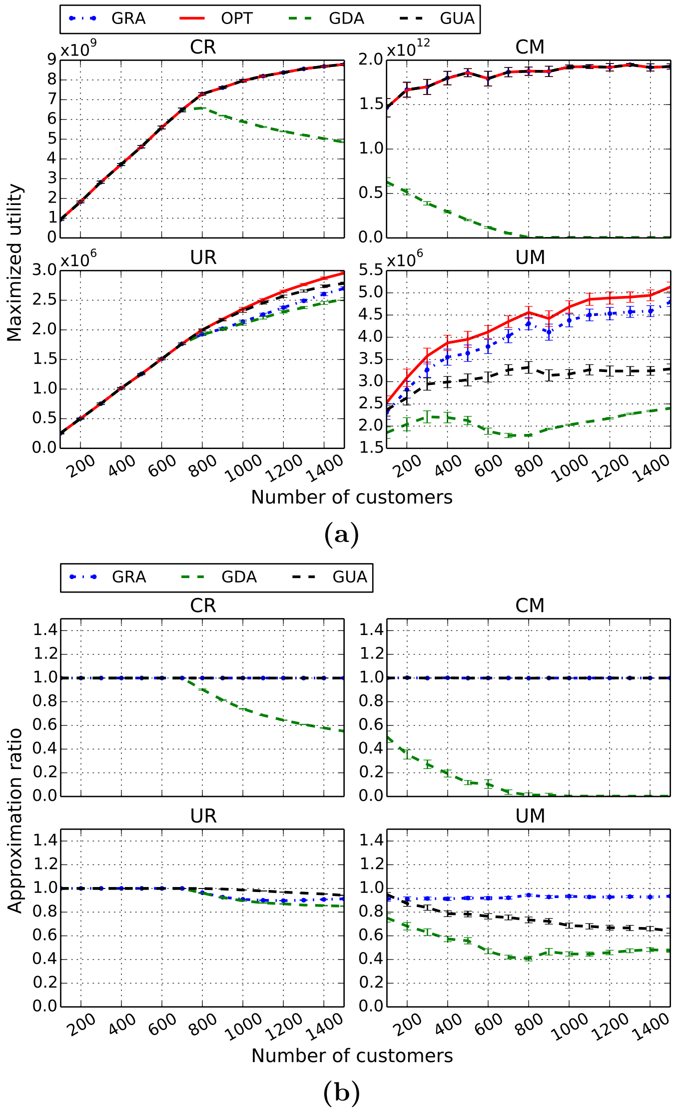


Figure 6.1: (a) The average approximation ratios of `GRA`, `GUA` and `GDA` against the number of users at 95% confidence interval. (b) The average objective values of `GRA`, `GUA`, `GDA` and OPT against the number of users at 95% confidence interval.

6.2.2 Running Time

As stated earlier a major problem with most of the available centralized demand management schemes is the computational time. It is very important to develop fast and efficient algorithms capable of executing optimal decisions when considering significant user participation. Thus, one of the key parameters utilized to evaluate the performance of the proposed algorithm is the computational time.

In this subsection the computational time of the proposed `GRA` is compared against the Gurobi solver. Computational time is of significant importance when designing centralized controllers for MGs since this will have implications on the stability of MG. Note that the running time complexity of `GUA` and `GDA` is the same as that of `GRA`. For clarity of presentation, however, we investigate only the running time of `GRA`.

The running time of `GRA` is compared to that of Gurobi optimizer in Fig. 6.2. For brevity, only the results related to a case study CR are presented in Fig. 6.2. However, it is worthy to note that nearly the same running time was observed for `GRA` and Gurobi optimizer when considering the rest case studies. The computational time reported is the average running time over 30 iterations. As can be observed, for an MG with roughly 600 users the centralized controller operated using Gurobi solver will take roughly 5 seconds.

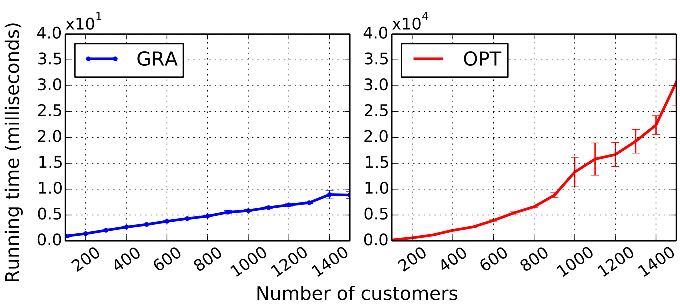


Figure 6.2: The average running time of `GRA` (left) and OPT (right) against the number of users at 95% confidence interval.

In these case studies it is expected that the MG will not be capable of maintaining stable operation. On the contrary, for the same number of users the computational time needed for the `GRA` is nearly 5 milliseconds. This would allow load management decisions to be made almost instantaneously and thus allowing for the MG to stabilize.

For an MG with up to 1400 users, the `GRA` is capable of providing close to optimal solutions in less than 10 milliseconds. Furthermore, it is worthy to mention that for some cases the Gurobi optimizer did not return an optimal solution within 200 seconds, but on the contrary `GRA` can always output a solution in nearly linear time.

6.3 Constant-Sized AC Electric Power Networks

6.3.1 Maximizing OPF

We consider two electric networks: a 38-node system adopted from (Singh *et al.*, 2007) (see Fig. 6.3), and the de-facto IEEE 123-node system.

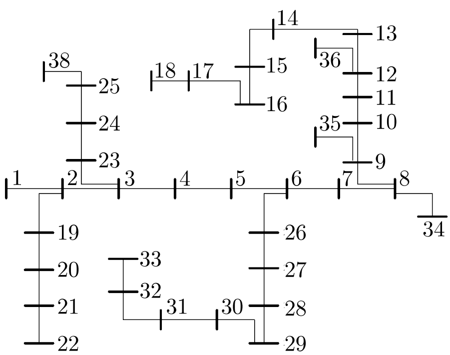


Figure 6.3: A 38-node electric network for evaluation.

For the 38-node system, the settings of line impedance and maximum capacity are provided in Table. 6.2.

In 38-node system, we assume that the generation source is attached to node 1, whereas the power demands are randomly generated at other 37 nodes uniformly.

Network Data											
Bus	**R** (p.u.)	**X** (p.u.)	**Cap.** (p.u.)	**Bus**	**R** (p.u.)	**X** (p.u.)	**Cap.** (p.u.)	**Bus**	**R** (p.u.)	**X** (p.u.)	**Cap.** (p.u.)
$(0,2)$	0.000574	0.000293	4.6	$(14,15)$	0.00368	0.003275	0.3	$(27,28)$	0.006594	0.005814	1.5
$(2,3)$	0.00307	0.001564	4.1	$(15,16)$	0.004647	0.003394	0.25	$(28,29)$	0.005007	0.004362	1.5
$(3,4)$	0.002279	0.001161	2.9	$(16,17)$	0.008026	0.010716	0.25	$(29,30)$	0.00316	0.00161	1.5
$(4,5)$	0.002373	0.001209	2.9	$(17,18)$	0.004558	0.003574	0.1	$(30,31)$	0.006067	0.005996	0.5
$(5,6)$	0.0051	0.004402	2.9	$(2,19)$	0.001021	0.000974	0.5	$(31,32)$	0.001933	0.002253	0.5
$(6,7)$	0.001166	0.003853	1.5	$(19,20)$	0.009366	0.00844	0.5	$(32,33)$	0.002123	0.003301	0.1
$(7,8)$	0.00443	0.001464	1.05	$(20,21)$	0.00255	0.002979	0.21	$(8,34)$	0.012453	0.012453	0.5
$(8,9)$	0.006413	0.004608	1.05	$(21,22)$	0.004414	0.005836	0.11	$(9,35)$	0.012453	0.012453	0.5
$(9,10)$	0.006501	0.004608	1.05	$(3,23)$	0.002809	0.00192	1.05	$(12,36)$	0.012453	0.012453	0.5
$(10,11)$	0.001224	0.000405	1.05	$(23,24)$	0.005592	0.004415	1.05	$(18,37)$	0.003113	0.003113	0.5
$(11,12)$	0.002331	0.000771	1.05	$(24,25)$	0.005579	0.004366	0.5	$(25,38)$	0.003113	0.003113	0.1
$(12,13)$	0.009141	0.007192	0.5	$(6,26)$	0.001264	0.000644	1.5	$S_{\text{base}} = 1\text{MV}$, $V_{\text{base}} = 12.66\text{KV}$			
$(13,14)$	0.003372	0.004439	0.45	$(26,27)$	0.00177	0.000901	1.5				

Table 6.2: The settings of line impedance and maximum capacity of the 38-node electric network.

The IEEE 123-node network is an unbalanced three-phase network with several devices that are not modeled in our formulation. As in (Gan *et al.*, 2015), we modify the IEEE network by the following:

- The three phases are assumed to be decoupled into three identical single phase networks.
- Closed circuit switches are modeled as shorted lines and ignore open circuit switches.
- Transformers are modeled as lines with appropriate impedances.

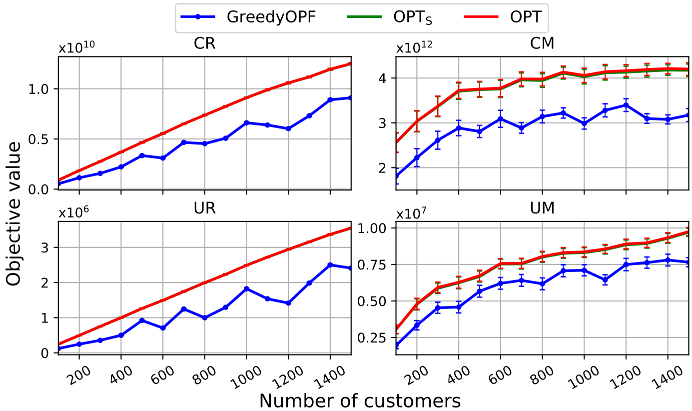
We assume that the generation source is attached to the substation (node 150), whereas the power demands are randomly generated at the other nodes uniformly.

In order to quantify the performance of our algorithms, we use Gurobi optimizer to obtain numerically close-to-optimal solutions for OPF and sOPF respectively. We denote output solution for OPF (resp., sOPF) obtained by Gurobi optimizer by Opt (resp., Opt_s). To ensure the feasibility of Opt_s, we perform a linear search similar to that in Algorithm 10 (see Sec. 3.4) with a small modification. Note that there is no guarantee that the optimizer will return an optimal solution nor it will terminate in a reasonable time (e.g., within 200 seconds for each run). Whenever the optimizer exceeds the time limit, the current best solution is considered to be optimal. We set the step size to be $\epsilon = 0.005$ (i.e., 0.5%) for both `GreedyOPF` and Opt_s.

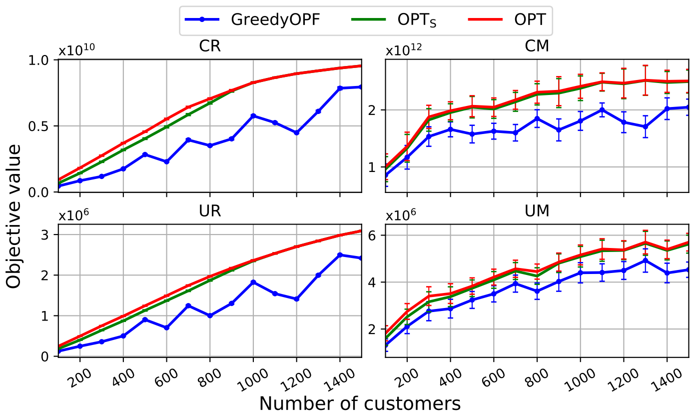
The simulations were evaluated using 2 Quad core Intel Xeon CPU E5607 2.27 GHz processors with 12 GB of RAM.

Optimality

Fig 6.4a (resp., 6.4b) presents the objective value attained by `GreedyOPF` with only inelastic demands, Opt, and Opt_s respectively using the 38-node system (resp., the IEEE 123-node system) for up to 1500 users. Each run is repeated 40 times. The utility values attained by Opt and Opt_s are almost identical in all scenarios. This is due to the insignificance of the terms associated with transmission power loss in OPF. We observe from the figure that `GreedyOPF` performs relatively



(a) 38-node network

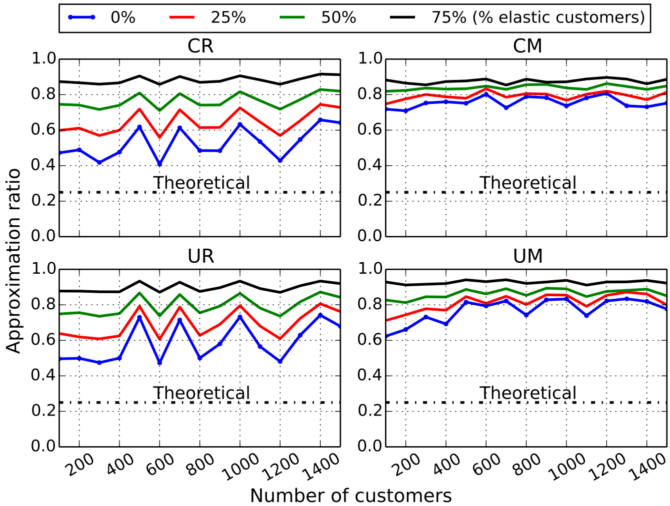


(b) IEEE 123-node network

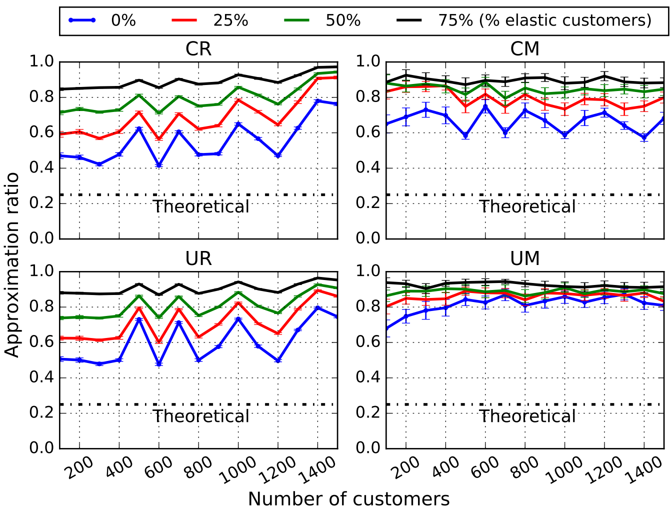
Figure 6.4: The average objective values of `GreedyOPF` with inelastic demands only against the number of users with 95% confidence interval.

better when loads are mixed between residential and industrial (CM and UM).

We note that `GreedyOPF` objective does not smoothly increase in the number of users which is due to the way users are arranged into different groups in algorithm `GreedysOPF`. User utility is rounded by the factor L which is a function of the number of users. We observe from the figure that such rounding sometimes obtains lower utilities by increasing the number of users.



(a) 38-node network



(b) IEEE 123-node network

Figure 6.5: The average approximation ratios of `GreedyOPF` applied to instances with different percentage of elastic demands, against the number of users with 95% confidence interval.

The empirical approximation ratios for the two networks are plotted in Fig. 6.5a and 6.5b against the number of users, along with the theoretical approximation ratio given by Theorem 3.11 part 1. The lines in Fig. 6.5a (resp., 6.5b) correspond to different percentages of elastic demands (i.e., $\frac{|\mathcal{F}|}{|\mathcal{N}|} = 0, 0.25, 0.50, 0.75$). As the percentage of elastic demands increases, `GreedyOPF` consistently achieves better solutions in all scenarios. The average empirical ratios are more than 0.4 in all cases which is well above the theoretical worst case results. This suggests that `GreedyOPF` performs relatively well in practice under difference scenarios.

Transmission Power Loss

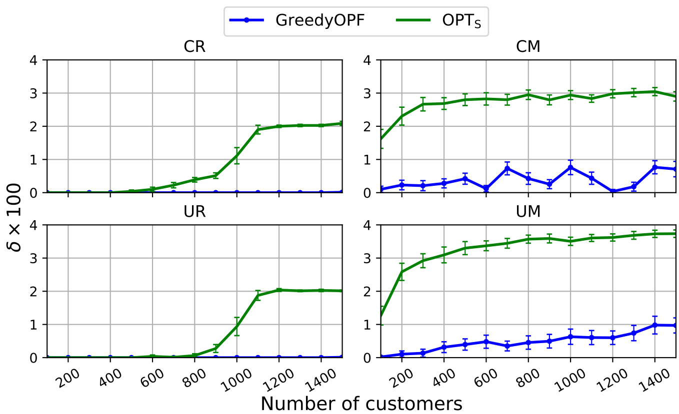
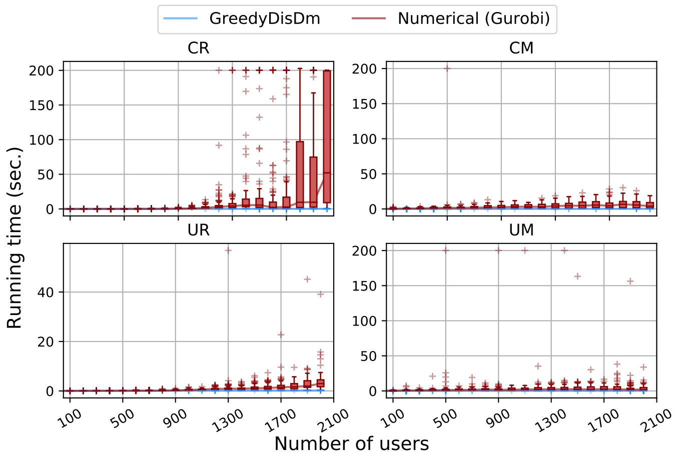


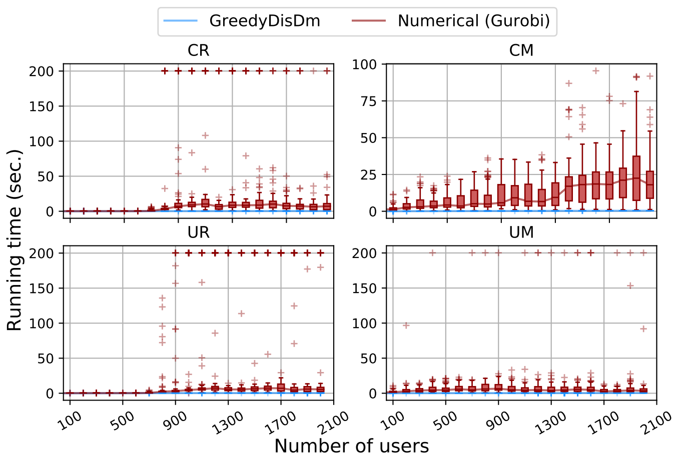
Figure 6.6: The average loss of `GreedyOPF` with inelastic demands only and $\textsc{Opt}_s$ for the 38 node system.

To understand the transmission power loss in practice, we evaluate the loss ratio (i.e., δ) in `GreedyOPF` (with inelastic demands only) and $\textsc{Opt}_s$ respectively for the 38-node system. The results are plotted in Fig. 6.6. As one may expect, $\textsc{Opt}_s$ has a higher loss percentage since it satisfies more demands than `GreedyOPF` in general. We observe that when users are all residential, `GreedyOPF` always obtains feasible solutions without any reduction in link capacities (i.e., $\delta = 0$). The maximum loss ratio obtained is 5.5% in UM scenario for both $\textsc{Opt}_s$ and `GreedyOPF`. The ramification is that `GreedyOPF` can attain a good empirical approximation ratio in practice, because the transmission power loss is usually small in practical electric networks.

Running Time



(a) 38-node netowrk



(b) IEEE 123-node network

Figure 6.7: The average running time of `GreedyDisDm` and Opt for (a) the 38-node system and (b) the IEEE 123-node system against the number of users.

One of the main goals of this work is to develop efficient algorithms that ensure a polynomial running time. The computational time of

`GreedyDisDm` is compared against the Gurobi solver. Computational time is of significant importance when designing centralized controllers for micro-grids since this will have implications on the overall stability. The running time is presented in Fig. 6.7 under different scenarios for up to 2000 users, each point is repeated 100 times.

We observe the running time of `GreedyDisDm` is always in milliseconds and linearly increases in the number of users n. On the other hand, the running time of OPT is much higher in many cases (measured in minuets) and has no polynomial guarantee. Throughout the simulations, we observed many timeouts especially in scenario CR. The actual running time of OPT may substantially increase if we increase the timeout parameter in Gurobi optimizer. The running time of OPT can be much higher if we consider larger network topologies, whereas, linear increase is expected for `GreedyDisDm` in practice. Therefore, our algorithm is far more scalable than any known optimal algorithm. We note that the implementation of our algorithms can be further optimized using C programming language since the current one is based on Python that is relatively slow.

6.3.2 Minimizing OPF

In this subsection, we evaluate `PTAS-cOPF` on the Bus 4 distribution system of the Roy Billinton Test System (RBTS) (Allan *et al.*, 1991; Huang *et al.*, 2017), which comprises of 13 nodes, in which the generation source is attached to the sub-station node 0, the base power capacity of this network is 8MVA, and the base voltage is 11KV. The evaluation was also performed on the IEEE 123-node network, in which the generation source is attached to node 150, and the base capacity and voltage are 5MVA and 4.16KV, respectively.

The single-line diagram and line data of the RBTS 13-node network are presented in Fig. 6.8 and Table 6.3, respectively.

We consider the following scenarios for *Cost-Demand Correlation*:

(a) *Correlated Setting* (C): The cost objective of each user is a function of his power demand:

$$h_k(\mathrm{Re}(s_k)) = \left(|\overline{s}_k| - \tfrac{\mathrm{Re}(s_k)}{\mathrm{Re}(\overline{s}_k)}|\overline{s}_k|\right)^2.$$

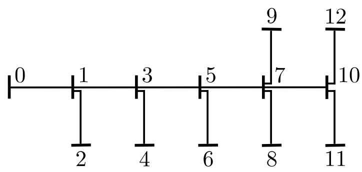


Figure 6.8: Single-line diagram of the RBTS 13-node electric network.

(b) *Uncorrelated Setting* (U): The cost objective of each user is independent of his power demand and is generated randomly from $[0, |\overline{s}_{\max}(k)|]$. Here $\overline{s}_{\max}(k)$ depends on the user type. If user k is an industrial user then $|\overline{s}_{\max}(k)| = 1\text{MVA}$, otherwise $|\overline{s}_{\max}(k)| = 5\text{KVA}$. More precisely, given a random $r \in |\overline{s}_{\max}(k)|$,

$$h_k(\mathrm{Re}(s_k)) = r - \tfrac{\mathrm{Re}(s_k)}{\mathrm{Re}(\overline{s}_k)} r.$$

As in previous sections, the case studies will be represented by the acronyms; for example, the case study named "CM" stands for the one with mixed users and correlated cost-demand setting. The simulations were evaluated using Intel i7-3770 CPU 3.40GHz processor with 32GB of RAM.

Bus	R (p.u.)	X (p.u.)	Capacity (p.u.)
$(0, 1)$	0.011636363636364	0.034380165289256	1
$(1, 2)$	0.026446280991736	0.158677685950413	0.125
$(1, 3)$	0.014545454545455	0.043636363636364	0.7625
$(3, 4)$	0.026446280991736	0.158677685950413	0.25
$(3, 5)$	0.017454545454546	0.042314049586777	0.75
$(5, 6)$	0.026446280991736	0.158677685950413	0.25
$(5, 7)$	0.011636363636364	0.03702479338843	0.75
$(7, 8)$	0.026446280991736	0.171900826446281	0.25
$(7, 9)$	0.031735537190083	0.185123966942149	0.25
$(7, 10)$	0.014545454545455	0.039669421487603	0.75
$(10, 11)$	0.013223140495868	0.161322314049587	0.25
$(10, 12)$	0.029090909090909	0.185123966942149	0.25

Table 6.3: Settings of line impedance and maximum capacity of the RBTS 13-node electric network.

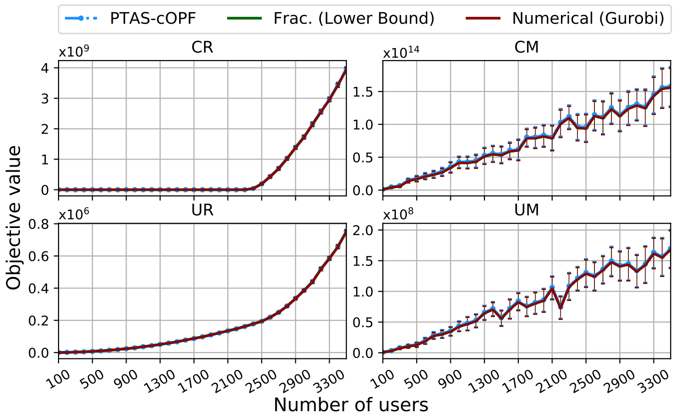
Optimality

Fig 6.9a (resp., 6.9b) presents the objective values obtained by `PTAS-cOPF`, Gurobi numerical solver, and the lower bounds to the true optimal values by fractional solutions with relaxed discrete demands (i.e., setting all $x_k \in [0, 1]$) respectively using the RBTS 13-node network (resp., IEEE 123-node network) for up to 3500 users. Each run was evaluated with over 40 random instances. `PTAS-cOPF` will terminate, when its objective value is close to the lower bound. The objective values of `PTAS-cOPF` are often close to the true optimal values. This is because the number of fractional components in the relaxed problem P1 is often small (see Sec. 3.3). Fig. 6.10 shows the ratio of fractional components over $4m$ (where m is the number of links) is close to 10%, which stays small when the number of users increases.

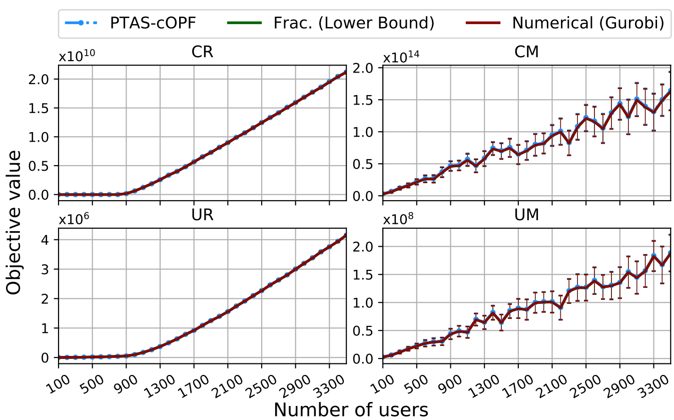
The empirical approximation ratios of `PTAS-cOPF` for the two networks are plotted in Fig. 6.11 against the number of users. We observe that the empirical approximation ratio is close to 1.2 in most cases. There are few instances with a larger empirical approximation ratio, but increasing partial guessing is able to resolve this issue, still within polynomial running time.

Running Time

The computation time of `PTAS-cOPF` is compared against that of Gurobi numerical solver. The running time is plotted in Fig. 6.12 under different case studies for IEEE 123-node network. Although the current implementation of `PTAS-cOPF` is not fully optimized, its running time is still substantially better than that of Gurobi, and is observed to scale linearly as the number of users. On the other hand, the running time of Gurobi is much higher in many cases, which does not provide any guarantee on the termination of execution, if timeout is not set. Many instances experienced timeouts, especially for the case study UR. The actual running time of Gurobi may substantially increase if the timeout value is further increased.



(a) RBTS 13-node network



(b) IEEE 123-node network

Figure 6.9: The average objective values of `PTAS-cOPF`, Gurobi numerical solver, and fractional solutions with relaxed discrete demands (as the lower bounds to the true optimal values), against the number of users with 95% confidence intervals.

6.4 Scheduling of AC Electric Power

In this section, we evaluate the empirical performance of `CSP-PTAS`, presented in Chapter 4. For simplicity, we considered a fixed generation capacity at 2MVA, and a scheduling horizon of 24 time steps, where each step represents 1 hour duration. Each user k has a single preference ($|D_k| = 1$) that arise at a uniformly random time step and remains for

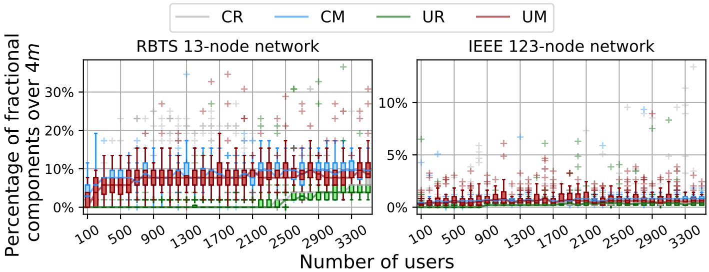


Figure 6.10: The ratio of fractional components after solving P1. The '+' points represent the outliers.

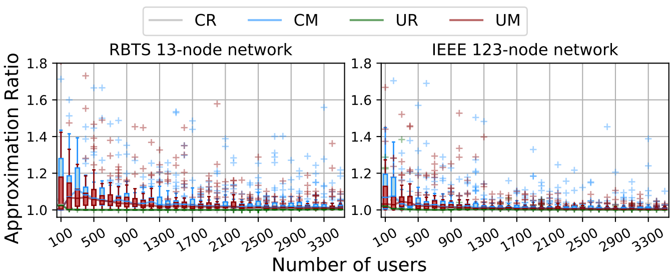


Figure 6.11: The empirical approximation ratios of `PTAS-cOPF` for different case studies, against the number of users.

a uniformly random duration. The simulations were evaluated using Intel i7-3770 CPU 3.40GHz processor with 32GB of RAM.

Optimality

Fig 6.13 presents the objective values obtained by `CSP-PTAS`, Gurobi numerical solver, and the upper bounds to the true optimal values by fractional solutions with relaxed discrete demands (i.e., setting all $x_k \in [0, 1]$) for up to 3500 users. Each run was evaluated with over 40 random instances. The objective values of `CSP-PTAS` are often close to the true optimal values. This is because, as in the previous section, the number of fractional components in the relaxed problem is small. Fig. 6.14b shows the ratio of fractional components over 48, the number of inequality constraints in LP (see Sec. 4.2), which remains fixed as we increase the number of users.

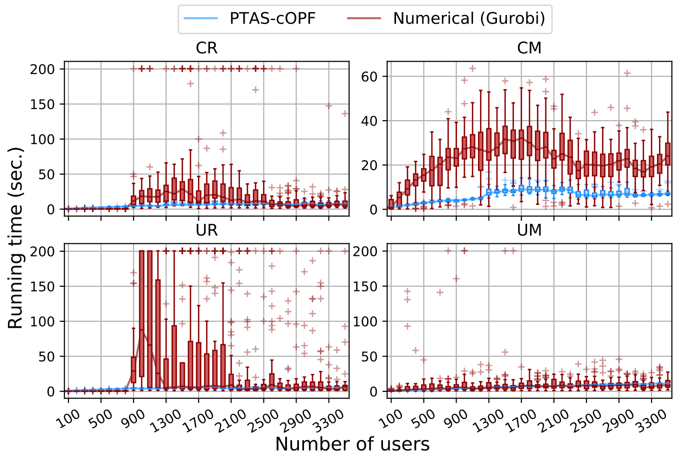


Figure 6.12: The median of running times of `PTAS-cOPF` and Gurobi numerical solver for different case studies in IEEE 123-node network.

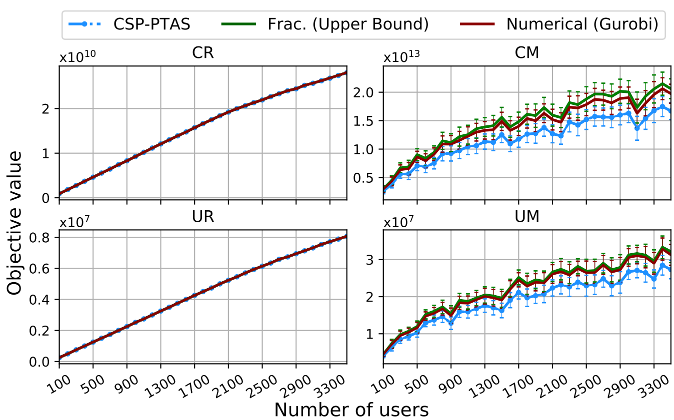
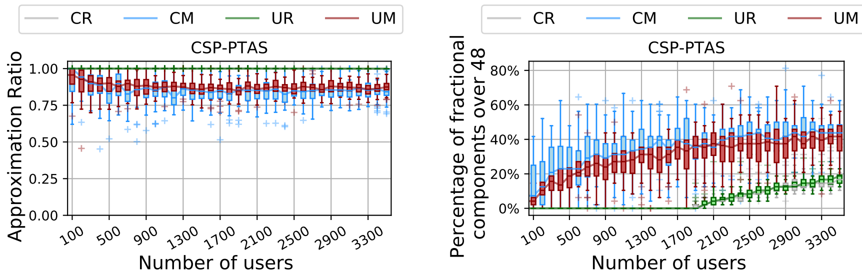


Figure 6.13: The average objective values of `CSP-PTAS`, Gurobi numerical solver, and fractional solutions with relaxed discrete demands (as upper bounds to the true optimal values), against the number of users with 95% confidence intervals.

The empirical approximation ratios of `CSP-PTAS` is plotted in Fig. 6.14a against the number of users. We observe that the empirical approximation ratio is close to optimal in most cases.



(a) **(b)**

Figure 6.14: (a) The empirical approximation ratios of `CSP-PTAS` for different case studies, against the number of users; and (b) The ratio of fractional components after solving LP.

Running Time

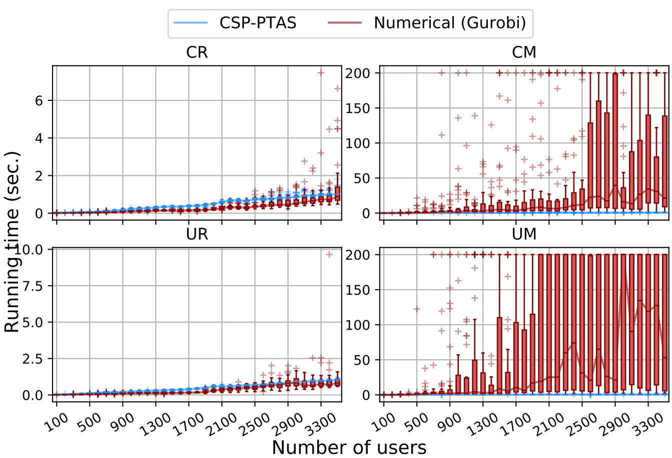


Figure 6.15: The median of running times of `CSP-PTAS` and Gurobi numerical solver for different case studies.

The computation time of `CSP-PTAS` is compared against that of Gurobi numerical solver. The running time is plotted in Fig. 6.15 Although the current implementation of `CSP-PTAS` is not fully optimized, its running time is still substantially better than that of Gurobi, and is observed to scale linearly as the number of users. On the other hand, the running time of Gurobi is much higher in many cases, which does not provide any guarantee on the termination of execution, if timeout is not set. Many instances experienced timeouts, especially for the case study UR.

6.5 Notes

The simulation results for the single link case, algorithm `GreedyRatio`, was presented in (Karapetyan *et al.*, 2018). For constant-sized network, the simulations for algorithm `GreedyOPF` was presented in (Khonji *et al.*, 2018a), while `PTAS-cOPFs` presented in (Khonji *et al.*, 2017). We observed that the performance of Gurobi solver for the OPF problem, both in terms of optimality and running time, is improved when the problem is modeled using the tree formulation for OPF (see Sec. 3.1.2). In many occasions when using the original recursive formulation, Gurobi optimizer fails to return feasible solution.

7

Future Directions and Conclusion

This monograph presented a study of combinatorial OPF with discrete demands, covering basic single-capacitated AC electric power systems, constant-sized AC electric grid networks with power flows, and scheduling of AC electric power. Our goal is to bridge power systems engineering and theoretical computer science, establishing a foundation for advancing the frontiers of both communities.

It is believed that such a bridge between power systems engineering and theoretical computer science will not be a short-lived endeavor. In fact, there are many open possibilities for further impactful results, based on the foundation laid in this monograph. In the following, we outline several extensions for future work.

7.1 Scalable Algorithms for Large Power Networks

In Chapter 3, we have given approximation algorithms for OPF with discrete demands, assuming a radial network with a fixed number of buses. While the hardness results in Chapter 5 show that is unlikely that (efficient) approximation algorithms can be obtained for general networks, it remains an interesting question whether the assumption on the network size can be dropped, assuming a tree network. Preliminary results indicate that this is indeed the case: a QPTAS for line networks with scalable network size is obtained in Elbassioni *et al.* (2017). Extending this work to general tree networks, under (some of) the assumptions in Section 3.1.1 is an interesting direction for future research. The main idea, as in Elbassioni *et al.* (2017), would be to relate the OPF problem on radial networks to the UFP problem on trees, via Corollary 3.7, and extend known techniques for UFP on trees such as the ones in Anagnostopoulos *et al.* (2013), Chekuri *et al.* (2009), and Chekuri *et al.* (2007).

7.2 Online Algorithms

Today's smart grid requires timely control decision-making in dynamic environments, while ensuring the robustness of electric networks. There is a high probability that a microgrid once operating in isolated mode will be short of power. This is specially the case when the microgrid encompasses a hybrid mix of traditional and renewable energy supplies that could collectively have a variable (depending on the availability of renewable energy and storage available) yet dispatchable capacity. This induces time-varying generation capacity. Constrained by the generation fluctuating over time, it will be required to make binary control decisions in real time so as to maximize the total utility of satisfied customers. Another interesting scenario is when the customers demands are not known in advance but are revealed one at a time. At each time step, the so-called *online* algorithm has to make irreversible decisions so as to satisfy the system's constraints while keeping the objective function as close as possible to the one produced by an algorithm that foresees all the future input. The typical framework to address this type of problems is through *competitive analysis*, where the performance of an online algorithm is compared to that of the best *offline* algorithm. A general primal-dual framework for developing competitive online algorithms for binary packing problems Buchbinder and Naor (2009) can be extended to solve the scheduling problem studied in Chapter 4; see Karapetyan *et al.* (2017). Extensions of these results to the more general OPF problem on trees is an interesting research direction.

7.3 Truthful Mechanisms

Future smart grids will be automated by agents representing individual users. Hence, one might expect these agents to be *self-interested* and may *untruthfully* report their valuations or demands. This motivates us to consider *truthful* (aka. *incentive-compatible*) approximation mechanisms, in which it is in the best interest of the agents to report their true parameters. In Yu and Chau (2013) a monotone 1/2-approximation algorithm that induces a deterministic truthful mechanism was devised for the CKP problem, which, however, assumes that all complex-valued demands lie in the positive quadrant. In Chau *et al.* (2016), a more complete study of truthful mechanisms for CKP was given: a truthful PTAS for the case $\phi \in [0, \frac{\pi}{2} - \delta]$, and a truthful resources augmented FPTAS for the case $\phi \in (\frac{\pi}{2}, \pi - \delta]$, where ϕ is the maximum argument of any complex-valued demand and $\epsilon, \delta > 0$ are arbitrarily small constants. Extension of these results to the OPF problem on trees is an interesting research direction. More specifically, in Corollary 3.7, we have shown how to use an LP-rounding approximation algorithm for a generalization of the unsplittable flow problem, to obtain an approximation algorithm for OPF. Based on this result and the approach in Lavi and Swamy (2011), it seems plausible that one might be able to obtain a truthful-in-expectation mechanism for OPF on trees, if an LP-based approximation algorithm can be developed for the same problem.

7.4 Efficient Algorithms for SOCP Relaxation of OPF

General convex programming solvers are typically computationally expensive and do not use the special structure of cOPF. Recently, more efficient *first-order methods* have been applied successfully to a number of special convex optimization problems (see e.g., Arora *et al.* (2012) and M.D. Grigoriadis and Villavicencio (2001)), and more specifically to a class of second-order cone programs Elbassioni *et al.* (2016). It would be interesting to study the extension of these and similar methods for solving the SOCP relaxation of OPF.

Acknowledgements

The authors are grateful to various colleagues and collaborators for helpful comments and constructive suggestions, such as Areg Karapetyan, Hatem Zeineldin, and Trung Thanh Nguyen.

We would like to extend our gratitude to Maria Ilic, Editor-in-Chief, and anonymous reviewers for helpful suggestions, as well as James Finlay and Mike Casey for managing the publication of this monograph.

We also thank David Hill and Minghua Chen for hosting our talks about this monograph.

Special thank goes to Steven Low and his group for their theoretical foundations that make this work possible.

References

Allan, R. N., R. Billinton, I. Sjarief, L. Goel, and K. So (1991). “A reliability test system for educational purposes-basic distribution system data and results”. *IEEE Trans. Power Sys.* 6(2): 813–820.

Anagnostopoulos, A., F. Grandoni, S. Leonardi, and A. Wiese (2013). “Constant Integrality Gap LP Formulations of Unsplittable Flow on a Path”. In: *Proceedings of International Conference Integer Programming and Combinatorial Optimization (IPCO).* 25–36.

Anagnostopoulos, A., F. Grandoni, S. Leonardi, and A. Wiese (2014). “A mazing 2+ ε approximation for unsplittable flow on a path”. In: *Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM. 26–41.

Andrews, M., J. Chuzhoy, S. Khanna, and L. Zhang (2005). “Hardness of the undirected edge-disjoint paths problem with congestion”. In: *Annual IEEE Symposium on Foundations of Computer Science.* IEEE. 226–241.

Arora, S., E. Hazan, and S. Kale (2012). “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications”. *Theory of Computing.* 8(1): 121–164.

Azar, Y. and O. Regev (2001). “Strongly Polynomial Algorithms for the Unsplittable Flow Problem”. In: *Integer Programming and Combinatorial Optimization.* Vol. 2081. *Lecture Notes in Computer Science.* 15–29.

Bai, X., H. Wei, K. Fujisawa, and Y. Wang (July 2008). "Semidefinite programming for optimal power flow problems". *International Journal of Electrical Power and Energy Systems*. 30(6): 383–392.

Baran, M. E. and F. F. Wu (1989a). "Optimal capacitor placement on radial distribution systems". *IEEE Transactions on Power Delivery*. 4(1): 725–734.

Baran, M. E. and F. F. Wu (1989b). "Optimal sizing of capacitors placed on a radial distribution system". *IEEE Transactions on Power Delivery*. 4(1): 735–743.

Bertsimas, D. and J. N. Tsitsiklis (1997). *Introduction to Linear Optimization*. Athena Scientific.

Bienstock, D. and A. Verma (2015). "Strong NP-hardness of AC power flows feasibility". *Tech. rep.* http://arxiv.org/abs/1512.07315.

Bose, S., D. F. Gayme, K. M. Chandy, and S. H. Low (Sept. 2015). "Quadratically Constrained Quadratic Programs on Acyclic Graphs With Application to Power Flow". *IEEE Transactions on Control of Network Systems*. 2(3): 278–287.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. New York, NY, USA: Cambridge University Press. ISBN: 0521833787.

Buchbinder, N. and J. (Naor (2009). "The Design of Competitive Online Algorithms via a Primal-Dual Approach". *Foundations and Trends in Theoretical Computer Science*. 3(2–3): 93–263.

Carpentier, J. (1962). "Contribution á l'étude du dispatching économique," *Bulletin de la Société Française des Électriciens,* 3(8): 431–447.

Carpentier, J. (1979). "Optimal power flows". *International Journal of Electrical Power & Energy Systems*. 1(1): 3–15.

Chau, C.-K., K. Elbassioni, and M. Khonji (2014). "Truthful Mechanisms for Combinatorial AC Electric Power Allocation". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. http://arxiv.org/abs/1403.3907.

Chau, C.-K., K. Elbassioni, and M. Khonji (Nov. 2016). "Truthful Mechanisms for Combinatorial Allocation of Electric Power in Alternating Current Electric Systems for Smart Grid". *ACM Transactions on Economics and Computation*. 5(1). http://arxiv.org/abs/1507.01762.

Chekuri, C., A. Ene, and N. Korula (2009). “Unsplittable flow in paths and trees and column-restricted packing integer programs”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer. 42–55.

Chekuri, C., S. Khanna, and F. B. Shepherd (2006). “An O $(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow”. *Theory of computing*. 2(7): 137–146.

Chekuri, C., M. Mydlarz, and B. Shepherd (2007). “Multicommodity demand flow in a tree and packing integer programs”. *ACM Transactions on Algorithms*. 3(3): 27.

Elbassioni, K. M., K. Makino, and W. Najy (2016). “A Multiplicative Weights Update Algorithm for Packing and Covering Semi-infinite Linear Programs”. In: *International Workshop of Approximation and Online Algorithms (WAOA), Revised Selected Papers*. 78–91.

Elbassioni, K., C.-K. Chau, and M. Khonji (2017). “From Electrical Power Flows to Unsplittabe Flows: A QPTAS for OPF with Discrete Demands in Line Distribution Networks”. *Tech. rep.* http://arxiv.org/abs/1709.05876.

Elbassioni, K. and T. T. Nguyen (2015). “Approximation Schemes for Multi-objective Optimization with Quadratic Constraints of Fixed CP-Rank”. In: *Algorithmic Decision Theory*.

Farivar, M. and S. H. Low (Aug. 2013a). “Branch Flow Model: Relaxations and Convexification–Part I”. *IEEE Transactions on Power Systems*. 28(3): 2565–2572.

Farivar, M. and S. H. Low (Aug. 2013b). “Branch Flow Model: Relaxations and Convexification–Part II”. *IEEE Transactions on Power Systems*. 28(3): 2565–2572.

Frank, S., I. Steponavice, and S. Rebennack (Sept. 2012a). “Optimal power flow: a bibliographic survey I”. *Energy Systems*. 3(3): 221–258.

Frank, S., I. Steponavice, and S. Rebennack (Sept. 2012b). “Optimal power flow: a bibliographic survey II”. *Energy Systems*. 3(3): 259–289.

Gan, L., N. Li, U. Topcu, and S. H. Low (2015). “Exact convex relaxation of optimal power flow in radial networks”. *IEEE Transactions on Automatic Control*. 60(1): 72–87.

Gan, L., U. Topcu, and S. H. Low (2012). “Stochastic Distributed Protocol for Electric Vehicle Charging with Discrete Charging Rate”. In: *IEEE Power and Energy Society General Meeting*.

Grainger, J. and W. Stevenson (1994). *Power System Analysis*. McGraw-Hill.

Grötschel, M., L. Lovász, and A. Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*. Springer.

Gurobi Optimization, I. (2017). “Gurobi Optimizer Reference Manual”. URL: http://www.gurobi.com.

Guruswami, V., S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis (2003). “Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems”. *Journal of Computer and System Sciences*. 67(3): 473–496.

Huang, S., Q. Wu, J. Wang, and H. Zhao (2017). “A sufficient condition on convex relaxation of AC optimal power flow in distribution networks”. *IEEE Transactions on Power Systems*. 32(2): 1359–1368.

Jabr, R. (Aug. 2006). “Radial distribution load flow using conic programming”. *IEEE Transactions on Power Systems*. 21(3): 1458–1459.

Karapetyan, A., M. Khonji, C.-K. Chau, K. Elbassioni, and H. Zeineldin (July 2018). “Efficient Algorithm for Scalable Event-based Demand Response Management in Microgrids”. *IEEE Transactions on Smart Grid*. 9(4): 2714–2725. http://arxiv.org/abs/1610.03002.

Karapetyan, A., M. Khonji, K. Elbassioni, and C.-K. Chau (2017). “Online Algorithm for Demand Response with Inelastic Demands and Apparent Power Constraint”. *Tech. rep.* https://arxiv.org/abs/1611.00559.

Karp, R. M. (1972). “Reducibility among Combinatorial Problems”. In: *Complexity of Computer Computations. The IBM Research Symposia Series*. 85–103. ISBN: 978-1-4684-2003-6.

Kellerer, H., U. Pferschy, and D. Pisinger (2010). *Knapsack Problems*. Springer.

Khonji, M., C.-K. Chau, and K. Elbassioni (2017). “Combinatorial Optimization of AC Optimal Power Flow with Discrete Demands in Radial Networks”. *Tech. rep.* https://arxiv.org/abs/1709.08431.

Khonji, M., C.-K. Chau, and K. Elbassioni (Mar. 2018a). "Optimal Power Flow with Inelastic Demands for Demand Response in Radial Distribution Networks". *IEEE Transactions on Control of Network Systems*. 5(1): 513–524. http://arxiv.org/abs/1601.02323.

Khonji, M., S. C.-K. Chau, and K. Elbassioni (2018b). "Approximation Scheduling Algorithms for Electric Vehicle Charging with Discrete Charging Options". In: *ACM International Conference on Future Energy Systems (e-Energy) Workshop on Electric Vehicle Systems, Data, and Applications (EV-Sys)*.

Khonji, M., S. C.-K. Chau, and K. Elbassioni (2018c). "Challenges in Scheduling Electric Vehicle Charging with Discrete Charging Rates in AC Power Networks". In: *ACM International Conference on Future Energy Systems (e-Energy)*.

Khonji, M., S. C.-K. Chau, and K. Elbassioni (2018d). "Combinatorial Optimization of Electric Vehicle Charging in AC Power Distribution Networks". In: *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGrid-Comm)*.

Khonji, M., A. Karapetyan, K. Elbassioni, and C.-K. Chau (2016). "Complex-Demand Scheduling Problem with Application in Smart Grid". In: *International Conference of Computing and Combinatorics (COCOON)*. http://arxiv.org/abs/1603.01786. Springer. 496–509.

Khonji, M., A. Karapetyan, K. Elbassioni, and S. C.-K. Chau (2018e). "Complex-Demand Scheduling Problem with Application in Smart Grid". *To appear in Theoretical Computer Science*.

Kleinberg, J. M. (1996). "Approximation algorithms for disjoint paths problems". *PhD thesis*.

Lavi, R. and C. Swamy (2011). "Truthful and Near-Optimal Mechanism Design via Linear Programming". *J. ACM*. 58(6): 25.

Lehmann, K., A. Grastien, and P. V. Hentenryck (Jan. 2016). "AC-Feasibility on Tree Networks is NP-Hard". *IEEE Transactions on Power Systems*. 31(1): 798–801.

Low, S. (Mar. 2014a). "Convex Relaxation of Optimal Power Flow, Part I: Formulations and Equivalence". *IEEE Transactions on Control of Network Systems*. 1(1): 15–27.

Low, S. (June 2014b). "Convex Relaxation of Optimal Power Flow, Part II: Exactness". *IEEE Transactions on Control of Network Systems*. 1(2): 177–189.

M.D. Grigoriadis L.G. Khachiyan, L. P. and J. Villavicencio (2001). "Approximate max-min resource sharing for structured concave optimization". *SIAM Journal of Optimization*. 41: 1081–1091.

National Electrical Code (2005). *(NEC) NFPA 70-2005*.

Patt-Shamir, B. and D. Rawitz (2010). "Vector bin packing with multiple-choice". In: *Proceedings of Scandinavian conference on Algorithm Theory (SWAT)*. 248–259.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Wiley.

Singh, D., R. Misra, and D. Singh (Nov. 2007). "Effect of Load Models in Distributed Generation Planning". *IEEE Transactions on Power Systems*. 22(4): 2204–2212.

Varadarajan, K. and G. Venkataraman (2004). "Graph decomposition and a greedy algorithm for edge-disjoint paths". In: *Proceedings of annual ACM-SIAM symposium on Discrete algorithms*. 379–380.

Vazirani, V. V. (2010). *Approximation Algorithms*. Springer.

Verma, A. (2009). "Power grid security analysis: An optimization approach". *Tech. rep.* PhD diss. Columbia University.

Woeginger, G. J. (2000). "When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)?" *INFORMS Journal on Computing*. 12(1): 57–74.

Yu, L. and C.-K. Chau (2013). "Complex-Demand Knapsack Problems and Incentives in AC Power Systems". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. http://arxiv.org/abs/1205.2285.