Combinatorial Optimization of AC Electric Power Systems
Bridging Power Engineering & Computer Science
[Tutorial]

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

June 14, 2022

Big Picture

- **Challenges**
  - **Complexity**: Non-linear constraints of AC power systems, energy storage, (combined heat & power) generators, etc.
  - **Uncertainty**: Intermittent renewable energy, dynamic market prices, fluctuating demands (e.g. EVs, datacenters)

- **Goal**: How to optimize the management of energy demands and resources efficiently and intelligently?
AC Electrical Systems 101

- Circular motion of dynamo generator \(\Rightarrow\) Periodic current and voltage

\[
V = |V|e^{i\omega t}, \quad I = |I|e^{i(\omega t + \phi)}
\]

- Expressed by complex numbers:
AC Electrical Systems 101

- Circular motion of dynamo generator ⇒ Periodic current and voltage

- Expressed by complex numbers:
  \[ V = |V|e^{i\omega t}, \quad I = |I|e^{i(\omega t + \phi)} \]

- Power: \( S = V \times I^* \) (also a complex number)
  - *Active* power: \( \Re(S) \)
  - *Reactive* power: \( \Im(S) \)
  - *Apparent* power: \( |S| = \sqrt{\Re(S)^2 + \Im(S)^2} \)
AC Electrical Systems 101

- **Active power** \( (\text{Re}(S)) \)
  - Deliver useful work at loads (unit: Watt)
  - Demands: *positive* active power
  - Supplies: *negative* active power

- **Reactive power** \( (\text{Im}(S)) \)
  - Contribute to electricity flows (unit: VAR): I
  - Inductors: *positive* reactive power
  - Capacitors: *negative* reactive power
AC Electrical Systems 101

- **Active power** $(\text{Re}(S))$
  - Deliver useful work at loads (unit: Watt)
  - Demands: *positive* active power
  - Supplies: *negative* active power

- **Reactive power** $(\text{Im}(S))$
  - Contribute to electricity flows (unit: VAR):
  - Inductors: *positive* reactive power
  - Capacitors: *negative* reactive power

- **Power factor** $(\text{PF} = \frac{\text{Re}(S)}{|S|})$
  - Normalized measure of reactive power
  - Power electronic standards usually require limited reactive power in most domestic appliances ($\text{PF} \geq 0.8$)

<table>
<thead>
<tr>
<th></th>
<th>Compressors/Pumps</th>
<th>Motors</th>
<th>Machining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power factor</td>
<td>0.75-0.8</td>
<td>0.5-0.8</td>
<td>0.4-0.65</td>
</tr>
</tbody>
</table>
Electric networks: set of nodes (\( \mathcal{V} \)), set of edges (\( \mathcal{E} \))

- Voltage at node \( i \): \( V_i \)
- Impedance between \( i \) & \( j \): \( z_{i,j} \)
- Current from \( i \) to \( j \): \( I_{i,j} \)
- Transmitted power from \( i \) to \( j \): \( S_{i,j} \)
Power Networks 101

- Electric networks: set of nodes ($\mathcal{V}$), set of edges ($\mathcal{E}$)
  - Voltage at node $i$: $V_i$
  - Impedance between $i$ & $j$: $z_{i,j}$
  - Current from $i$ to $j$: $I_{i,j}$
  - Transmitted power from $i$ to $j$: $S_{i,j}$

- **Direct Current (DC) electric systems**: $(V_i, I_{i,j}, z_{i,j}, S_{i,j})$ are real numbers ($\mathbb{R}$)

- **Alternating Current (AC) electric systems**: $(V_i, I_{i,j}, z_{i,j}, S_{i,j})$ are complex numbers ($\mathbb{C}$)
Power Flow Model

- **Basic electricity laws:**
  - **Ohm’s Law:** For each edge \((i, j) \in \mathcal{E},\)
    \[
    V_i - V_j = z_{i,j} I_{i,j}.
    \]
  - **Kirchhoff’s Current Law:** For each node \(i \in \mathcal{V},\)
    \[
    \sum_{j \in \mathcal{V} : (i, j) \in \mathcal{E}} I_{i,j} = 0.
    \]
  - **Electric Power Formula:** For each edge \((i, j) \in \mathcal{E},\)
    \[
    S_{i,j} = V_i I_i^{*}_{i,j}.
    \]
Power Flow Model

- Basic electricity laws:
  - **Ohm’s Law**: For each edge \((i, j) \in \mathcal{E}\),
    \[ V_i - V_j = z_{i,j} I_{i,j}. \]
  - **Kirchhoff’s Current Law**: For each node \(i \in \mathcal{V}\),
    \[ \sum_{j \in \mathcal{V} : (i, j) \in \mathcal{E}} I_{i,j} = 0. \]
  - **Electric Power Formula**: For each edge \((i, j) \in \mathcal{E}\),
    \[ S_{i,j} = V_i I_{i,j}^*. \]

**Definition (Branch Flow Model)**

\[
S_{i,j} = z_{i,j} |I_{i,j}|^2 + \sum_{l : (j,l) \in \mathcal{E}} S_{j,l}, \quad \text{for all } j \in \mathcal{V},
\]
\[
V_i - V_j = z_{i,j} I_{i,j}, \quad \text{for all } (i, j) \in \mathcal{E},
\]
\[
S_{i,j} = V_i I_{i,j}^*, \quad \text{for all } (i, j) \in \mathcal{E}.
\]
Power Flow Model

Definition (Branch Flow Model with Angle Relaxation)

Let \( v_i = |V_i|^2 \) and \( \ell_{i,j} = |I_{i,j}|^2 \). Omit the phase angles:

\[
S_{i,j} = z_{i,j} \ell_{i,j} + \sum_{l:(j,l)\in\mathcal{E}} S_{j,l}, \quad \text{for all } j \in \mathcal{V},
\]

\[
v_i - v_j = 2 \text{Re}(z_{i,j}^* S_{i,j}) - |z_{i,j}|^2 \ell_{i,j}, \quad \text{for all } (i, j) \in \mathcal{E},
\]

\[
|S_{i,j}|^2 = v_i \ell_{i,j}, \quad \text{for all } (i, j) \in \mathcal{E}.
\]

- Angle relaxation reduces complex-valued variables, and hence is more tractable.
- Always possible to recover \((V_i, I_{i,j})\) from \((v_i, \ell_{i,j})\), when it is a tree network [Low et al.]
- Assume a tree (radial) distribution network
Control Variables & Operating Constraints

- Each user $k \in \mathcal{N}$ can control individual demand $s_k$

- Some have **discrete** (inelastic) power demands ($\mathcal{I} \subseteq \mathcal{N}$)
  - A discrete demand is either completely satisfied or dropped
  - E.g., equipment is either switched on with a fixed power consumption rate or completely off
  - $s_k = \bar{s}_k x_k$, where $x_k \in \{0, 1\}$

- Others have **continuous** (elastic) demands ($\mathcal{F} \subseteq \mathcal{N}$)
  - $\underline{s}_k \leq s_k \leq \bar{s}_k$
Control Variables & Operating Constraints

- Each user $k \in \mathcal{N}$ can control individual demand $s_k$
- Some have **discrete** (inelastic) power demands ($\mathcal{I} \subseteq \mathcal{N}$)
  - A discrete demand is either completely satisfied or dropped
  - E.g., equipment is either switched on with a fixed power consumption rate or completely off
  - $s_k = \bar{s}_k x_k$, where $x_k \in \{0, 1\}$
- Others have **continuous** (elastic) demands ($\mathcal{F} \subseteq \mathcal{N}$)
  - $s_k \leq s_k \leq \bar{s}_k$

Operating Constraints of Power Systems:
- **Power Capacity Constraints:** $|S_{i,j}| \leq \bar{S}_{i,j}$
- **Current Thermal Constraints:** $\ell_{i,j} \leq \bar{\ell}_{i,j}$
- **Voltage Constraints:** $\underline{v}_j \leq v_j \leq \bar{v}_j$
Optimal Power Flow Problem (OPF)

**Definition (Optimal Power Flow Problem)**

\[
\begin{align*}
\text{(OPF)} \quad & \max_{s_0, s, x, S, v, \ell} f(s_0, s) \\
\text{subject to} \quad & \ell_{i,j} = \frac{|S_{i,j}|^2}{v_i}, \quad \forall (i, j) \in \mathcal{E}, \\
& S_{i,j} = \sum_{k \in \mathcal{U}_j} s_k + \sum_{l: (j, l) \in \mathcal{E}} S_{j,l} + z_{i,j} \ell_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \\
& v_j = v_i + |z_{i,j}|^2 \ell_{i,j} - 2\text{Re}(z_{i,j}^* S_{i,j}), \quad \forall (i, j) \in \mathcal{E}, \\
& v_j \leq v_j \leq \overline{v}_j, \quad \forall j \in \mathcal{V}^+, \\
& |S_{i,j}| \leq \overline{S}_{i,j}, \quad |S_{j,i}| \leq \overline{S}_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \\
& \ell_{i,j} \leq \overline{\ell}_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \\
& s_k \leq s_k \leq \overline{s}_k, \quad \forall k \in \mathcal{F}, \\
& s_k = \overline{s}_k x_k, \quad x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}, \\
& v_j \in \mathbb{R}^+, \forall j \in \mathcal{V}^+, \ell_{i,j} \in \mathbb{R}^+, S_{i,j} \in \mathbb{C}, \quad \forall (i, j) \in \mathcal{E}.
\end{align*}
\]
Hardness of OPF

- **Non-Convex Constraints** involving complex-valued variables and parameters of AC electric power systems
  - E.g. $\ell_{i,j} = \frac{|S_{i,j}|^2}{v_i}$

- **Combinatorial Constraints** involving binary control decision variables for the operations of power systems
  - E.g. $s_k = \bar{s}_k x_k, \ x_k \in \{0, 1\}$

- Even without combinatoric constraints, checking feasibility of OPF (with voltage & capacity constraints) is NP-hard
  - Ref. [ Lehmann et al.] [Verma]

- Need to relax some constraints
  - Namely, considering a less restrictive optimization problem
Convex Relaxation of OPF

Definition (Convex Relaxed OPF)

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

subject to

\[\begin{align*}
\ell_{i,j} & \geq \frac{|S_{i,j}|^2}{v_i}, & \forall (i, j) \in \mathcal{E}, \\
S_{i,j} & = \sum_{k \in \mathcal{U}_j} s_k + \sum_{l:(j,l) \in \mathcal{E}} S_{j,l} + z_{i,j} \ell_{i,j}, & \forall (i, j) \in \mathcal{E}, \\
v_j & = v_i + |z_{i,j}|^2 \ell_{i,j} - 2\text{Re}(z_{i,j}^* S_{i,j}), & \forall (i, j) \in \mathcal{E}, \\
v_j & \leq v_j \leq \bar{v}_j, & \forall j \in \mathcal{V}^+, \\
|S_{i,j}| & \leq \bar{S}_{i,j}, \quad |S_{j,i}| \leq \bar{S}_{i,j}, & \forall (i, j) \in \mathcal{E}, \\
\ell_{i,j} & \leq \bar{\ell}_{i,j}, & \forall (i, j) \in \mathcal{E}, \\
s_k & \leq s_k \leq \bar{s}_k, & \forall k \in \mathcal{F}, \\
s_k & = \bar{s}_k x_k, \quad x_k \in \{0, 1\}, & \forall k \in \mathcal{I}, \\
v_j & \in \mathbb{R}^+, \forall j \in \mathcal{V}^+, \quad \ell_{i,j} \in \mathbb{R}^+, \quad S_{i,j} \in \mathbb{C}, & \forall (i, j) \in \mathcal{E}.
\end{align*}\]
Convex Relaxation of OPF

- **Second Order Cone Problem**: Easier convex problem with polynomial-time algorithms
- **Exactness**: Solution of cOPF $\Rightarrow$ Solution of OPF
  - Under certain sufficient conditions [Low et al.] [Huang et al.]
  - Conversion is polynomial-time
Convex Relaxation of OPF

- **Second Order Cone Problem**: Easier convex problem with polynomial-time algorithms
- **Exactness**: Solution of cOPF $\Rightarrow$ Solution of OPF
  - Under certain sufficient conditions [Low et al.] [Huang et al.]
  - Conversion is polynomial-time

**Assumptions**

**A1**: $z_e \geq 0, \forall e \in \mathcal{E}$, naturally holds in distribution networks

**A2**: $v_j < v_0 < \bar{v}_j, \forall j \in \mathcal{V}^+$

**C2**: Given a solution $s$, it satisfies

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

where $\mathcal{N}_j$ is the set of attached users within subtree from node $j$, and $\mathcal{E}_j$ is the set of edges of subtree node $j$
Theorem

Assuming A1, A2, C2, an optimal solution to cOPF, $F^* = (s_0^*, s^*, x^*, S^*, v^*, \ell^*)$ can be converted to an optimal solution to OPF, $\hat{F}^* = (\hat{s}_0^*, s^*, x^*, \hat{S}^*, \hat{v}^*, \hat{\ell}^*)$, in polynomial-time, by solving the following convex problem:

$$\left( \text{cOPF}[F^*] \right) \min_{s_0, S, v, \ell} \sum_{e \in \mathcal{E}} \ell_e$$

subject to constraints of cOPF

\[ s = s^* \]

\[ f(s_0, s^*) \geq f(s_0^*, s^*) \]
Convex Relaxation of OPF

Theorem

Assuming A1,A2,C2, an optimal solution to cOPF, \( F^* = (s_0^*, s^*, x^*, S^*, v^*, \ell^*) \) can be converted to an optimal solution to OPF, \( \hat{F}^* = (\hat{s}_0^*, s^*, x^*, \hat{S}^*, \hat{v}^*, \hat{\ell}^*) \), in polynomial-time, by solving the following convex problem:

\[
(\text{cOPF}[F^*]) \quad \min_{s_0, S, v, \ell} \sum_{e \in \mathcal{E}} \ell_e \\
\text{subject to constraints of cOPF} \\
s = s^* \\
f(s_0, s^*) \geq f(s_0^*, s^*)
\]

Question

How to solve cOPF with discrete demands?
Main Ideas

- Solve OPF by studying simplified problems
- **Complex-demand Knapsack Problem** (CKP)
  - Single-capacitated AC system without impedance
  - Demands are complex numbers
- **Knapsack Problem** (KP)
  - Classical computer science problem
  - Packing discrete items subject to capacity constraint
  - Demands are non-negative real numbers
What is Knapsack Problem?

My Hobby:
Embedding NP-Complete Problems in Restaurant Orders

Chotchkies Restaurant

Appetizers
- Mixed Fruit 2.15
- French Fries 2.75
- Side Salad 3.35
- Hot Wings 3.55
- Mozzarella Sticks 4.20
- Sampler Plate 5.80

Sandwiches
- Barbecue 6.55

We’d like exactly $15.05 worth of appetizers, please.

... Exactly? Uhh...

Here, these papers on the knapsack problem might help you out.

Listen, I have six other tables to get to—

As fast as possible, of course. Want something on traveling salesman?
Complex-demand Knapsack Problem (CKP)

Knapsack Problem $\Rightarrow$ Complex-demand Knapsack Problem
(1D) Knapsack Problem (KP)

- $N = \{1, \ldots, n\}$: a set of users (or items)
- $s^R_k$: positive real-valued demand of $k$-th user (e.g. weight)
- $u_k$: utility of $k$-th user when $s^R_k$ is satisfied (e.g. value)
- $C$: real-valued capacity
- $x_k$: decision variable of allocation
  - $x_k = 1$, if $k$-th user’s demand is satisfied
  - $x_k = 0$, otherwise

Definition (1DKP)

$$
\begin{align*}
\max_{x_k \in \{0,1\}} & \sum_{k \in N} x_k u_k \\
\text{subject to} & \\
\sum_{k \in N} s^R_k x_k & \leq C
\end{align*}
$$
(1D) Knapsack Problem (KP)

- $\mathcal{N} = \{1, \ldots, n\}$: a set of users (or items)
- $s_k^R$: positive real-valued demand of $k$-th user (e.g. weight)
- $u_k$: utility of $k$-th user when $s_k^R$ is satisfied (e.g. value)
- $C$: real-valued capacity
- $x_k$: decision variable of allocation
  - $x_k = 1$, if $k$-th user’s demand is satisfied
  - $x_k = 0$, otherwise

**Definition (1DKP)**

$$\max_{x_k \in \{0,1\}} \sum_{k \in \mathcal{N}} x_k u_k$$
subject to

$$\sum_{k \in \mathcal{N}} s_k^R x_k \leq C$$

- 1DKP is NP-Hard, but can be approximately solved efficiently
Complex-demand Knapsack Problem (CKP)

- $s_k$ is complex-valued demand of $k$-th user ($s_k = s_k^R + i s_k^I$)

\[
\text{minimize} \quad \sum_k x_k \in \{0, 1\} \\
\text{subject to} \quad \sum_k x_k \leq C
\]
Complex-demand Knapsack Problem (CKP)

- $s_k$ is complex-valued demand of $k$-th user ($s_k = s_k^R + is_k^I$)

Definition (CKP)

$$\max_{x_k \in \{0,1\}} \sum_{k \in \mathcal{N}} u_k x_k$$
subject to

$$\left| \sum_{k \in \mathcal{N}} s_k x_k \right| \leq C$$

Definition (1DKP)

$$\max_{x_k \in \{0,1\}} \sum_{k \in \mathcal{N}} u_k x_k$$
subject to

$$\sum_{k \in \mathcal{N}} s_k^R x_k \leq C$$
Approximation Algorithms

- NP-Hard Problems can’t be solved exactly
- But efficient approximation solutions exist
Approximation Algorithms

- NP-Hard Problems can’t be solved exactly
- But efficient approximation solutions exist
- Denote a solution by \((\hat{x}_k)_{k \in \mathcal{N}} \in \{0, 1\}^n\)
- Denote an optimal solution by \((x^*_k)_{k \in \mathcal{N}}\)
- Let \(u(x) \triangleq \sum_k u_k x_k\) be the objective value of \(x\)
Approximation Algorithms

- NP-Hard Problems can’t be solved exactly
- But efficient approximation solutions exist
- Denote a solution by $\left(\hat{x}_k\right)_{k \in \mathcal{N}} \in \{0, 1\}^n$
- Denote an optimal solution by $\left(x^*_k\right)_{k \in \mathcal{N}}$
- Let $u(x) \triangleq \sum_k u_k x_k$ be the objective value of $x$

Definition

Given $\alpha \in (0, 1]$ and $\beta \geq 1$, a bi-criteria $(\alpha, \beta)$-approximation to CKP is $\left(\hat{x}_k\right)_{k \in \mathcal{N}}$ satisfying:

$$u(\hat{x}) \geq \alpha \cdot u(x^*)$$

$$\left| \sum_{k \in \mathcal{N}} s_k \hat{x}_k \right| \leq \beta \cdot C,$$
Approximation Algorithms

- NP-Hard Problems can't be solved exactly
- But efficient approximation solutions exist
- Denote a solution by $(\hat{x}_k)_{k \in \mathcal{N}} \in \{0, 1\}^n$
- Denote an optimal solution by $(x^*_k)_{k \in \mathcal{N}}$
- Let $u(x) \triangleq \sum_k u_k x_k$ be the objective value of $x$

Definition

Given $\alpha \in (0, 1]$ and $\beta \geq 1$, a bi-criteria $(\alpha, \beta)$-approximation to CKP is $(\hat{x}_k)_{k \in \mathcal{N}}$ satisfying:

$$u(\hat{x}) \geq \alpha \cdot u(x^*)$$

$$\left\lvert \sum_{k \in \mathcal{N}} s_k \hat{x}_k \right\rvert \leq \beta \cdot C,$$

- **PTAS** ($\alpha = 1 - \epsilon$, $\beta = 1$): Running time is polynomial in $n$ for any fixed $\epsilon$
Approximation Algorithms

- NP-Hard Problems can’t be solved exactly
- But efficient approximation solutions exist
- Denote a solution by \((\hat{x}_k)_{k \in \mathcal{N}} \in \{0, 1\}^n\)
- Denote an optimal solution by \((x^*_k)_{k \in \mathcal{N}}\)
- Let \(u(x) \triangleq \sum_k u_k x_k\) be the objective value of \(x\)

**Definition**

Given \(\alpha \in (0, 1]\) and \(\beta \geq 1\), a bi-criteria \((\alpha, \beta)\)-approximation to CKP is \((\hat{x}_k)_{k \in \mathcal{N}}\) satisfying:

\[
\begin{align*}
  u(\hat{x}) &\geq \alpha \cdot u(x^*) \\
  \left| \sum_{k \in \mathcal{N}} s_k \hat{x}_k \right| &\leq \beta \cdot C,
\end{align*}
\]

- **PTAS** \((\alpha = 1 - \epsilon, \beta = 1)\): Running time is polynomial in \(n\) for any fixed \(\epsilon\)
- **FPTAS**: PTAS with running time also polynomial in \(1/\epsilon\)
Results for CKP

- Let $\phi$ be the maximum angle between any two demands.
- The (in)approximability is dependent on $\phi$.
- CKP is rotational invariant
  - When all demands are rotated by the same angle.
Results for CKP

CKP\([0, \frac{\pi}{2}]\):

- **PTAS** \((1 - \epsilon, 1)\)-approx
- **No FPTAS** unless \(P=NP\)

\[\epsilon = \frac{1}{\text{poly}(n)}\] is a bi-criteria FPTAS \((1, 1 + \epsilon)\)-approx

\[\epsilon = \frac{1}{\text{super-pol}(n)}\] is no \((\alpha, \beta)\)-approximation unless \(P=NP\)
Results for CKP

CKP\([0, \frac{\pi}{2}]\):
- PTAS \((1 - \epsilon, 1)\)-approx
- No FPTAS unless P=NP

CKP\([\frac{\pi}{2}, \pi - \epsilon]\):
- \(\epsilon = 1/\text{poly}(n)\)
- Bi-criteria FPTAS \((1, 1+\epsilon)\)-approx
- No \((\alpha, 1)\)-approximation unless P=NP
Results for CKP

CKP\([0, \frac{\pi}{2}]\):
- **PTAS** \((1 - \epsilon, 1)\)-approx
- **No FPTAS** unless \(P=NP\)

CKP\([\frac{\pi}{2}, \pi - \epsilon]\):
- \(\epsilon = 1/\text{poly}(n)\)
- **Bi-criteria FPTAS** \((1, 1 + \epsilon)\)-approx
- **No \((\alpha, 1)\)-approximation** unless \(P=NP\)

CKP\([\frac{\pi}{2}, \pi - \epsilon]\):
- \(\epsilon = 1/\text{super-pol}(n)\)
- **No \((\alpha, \beta)\)-approximation** unless \(P=NP\)
Results for CKP

Other results: Greedy Algorithm

- Running time $O(n \log n)$
- Constant factor approximation algorithm ($\alpha = \frac{1}{2} \cos \frac{\phi}{2}$)
PTAS for CKP $[0, \frac{\pi}{2}]$

- Assume $\phi \in [0, \frac{\pi}{2}]$ (i.e. bounded power factor $\geq 0.75$)
PTAS for CKP$[0, \frac{\pi}{2}]$

- Assume $\phi \in [0, \frac{\pi}{2}]$ (i.e. bounded power factor $\geq 0.75$)
- Basic Ideas:
  1. Fix partial solution by guessing
  2. Solve convex relaxation with fractional solution
  3. Round the obtained fractional solution to integral solution
  4. Enumerate all possible partial guessing and find the best solution
PTAS for $\text{CKP}[0, \frac{\pi}{2}]$

- Recall integer quadratic formulation:

$$\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k$$

subject to:

$$\left( \sum_{k \in \mathcal{N}} s_{k}^{R} \cdot x_{k} \right)^2 + \left( \sum_{k \in \mathcal{N}} s_{k}^{I} \cdot x_{k} \right)^2 \leq C^2,$$

$$x_k \in \{0, 1\}, \ \forall k \in \mathcal{N}$$
PTAS for CKP\([0, \frac{\pi}{2}]\)

- Recall integer quadratic formulation:

\[
\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k \\
\text{subject to } \left( \sum_{k \in \mathcal{N}} s_k^R \cdot x_k \right)^2 + \left( \sum_{k \in \mathcal{N}} s_k^I \cdot x_k \right)^2 \leq C^2, \\
x_k \in \{0, 1\}, \ \forall k \in \mathcal{N}
\]

**Definition (Quadratic Relaxation)**

\[
\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k \\
\text{subject to } \left( \sum_{k \in \mathcal{N}} s_k^R \cdot x_k \right)^2 + \left( \sum_{k \in \mathcal{N}} s_k^I \cdot x_k \right)^2 \leq C^2, \\
x_k \in [0, 1], \ \forall k \in \mathcal{N}
\]

- The relaxation is convex because \(s_k^R, s_k^I \geq 0\) for all \(k\).
PTAS for CKP\([0, \frac{\pi}{2}]\)

- Fix some subsets \(I_1, I_0 \subseteq \mathcal{N}\)
- Quadratic Relaxation with *partial guessing* of variables \(\{x_k\}\):

\[
\text{Definition (RCKP}[I_1, I_0])
\]

\[
\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k
\]

subject to \((\sum_{k \in \mathcal{N}} s_k^R \cdot x_k)^2 + (\sum_{k \in \mathcal{N}} s_k^I \cdot x_k)^2 \leq C^2,\)

\(x_k \in [0, 1], \ \forall k \in \mathcal{N}(I_0 \cup I_1)\)

\(x_k = 0, \ \forall k \in I_0\)

\(x_k = 1, \ \forall k \in I_1\)
PTAS for CKP\([0, \frac{\pi}{2}]\)

- Fix some subsets \(I_1, I_0 \subseteq \mathcal{N}\)
- Quadratic Relaxation with \textit{partial guessing} of variables \(\{x_k\}\):

\[
\begin{align*}
\text{Definition (RCKP}^{I_1, I_0}]\) & \\
\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k & \\
\text{subject to } \left( \sum_{k \in \mathcal{N}} s^R_k \cdot x_k \right)^2 + \left( \sum_{k \in \mathcal{N}} s^I_k \cdot x_k \right)^2 \leq C^2, & \\
& x_k \in [0, 1], \quad \forall k \in \mathcal{N} \setminus (I_0 \cup I_1) \\
x_k = 0, \quad \forall k \in I_0 & \\
x_k = 1, \quad \forall k \in I_1
\end{align*}
\]

- Carefully selection of \(I_1, I_0\):
  - \(|I_1| \leq \frac{4}{\epsilon} \) (depending on \(\epsilon\))
  - \(I_0 = \{k \in \mathcal{I}\setminus I_1 \mid u_k > \min_{k' \in I_1} u_{k'}\}\)
- \textit{Intuition:} \(I_0, I_1\) are some users with large utilities
PTAS for $\text{CKP}[0, \frac{\pi}{2}]$

- Let $x'$ be an optimal solution to $\text{RCKP}[I_1, I_0]$

**Definition (LP[$x', I_1 \cup I_0$])**

$$\max_{x_k} \sum_{k \in \mathcal{N}} u_k x_k$$

subject to

$$\sum_{k \in \mathcal{N}} s_k^R \cdot x_k \leq \sum_{k \in \mathcal{N}} s_k^R \cdot x'_k,$$

$$\sum_{k \in \mathcal{N}} s_k^I \cdot x_k \leq \sum_{k \in \mathcal{N}} s_k^I \cdot x'_k,$$

$x_k \in [0, 1], \ \forall k \in \mathcal{N} \setminus (I_0 \cup I_1)$

$x_k = x'_k, \ \forall k \in I_1 \cup I_0$
PTAS for CKP[0, $\frac{\pi}{2}$]

- Let $x'$ be an optimal solution to $\text{RCKP}[I_1, I_0]

**Definition (LP[$x', I_1 \cup I_0$])**

\[
\max_{x_k} \sum_{k \in N} u_k x_k
\]

subject to

\[
\sum_{k \in N} s_k^R \cdot x_k \leq \sum_{k \in N} s_k^R \cdot x'_k,
\]

\[
\sum_{k \in N} s_k^I \cdot x_k \leq \sum_{k \in N} s_k^I \cdot x'_k,
\]

$x_k \in [0, 1], \ \forall k \in N \setminus (I_0 \cup I_1)$

$x_k = x'_k, \ \forall k \in I_1 \cup I_0$

- Because of basic solution of LP, at most 2 coordinates in optimal basic solution to LP[$x', I_1 \cup I_0$] are fractional
  - Basic solutions are *vertices* in the polytope of feasible solution set
  - Rounding fractional components down to integral components has limited deviation from optimal
PTAS for CKP[0, $\pi/2$]

Algorithm PTAS-CKP

- Pick a guess of partial solution by choosing $I_1, I_0$
  - Each guess sets variables $x_k$ in $I_1$ be 1 and $I_0$ be 0
  - Solve optimal solution of $\text{RCKP}[I_1, I_0]$, called $x'$
  - Solve basic optimal solution of $\text{LP}[x', I_1 \cup I_0]$, called $x''$
  - Round fractional components of $x''$ down to 0:
    $\hat{x}_k = \lfloor x''_k \rfloor$, $\forall k \in \mathcal{N}$

- Return the solution $\hat{x}$ with the highest utility among all guesses

Theorem

For any $\epsilon > 0$, PTAS-CKP obtains $(1 - \epsilon, 1)$-approximation in polynomial time.
PTAS for CKP\([0, \frac{\pi}{2}]\)

**Algorithm PTAS-CKP**

- Pick a guess of partial solution by choosing \(I_1, I_0\)
  - Each guess sets variables \(x_k\) in \(I_1\) be 1 and \(I_0\) be 0
  - Solve optimal solution of \(\text{rCKP}[I_1, I_0]\), called \(x'\)
  - Solve basic optimal solution of \(\text{LP}[x', I_1 \cup I_0]\), called \(x''\)
  - Round fractional components of \(x''\) down to 0:
    \[
    \hat{x}_k = \lfloor x''_k \rfloor, \quad \forall k \in \mathcal{N}
    \]
- Return the solution \(\hat{x}\) with the highest utility among all guesses

**Theorem**

*For any \(\epsilon > 0\), PTAS-CKP obtains \((1 - \epsilon, 1)\)-approximation in polynomial time.*
**Convex Relaxation of OPF**

**Definition (Convex Relaxed OPF)**

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

subject to

1. \[ l_{i,j} \geq \frac{|S_{i,j}|^2}{v_i}, \quad \forall (i, j) \in \mathcal{E}, \]
2. \[ S_{i,j} = \sum_{k \in \mathcal{U}_j} s_k + \sum_{l: (j, l) \in \mathcal{E}} S_{j,l} + z_{i,j} \ell_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \]
3. \[ v_j = v_i + |z_{i,j}|^2 \ell_{i,j} - 2\text{Re}(z_{i,j}^* S_{i,j}), \quad \forall (i, j) \in \mathcal{E}, \]
4. \[ v_j \leq v_j \leq \bar{v}_j, \quad \forall j \in \mathcal{V}^+, \]
5. \[ |S_{i,j}| \leq \bar{S}_{i,j}, \quad |S_{j,i}| \leq \bar{S}_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \]
6. \[ l_{i,j} \leq \bar{l}_{i,j}, \quad \forall (i, j) \in \mathcal{E}, \]
7. \[ s_k \leq s_k \leq \bar{s}_k, \quad \forall k \in \mathcal{F}, \]
8. \[ s_k = \bar{s}_k x_k, \quad x_k \in \{0, 1\}, \quad \forall k \in \mathcal{I}, \]
9. \[ v_j \in \mathbb{R}^+, \forall j \in \mathcal{V}^+, \quad l_{i,j} \in \mathbb{R}^+, \quad S_{i,j} \in \mathbb{C}, \quad \forall (i, j) \in \mathcal{E}. \]
Hardness Results for OPF

- **OPF**\_\_V: OPF with voltage constraints only \((v_j \leq v_j \leq \overline{v}_j)\)

**Theorem**

Unless P=NP, there is no \((\alpha, \beta)\)-approximation for OPF\_\_V (even when \(|E| = 1\)), for any \(\alpha\) and \(\beta\) that have polynomial number of bits in \(n\).
Hardness Results for OPF

- **OPF\(_V\):** OPF with voltage constraints only \((v_j \leq v_j \leq \bar{v}_j)\)

**Theorem**

*Unless \(P=NP\), there is no \((\alpha, \beta)\)-approximation for OPF\(_V\) (even when \(|E| = 1\)), for any \(\alpha\) and \(\beta\) that have polynomial number of bits in \(n\).*

**Remark**

To obtain approximation algorithms, one has to relax some constraints
Hardness Results for OPF

- **OPF\(_C\):** OPF with capacity constraints only (\(|S_{i,j}| \leq \overline{S}_{i,j}\))

**Theorem**

Unless P=NP, there exists no \((\alpha, \beta)\)-approximation for \(\text{OPF}_C\) in general networks, even in purely resistive electric networks (i.e. \(\text{Im}(z_{i,j}) = 0\) for all \((i, j) \in \mathcal{E}\) and \(\text{Im}(s_k) = 0\) for all \(k \in \mathcal{N}\)).
Hardness Results for OPF

- $\text{OPF}_C$: OPF with capacity constraints only ($|S_{i,j}| \leq \overline{S}_{i,j}$)

**Theorem**

Unless $P=NP$, there exists no $(\alpha, \beta)$-approximation for $\text{OPF}_C$ in general networks, even in purely resistive electric networks (i.e. $\text{Im}(z_{i,j}) = 0$ for all $(i, j) \in \mathcal{E}$ and $\text{Im}(s_k) = 0$ for all $k \in \mathcal{N}$).

**Remark**

To obtain approximation algorithms, one has to consider acyclic networks (i.e. trees)
Assumptions on OPF

- OPF with discrete demands is hard to solve
- Some assumptions are required to facilitate the solutions

Assumptions

A1: \( z_e \geq 0, \forall e \in \mathcal{E} \), naturally holds in distribution networks

A2: \( v_j < v_0 < \overline{v}_j, \forall j \in \mathcal{V}^+ \)

A3: \( \text{Re}(z^*_e \overline{s}_k) \geq 0, \forall k \in \mathcal{I}, e \in \mathcal{E} \)

- Namely, the phase angle difference between any \( z_e \) and \( s_k \) for \( k \in \mathcal{I} \) is at most \( \frac{\pi}{2} \)
- This assumption holds, if discrete demands do not have large negative reactive power

A4: \( |\angle \overline{s}_k - \angle \overline{s}_{k'}| \leq \frac{\pi}{2} \) for any \( k, k' \in \mathcal{I} \)

- Namely, discrete demands have similar power factors
- It can also be restated as \( \text{Re}(\overline{s}_k^* \overline{s}_{k'}) \geq 0 \)
Basic Ideas of PTAS for OPF

Non-convex OPF with discrete demands

Exact Convex Relaxation

Convex OPF with discrete demands

Rotation

(Rotated) Convex OPF with discrete demands

Relaxation with Partial Guessing

Approximation Solution

(Rotated) Convex OPF with partial continuous demands

(Rotated) Convex OPF with partial guessed discrete demands

(Rotated) Convex OPF with partial guessed discrete demands

(Rotated) Convex OPF with partial guessed discrete demands

Rounding
Rotational Invariance

- If complex-valued parameters $z_e$ and $s_k$ are rotated by the same angle (say $\phi$)
- And objective func $f(s_0, s)$ is counter-rotated by $\phi$ in $s_0$
If complex-valued parameters $z_e$ and $s_k$ are rotated by the same angle (say $\phi$)

And objective func $f(s_0, s)$ is counter-rotated by $\phi$ in $s_0$

Then there is a bijection between the rotated OPF and original OPF

$$(z_e, s_k, f(s_0, s)) \leftrightarrow (z_e e^{i\phi}, s_k e^{i\phi}, f(s_0 e^{-i\phi}, s))$$

Therefore, assume all $s_k, z_e$ are in the first quadrant
PTAS for OPF

- Assume constant-sized network ($|\mathcal{V}^+| = |\mathcal{E}| = m$)
- But number of users ($|\mathcal{N}| = n$) is a scalable parameter
- Define a variant of cOPF with partially guessing:

**Definition (P1[\mathcal{I}_0, \mathcal{I}_1])**

$$\max_{s_0, s, x, S, v, \ell} f(s_0, s)$$

subject to Constraints of cOPF

$$s_k = \overline{s}_k x_k, \forall k \in \mathcal{I},$$

$$x_k = 1, \forall k \in \mathcal{I}_1, \quad x_k = 0, \forall k \in \mathcal{I}_0,$$

$$x_k \in [0, 1], \forall k \in \mathcal{I}\setminus(\mathcal{I}_0 \cup \mathcal{I}_1)$$

- Let optimal solution of P1 be $F' = (s'_0, s', x', S', v', \ell')$
PTAS for OPF

- Define $\bar{f}_k \triangleq f_k(1)$ for $k \in \mathcal{I}$, and define LP as:

**Definition (P2[$F', I_0 \cup I_1$])**

$$
\max_{x_k \in [0,1], k \in \mathcal{I}'} \sum_{k \in \mathcal{I}} \bar{f}_k x_k
$$

subject to

$$
0 \leq \sum_{k \in \mathcal{N}} \Re \left( \sum_{(h,l) \in \mathcal{P}_k \cap \mathcal{P}_j} z_{h,l}^* s_k \right)
\leq \sum_{k \in \mathcal{N}} \Re \left( \sum_{(h,l) \in \mathcal{P}_k \cap \mathcal{P}_j} z_{h,l}' s_k' \right), \forall j \in \mathcal{V}^+
$$

$$
\sum_{k \in \mathcal{N}_j} \Re (s_k) \leq \sum_{k \in \mathcal{N}_j} \Re (s_k'), \forall j \in \mathcal{V}^+
$$

$$
\sum_{k \in \mathcal{N}_j} \Im (s_k) \leq \sum_{k \in \mathcal{N}_j} \Im (s_k'), \forall j \in \mathcal{V}^+
$$

$$
s_k = \bar{s}_k x_k, \forall k \in \mathcal{N} \setminus (I_0 \cup I_1)
$$

$$
s_k = s_k', \forall I_0 \cup I_1$$
PTAS for OPF

**Algorithm PTAS-cOPF**

- Guess partial solution by $I_1, I_0$, where $|I_1| \leq \frac{4m}{\epsilon}$
  - Each guess sets variables $x_k$ in $I_1$ be 1 and $I_0$ be 0
  - Solve optimal solution of $P1[I_1, I_0]$, called $F'$
  - Solve optimal solution of $P2[F', I_0 \cup I_1]$, called $x''$
  - Round fractional components of $x''$ down to 0:
    \[
    \hat{x}_k = \lfloor x''_k \rfloor, \quad \forall k \in \mathcal{N}
    \]
- Return the solution $\hat{x}$ with the highest utility among all guesses

**Theorem**

Assuming $A1,A2,A3,A4$, for any $\epsilon > 0$, PTAS-cOPF obtains $(1 - \epsilon, 1)$-approximation in polynomial time for constant-sized tree electric networks.
## Algorithm PTAS-cOPF

- Guess partial solution by $I_1, I_0$, where $|I_1| \leq \frac{4m}{\epsilon}$
  - Each guess sets variables $x_k$ in $I_1$ be 1 and $I_0$ be 0
  - Solve optimal solution of $P1[I_1, I_0]$, called $F'$
  - Solve optimal solution of $P2[F', I_0 \cup I_1]$, called $x''$
  - Round fractional components of $x''$ down to 0:
    \[ \hat{x}_k = \lfloor x''_k \rfloor, \quad \forall k \in \mathcal{N} \]
- Return the solution $\hat{x}$ with the highest utility among all guesses

## Theorem

**Assuming A1,A2,A3,A4, for any $\epsilon > 0$, PTAS-cOPF obtains $(1 - \epsilon, 1)$-approximation in polynomial time for constant-sized tree electric networks.**
Simulation Settings

- Compare approx algorithm against the optimal solutions
- Use numerical solver (Gurobi) to obtain optimal solution

Figure: Test electric network from [Singh, Baran]
Simulation Settings

- **User Types:**
  1. *Residential* (R): Users have small power demands ranging from 500VA to 5KVA
  2. *Industrial* (I): Users have big demands ranging from 300KVA to 1MVA with non-negative reactive power
  3. *Mixed* (M): Users consist of a mix of industrial and residential users, with less than 20% industrial users

- **Cost-Demand Correlation:**
  1. *Correlated Setting* (C): The objective of each user is a function of demand:

\[
f_k(|s_k|) = a \cdot |s_k|^2 + b \cdot |s_k| + c
\]

  2. *Uncorrelated Setting* (U): Coefficients in \( f_k \) of each user are generated randomly according to \(|s_{\text{max}}(k)|\)

    - For industrial user, \(|s_{\text{max}}(k)| = 1 \text{MVA}\), otherwise \(|s_{\text{max}}(k)| = 5 \text{KVA}\)
Simulation Results

![Graphs showing simulation results]

- **CR (Crack Detection Rate)**: The x-axis represents the number of users, and the y-axis shows the objective value. The graphs illustrate the performance of PTAS, Frac. (Lower Bound), and Numerical (Gurobi) methods.

- **CM (Categorization Accuracy)**: Similar to CR, the graphs plot objective values against the number of users.

- **UR (User Recognition Rate)**: The graphs for UR are structured similarly to CR and CM.

- **UM (User Match Quality)**: The objective values are plotted against the number of users, similar to the other graphs.

The graphs highlight the performance trends and comparisons among the different methods as the number of users increases.
Running Time

![Graph showing Running Time vs. Number of Users for different scenarios: CR, CM, UR, UM. The graphs display data for PTAS and Numerical (Gurobi) methods, with error bars indicating variability.]
Summary

- Complex-demand Knapsack Problem (CKP):
  - Hardness results
  - PTAS algorithm

- Optimal Power Flow with discrete demands (OPF):
  - Hardness results
  - PTAS algorithm
Summary

- Complex-demand Knapsack Problem (CKP):
  - Hardness results
  - PTAS algorithm

- Optimal Power Flow with discrete demands (OPF):
  - Hardness results
  - PTAS algorithm

- Other results:
  - Bi-criteria FPTAS
  - Scheduling problem
  - Scalable-sized networks
  - Truthful mechanisms
Related Papers

- Khonji, Karapetyan, Chau, Elbassioni. “Complex-demand Scheduling Problem with Application in Smart Grid”, Theoretical Computer Science, 2018