Efficient Algorithms for Renewable Energy Allocation to Delay Tolerant Consumers

Michael J. Neely, Arash Saber Tehrani, Alexandros G. Dimakis
University of Southern California

*Paper to appear at: 1st IEEE International Conf. on Smart Grid Communications, 2010
PDF on Stochastic Network Optimization Homepage: http://ee.usc.edu/stochastic-nets/
*Sponsored in part by the NSF Career CCF-0747525
Renewable sources of energy can have **variable** and **unpredictable** supplies $s(t)$.

We can integrate renewable sources more easily if consumers tolerate service within some **maximum allowable delay** $D_{\text{max}}$.

Might sometimes need to purchase energy from non-renewable source to meet the deadlines, and **purchase price can be highly variable**.
Example Data:  (Top Row) Spot Market Price
(Bottom Row) Energy Production in a California Wind Turbine
Talk Outline:

• **First Problem**: Minimize time average cost of purchasing non-renewable energy (i.i.d. case)

• **Second Problem**: Joint pricing of customers and purchasing of non-renewables (i.i.d. case).

• Generalize to *arbitrary sample paths* using “Universal Scheduling Theory” of Lyapunov Optimization.

• Simulation results using CAISO spot market prices \( \gamma(t) \) and 10-minute energy production \( s(t) \) from Western Wind resources Dataset (from National Renewable Energy Lab).
Problem 1: Minimize Average Cost of Non-Renewable Purchases

- Slotted Time: \( t = \{0, 1, 2, \ldots\} \)
- \( a(t) \) = energy requests on slot \( t \) (serve with max delay \( D_{\text{max}} \)).
- \( s(t) \) = renewable energy supply on slot \( t \). (“use-it-or-lose-it”)
- \( x(t) \) = amount non-renewable energy purchased on slot \( t \).
- \( \gamma(t) \) = $/unit energy price of non-renewables on slot \( t \).
- \( Q(t) \) = Energy request queue

\[
Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t), \quad \text{cost}(t) = x(t)\gamma(t)
\]
Problem 1: Minimize Average Cost of Non-Renewable Purchases

\[ Q(t+1) = \max\{Q(t) - s(t) - x(t), 0\} + a(t) \quad \text{, cost}(t) = x(t)\gamma(t) \]

Assumptions:

• For all slots \( t \) we have:
  
  \[ 0 \leq a(t) \leq a_{\text{max}} \quad , \quad 0 \leq s(t) \leq s_{\text{max}} \quad , \quad 0 \leq \gamma(t) \leq \gamma_{\text{max}} \quad , \quad 0 \leq x(t) \leq x_{\text{max}} \]

• \( x_{\text{max}} \) units of energy always available for purchase from non-renewable (but at variable price \( \gamma(t) \)).

• \( a_{\text{max}} \leq x_{\text{max}} \) (possible to meet all demands in 1 slot at high cost)

• \((a(t), s(t), \gamma(t))\) vector is i.i.d. over slots with unknown distribution
Problem 1: Minimize Average Cost of Non-Renewable Purchases

\[ Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \], \quad \text{cost}(t) = x(t)\gamma(t) \]

Possible formulation via Dynamic Programming (DP):

“Minimize average cost subject to max-delay \(D_{\text{max}}\).”

- This can be written as a DP, but requires distribution knowledge.
- Recent work on delay tolerant electricity consumers using DP is: [Papavasiliou and Oren, 2010]

We will not use DP. We will take a different approach...
Problem 1: Minimize Average Cost of Non-Renewable Purchases

\[ Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) , \quad \text{cost}(t) = x(t)\gamma(t) \]

Relaxed Formulation via Lyapunov Optimization for Queue Networks:

Minimize: \( \mathbb{E}\{\text{cost}\} \) (time average)
Subject to: (1) \( \mathbb{E}\{Q\} < \infty \) (a “queue stability” constraint)
(2) \( 0 \leq x(t) \leq x_{\max} \) for all \( t \)

• Define \( \text{cost}^* = \min \text{cost subject to stability} \)
• By definition: \( \text{cost}^* \leq \text{cost delivered by any other alg} \) (including DP)
• We will get within \( O(\delta) \) of \( \text{cost}^* \), with worst-case delay of \( 1/\delta \).

![Graph showing Avg. Cost vs. Worst Case Delay with our performance, optimal DP, and cost* compared to \( O(\delta) \).]
Advantages of Lyapunov Optimization for Queueing Networks:

• No knowledge of distribution information is required.
• Explicit $[O(\delta), O(1/\delta)]$ performance guarantees.
• Robust to changes in statistics, arbitrary correlations, non-ergodic, arbitrary sample paths (as we will show in this work).
• Worst case delay bounds (as we will show in this work).
• No curse of dimensionality: Implementation is just as easy in extended formulations with many dimensions:

General Lyapunov Optimization Problem: [Georgiadis, Neely, Tassiulas, F&T 2006]

Minimize: $\mathbb{E}\{y\}$
Subject to: 
(1) $\mathbb{E}\{x_i\} \leq 0$ for all $i$ in $\{1, \ldots, N\}$
(2) Queue $k$ is stable for all $k$ in $\{1, \ldots, K\}$
(3) Control action on slot $t$ in $ActionSpace(t)$
   (for all $t$ in $\{0, 1, 2, \ldots\}$)
Virtual Queue for Worst-Case Delay Guarantee (fix $\varepsilon > 0$):

<table>
<thead>
<tr>
<th>a(t)</th>
<th>Q(t)</th>
<th>s(t) + x(t)</th>
<th>Actual Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z(t)</td>
<td>s(t) + x(t)</td>
<td>Virtual Queue</td>
</tr>
</tbody>
</table>

$\varepsilon 1\{Q(t) > 0\}$

$Z(t+1) = \max[Z(t) - s(t) - x(t) + \varepsilon 1\{Q(t) > 0\}, 0]$

Theorem: Any algorithm with bounded queues $Q(t) \leq Q_{\text{max}}$, $Z(t) \leq Z_{\text{max}}$ for all $t$ yields worst-case delay of:

$$D_{\text{max}} = \left\lceil \frac{Q_{\text{max}} + Z_{\text{max}}}{\varepsilon} \right\rceil \text{ slots}$$

Proof Sketch: Suppose not. Consider slot $t$, $a(t)$:

Then: $\sum_{\tau=t}^{t+D_{\text{max}}} [s(\tau) + x(\tau)] \leq Q_{\text{max}}$

Implies: $Z(t+D_{\text{max}}) > Z_{\text{max}}$ (contradiction)
Stabilize $Z(t)$ and $Q(t)$ while minimizing average cost $\text{cost}(t)$:

Lyapunov Function: $L(t) = Z(t)^2 + Q(t)^2$

Lyapunov Drift: $\Delta(t) = E\{L(t+1) - L(t) | Z(t), Q(t)\}$

Take actions to greedily minimize “Drift-Plus-Weighted-Penalty”:

Minimize: $\Delta(t) + V\gamma(t)x(t)$

where $V$ is a positive constant that affects the $[O(1/V), O(V)]$ Cost-delay tradeoff.

(using $V=1/\delta$ recovers the $[O(\delta), O(1/\delta)]$ tradeoff.)
**Resulting Algorithm:** Every slot $t$, observe $(Z(t), Q(t), γ(t))$. Then:

- Choose $x(t) = \begin{cases} 0, & \text{if } Q(t) + Z(t) \leq Vγ(t) \\ x_{\max}, & \text{if } Q(t) + Z(t) > Vγ(t) \end{cases}$

- Update virtual queues $Q(t)$ and $Z(t)$ according to their equations

**Define:** $Q_{\max} = Vγ_{\max} + a_{\max}$, $Z_{\max} = Vγ_{\max} + ε$

**Theorem:** Under the above algorithm:
(a) $Q(t) \leq Q_{\max}$, $Z(t) \leq Z_{\max}$ for all $t$.
(b) Delay $\leq (Q_{\max} + Z_{\max})/ε = O(V)$

Further, if $(s(t), a(t), γ(t))$ i.i.d. over slots, and if $ε \leq \max[E\{a(t)\}, E\{s(t)\}]$ Then:

$$E\{\text{cost}\} \leq \text{cost}^* + B/V$$

[where $B = (s_{\max} + x_{\max})^2 + a_{\max}^2 + ε^2$]
Problem 2: Joint Pricing and Energy Allocation

Same system model, with following extensions:
• a(t) = arrivals = Random function of pricing decision p(t)
• h(t) = additional “demand state” (e.g. “HIGH, MED, LOW”)
• E{a(t) | p(t), h(t), γ(t)} = F(p(t), h(t), γ(t)) = Demand Function

Example:
Problem 2: Joint Pricing and Energy Allocation

Same system model, with following extensions:
• $a(t) = \text{arrivals} = \text{Random function of pricing decision } p(t)$
• $h(t) = \text{additional “demand state” (e.g. “HIGH, MED, LOW”)}$
• $E\{a(t) \mid p(t), h(t), \gamma(t)\} = F(p(t), h(t), \gamma(t)) = \text{Demand Function}$

New Problem:
• $\text{Profit}(t) = a(t)p(t) - x(t)\gamma(t)$

• Maximize Time Average Profit!

• $\text{Profit}^* = \text{Optimal Time Avg. Profit Subject to Stability}$
**Problem 2: Joint Pricing and Energy Allocation**

\[ E\{a(t)\} = F(p(t), h(t), \gamma(t)) \]

\[ \Delta(t) - V E\{\text{Profit}(t) | Z(t), Q(t)\} = \Delta(t) - V E\{a(t)p(t) - x(t)\gamma(t) | Z(t), Q(t)\} \]

**Drift-Plus-Penalty for New Problem:**

\[ \Delta(t) - V E\{\text{Profit}(t) | Z(t), Q(t)\} = \Delta(t) - V E\{a(t)p(t) - x(t)\gamma(t) | Z(t), Q(t)\} \]

**Resulting Algorithm:**

Every slot \( t \), observe \((h(t), Z(t), Q(t), \gamma(t))\). Then:

- (Pricing) Choose \( p(t) \) in \([0, p_{\max}]\) to solve:

  \[
  \text{Maximize: } F(p(t), h(t), \gamma(t))(Vp(t) - Q(t)) \\
  \text{Subject to: } 0 \leq p(t) \leq p_{\max}
  \]

- (Purchasing) Choose \( x(t) \) same as before.
- Update queues \( Q(t), Z(t) \) same as before.
**Problem 2: Joint Pricing and Energy Allocation**

\[ E\{a(t)\} = F(p(t), h(t), \gamma(t)) \]

Drift-Plus-Penalty for New Problem:
\[ \Delta(t) - VE\{Profit(t)|Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t)|Z(t),Q(t)\} \]

Resulting Algorithm:
Every slot \( t \), observe \( (h(t), Z(t), Q(t), \gamma(t)) \). Then:
• (Pricing) Choose \( p(t) \) in \([0, p_{max}]\) to solve:

\[
\text{Maximize: } F(p(t), h(t), \gamma(t))(Vp(t) - Q(t)) \\
\text{Subject to: } 0 \leq p(t) \leq p_{max}
\]

• (Purchasing) Choose \( x(t) \) same as before.
• Update queues \( Q(t), Z(t) \) same as before.

*If \( F(p, h, \gamma) = \beta(h)G(p, \gamma) \), don’t need to know demand state \( h(t) \)!
Problem 2: Joint Pricing and Energy Allocation

Drift-Plus-Penalty for New Problem:
\[ \Delta(t) - \text{VE}\{\text{Profit}(t) | Z(t), Q(t)\} = \Delta(t) - \text{VE}\{a(t)p(t) - x(t)\gamma(t) | Z(t), Q(t)\} \]

Resulting Algorithm:
Every slot \( t \), observe \((h(t), Z(t), Q(t), \gamma(t))\). Then:
• (Pricing) Choose \( p(t) \) in \([0, p_{\text{max}}]\) to solve:
  
  \[
  \text{Maximize: } \beta(h(t))G(p(t), \gamma(t))(Vp(t) - Q(t)) \\
  \text{Subject to: } 0 \leq p(t) \leq p_{\text{max}}
  \]

• (Purchasing) Choose \( x(t) \) same as before.
• Update queues \( Q(t), Z(t) \) same as before.

*If \( F(p, h, \gamma) = \beta(h)G(p, \gamma) \), don’t need to know demand state \( h(t) \)!
Theorem: Under the joint pricing and energy allocation algorithm:

(a) Worst case queue bounds $Q_{\text{max}}, Z_{\text{max}}$ same as before.

(b) Worst case delay bound $D_{\text{max}}$ same as before, i.e., $O(V)$.

(c) If $(s(t), \gamma(t), h(t))$ i.i.d. over slots, and $\varepsilon \leq E\{s(t)\}$, then:

$$E\{\text{profit}\} \geq \text{profit}^* - O(1/V)$$

Consider the first problem again ($x(t) = \text{only decision variable}$): Suppose $(s(t), \gamma(t), a(t))$ have \textit{arbitrary sample path!} (assume they are still bounded: $[0, s_{\text{max}}], [0, \gamma_{\text{max}}], [0, a_{\text{max}}]$.)

Universal Scheduling Theorem:
(a) Worst case queue bounds $Q_{\text{max}}, Z_{\text{max}}$ same as before.
(b) Worst case delay bound $D_{\text{max}}$ same as before, i.e., $O(V)$.
(c) For any integers $T>0$, $R>0$:

$$
\frac{1}{RT} \sum_{t=0}^{RT-1} x(t)\gamma(t) \leq \frac{1}{R} \sum_{r=0}^{R-1} C_r^* + BT/V
$$

“Genie-Aided” T-Slot Lookahead Cost!
For every $R>0$, $T>0$:

$$\frac{1}{RT} \sum_{t=0}^{RT-1} x(t)\gamma(t) \leq \frac{1}{R} \sum_{r=0}^{R-1} C_r^* + BT/V$$

$R$ frames of size $T$ slots:

Frame 1 | Frame 2 | Frame 3 | ... | Frame $R$

T-Slot Lookahead Problem for frame $r$ in $\{0, \ldots, R-1\}$:
$c_r^*$ computed below, *assuming future values of $(a(\tau), s(\tau), \gamma(\tau))$ are fully known* in frame $r$:

Minimize:

$$c_r^* \triangleq \frac{1}{T} \sum_{\tau=rT}^{(r+1)T-1} \gamma(\tau)x(\tau)$$

Subject to:

1. $\sum_{\tau=rT}^{(r+1)T-1} [s(\tau) + x(\tau) - a(\tau)] \geq 0$
2. $\sum_{\tau=rT}^{(r+1)T-1} [s(\tau) + x(\tau) - \epsilon] \geq 0$
3. $0 \leq x(\tau) \leq x_{max} \forall \tau \in \{rT, \ldots, (r+1)T - 1\}$
Simulations over Real Data Sets:
• We used 10 minute slot sizes (granularity of the available data)
• Compare to simple “Purchase at Deadline” algorithm.
• We chose $V=100 \Rightarrow D_{\text{max}} = 400$ slots (70 hours)
Same experiment: Histogram of Delay (V=100, ε= 87.5):
Our algorithm yields worst-case delay considerably less than the bound $D_{\text{max}}$. Worst case observed delay was 60 slots (10 hours)
Some more simulations: Changing the $\varepsilon$ parameter:
Some more simulations: Changing the V parameter:
• Lyapunov Optimization for Renewable Energy Allocation
• No need to know distribution. Robust to arbitrary sample paths.
• Explicit [O(1/V), O(V)] performance-delay tradeoff
Explanation of Why Delay is small even with \( \varepsilon=0 \)... 

Even with \( \varepsilon=0 \), we still get the same \( Q_{\text{max}} \) bound. (\( Q(t) \leq Q_{\text{max}} \) for all \( t \)).

Delay of requests that arrive on slot \( t \) is equal to the smallest integer \( T \) such that:

\[
\sum_{\tau=t}^{t+T} [s(\tau) + x(\tau)] \geq Q(t)
\]

So delay will be less than or equal to \( T \) whenever:

\[
\sum_{\tau=t}^{t+T} s(\tau) \geq Q_{\text{max}}
\]

There is no guarantee on how long this will take for arbitrary \( s(t) \) processes, but one can compute probabilities of exceeding a certain value if we try to use a stochastic model for \( s(t) \).