Efficient Algorithms for Renewable Energy Allocation to Delay Tolerant Consumers

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

A STORY

calculating Π

-- Feynman Point

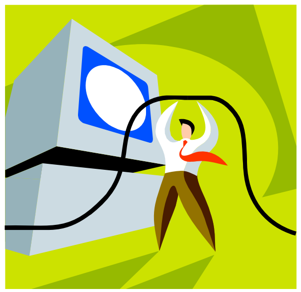
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899
8628034825342117067982148086513282306647093844609550582231725359408128481117450284
1027019385211055596446229489549303819644288109756659334461284756482337867831652712
0190914564856692346034861045432664821339360726024914127372458700660631558817488152
0920962829254091715364367892590360011330530548820466521384146951941511609433057270
3657595919530921861173819326117931051185480744623799627495673518857527248912279381
8301194912983367336244065664308602139494639522473719070217986094370277053921717629
3176752384674818467669405132000568127145263560827785771342757789609173637178721468
4409012249534301465495853710507922796892589235420199561121290219608640344181598136
2977477130996051870721134999999837297804995105973173281609631859502445945534690830
2642522308253344685035261931188171010003137838752886587533208381420617177669147303
5982534904287554687311595628638823537875937519577818577805321712268066130019278766
1119590921642019893809525720106548586327886593615338182796823030195203530185296899
5773622599413891249721775283479131515574857242454150695950829533116861727855889075
0983817546374649393192550604009277016711390098488240128583616035637076601047101819
429555961989467678374494482553797747268471040475346462080466842590694912...

A Story

calculating Π

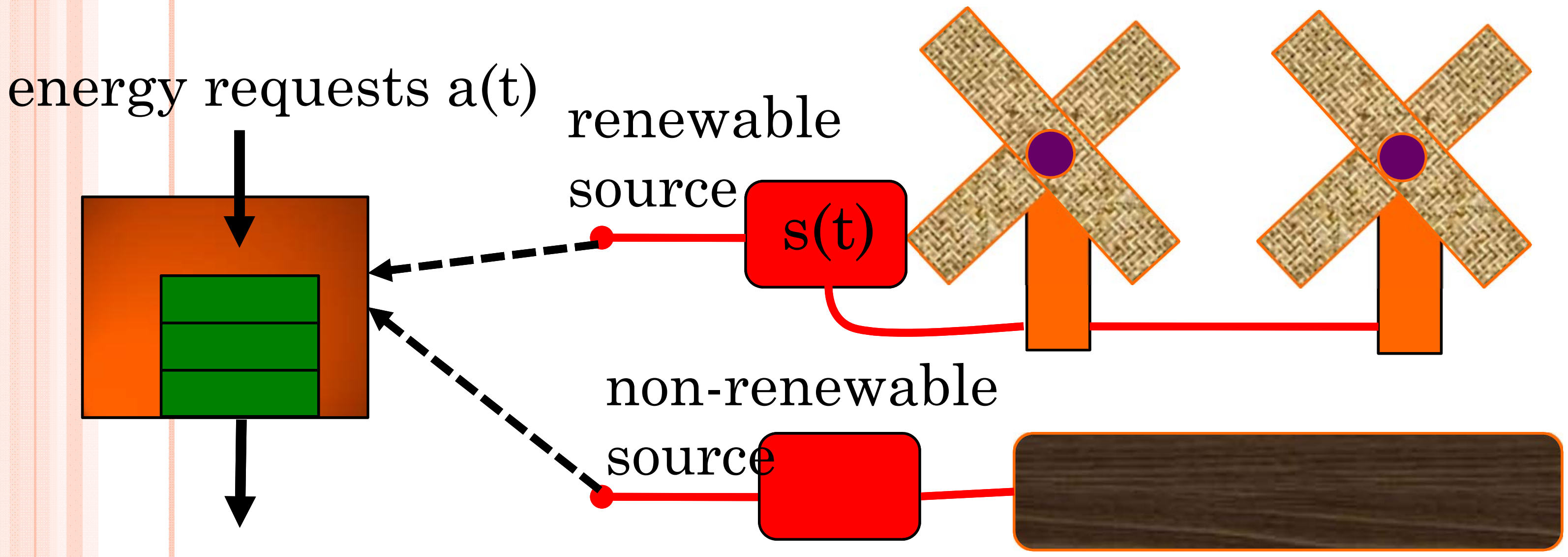
-- Feynman Point

Server



Me

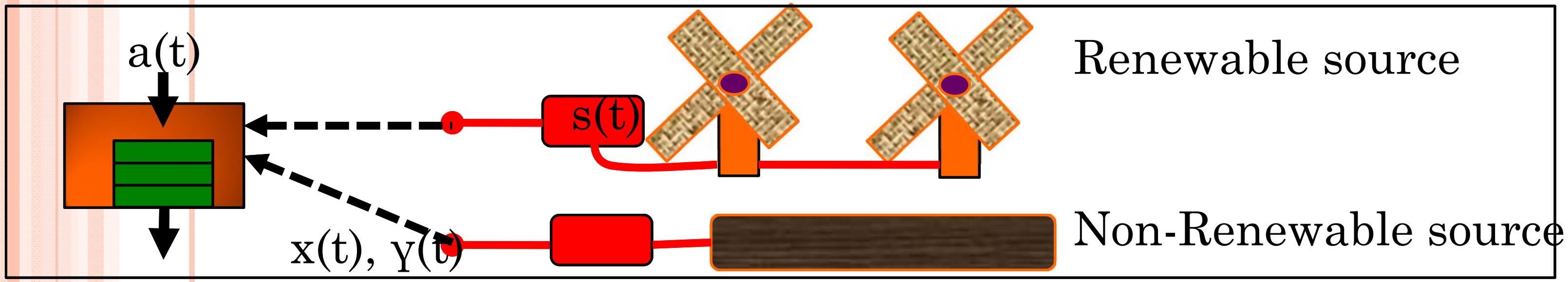
Other Users



•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_{max}.

•Might sometimes need to purchase energy from non-renewable source to meet the deadlines, and ***purchase price can be highly variable***.

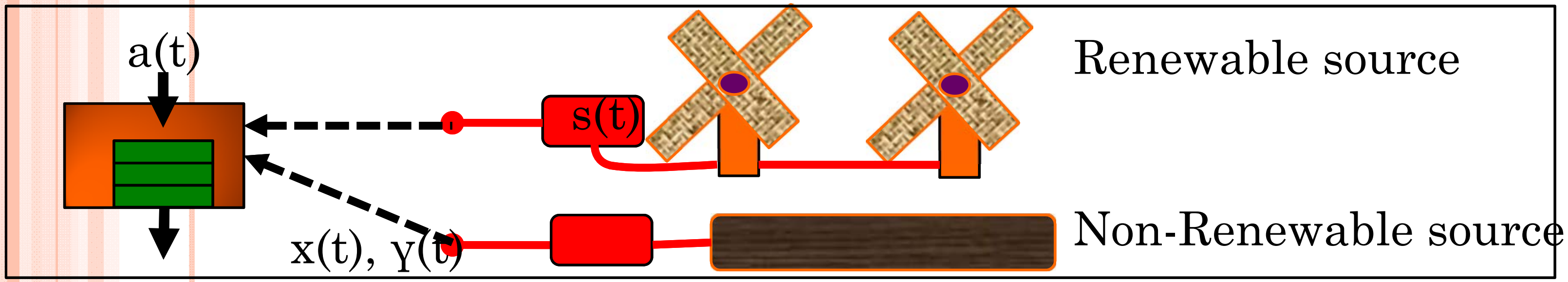


Outline:

- First Problem: Minimize time average cost of purchasing non-renewable energy

- Second Problem: Joint pricing of customers and purchasing of non-renewables

Problem 1: Minimize Average Cost of Non-Renewable Purcha



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

requests (random)

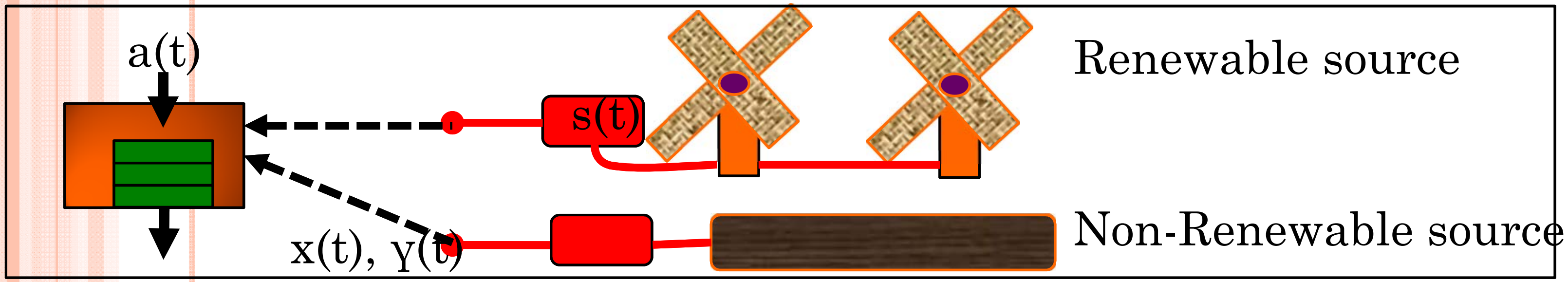
$$Q(t+1) = \max[Q(t) - s(t) - x(t), 0] + a(t) \quad , \quad cost(t) = x(t)\gamma(t$$

Renewable supply
(random)
(use-it-or-loose-it)

Non-Renewables purchased
(decision variable)

purchase price
(random)

Problem 1: Minimize Average Cost of Non-Renewable Purcha



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

Assumptions:

- For all slots t we have:

  $0 \leq a(t) \leq a_{max}$, $0 \leq s(t) \leq s_{max}$, $0 \leq \gamma(t) \leq \gamma_{max}$, $0 \leq x(t) \leq$

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

- $a_{max} \leq x_{max}$ (possible to meet all demands in 1 slot at high co

- (a(t), s(t), γ(t)) vector is i.i.d. over slots with ***unknown distr***

Problem 1: Minimize Average Cost of Non-Renewable Purcha

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

Possible formulation via Dynamic Programming (DP):

"Minimize average cost subject to max-delay D_{max}."

- This can be written as a DP, but requires distribution know

We will not use DP. We will take a different approach…

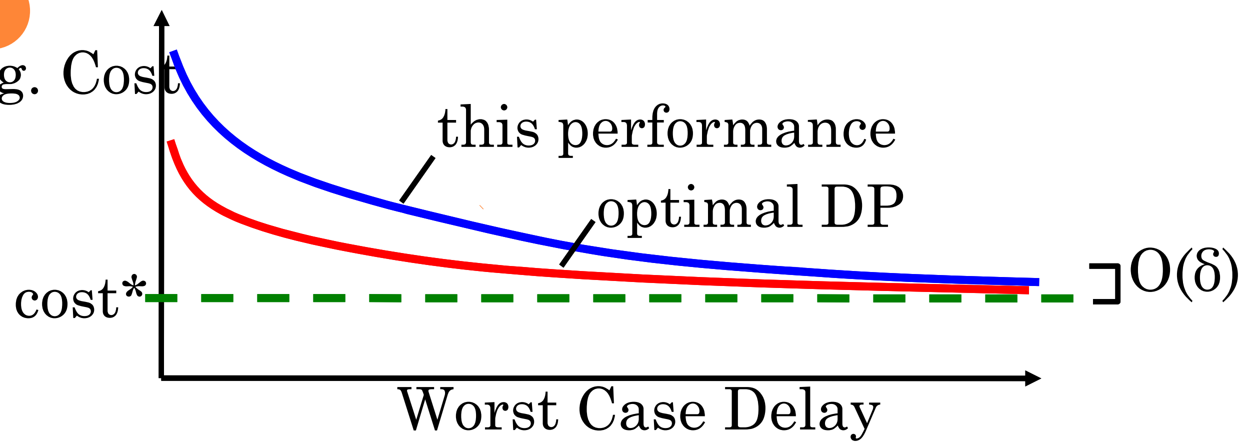
Problem 1: Minimize Average Cost of Non-Renewable Purcha

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

Relaxed Formulation via Lyapunov Optimization for Queue Ne

Minimize:	$E\{cost\}$ (time average)	
Subject to:	(1) $E\{Q\} <$ infinity	(a "queue stability" constrai
	(2) $0 \leq x(t) \leq x_{max}$ for all t	

- Define cost* = min cost subject to stability
- By definition: cost* ≤ cost delivered by ***any other alg*** (includi
- We will get within $O(\delta)$ of cost*, with ***worst-case delay***



Advantages of Lyapunov Optimization for Queueing Networks:

- No knowledge of distribution information is required.
- Explicit [O(δ), O(1/δ)] performance guarantees.
- Robust to changes in statistics, arbitrary correlations …
- Worst case delay bounds

Virtual Queue for Worst-Case Delay Guarantee (fix $\varepsilon>0$)

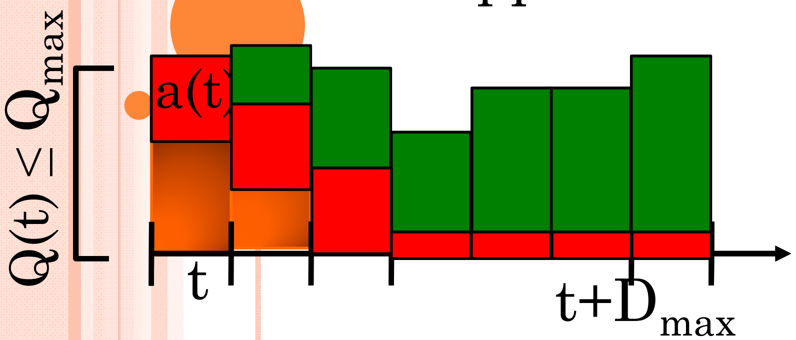
$a(t) \rightarrow$	$Q(t)$	$\rightarrow s(t)+x(t)$	Actual Queue
$\varepsilon 1\{Q(t)>0\} \rightarrow$	$Z(t)$	$\rightarrow s(t)+x(t)$	Virtual Queue (enforces ε-persistent service)

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

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

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

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



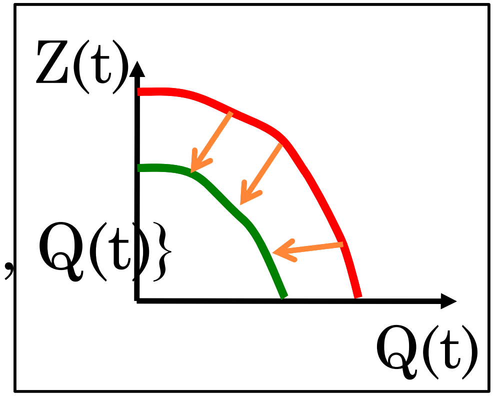
Then: $$\sum_{\tau=t}^{t+D_{max}} [s(\tau)+x(\tau)] \leq Q_{max}$$

Implies: $Z(t+D_{max}) > Z_{max}$ (contradiction)

Stabilize Z(t) and Q(t) while minimizing average cost c

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

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



Take actions to greedily minimize "***Drift-Plus-Weighted-Penalty***":

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

where V is a postiive 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 $(Z(t), Q(t), \gamma(t))$.

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

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

Define: $Q_{max} = V\gamma_{max} + a_{max}$, $Z_{max} = V\gamma_{max} + \varepsilon$

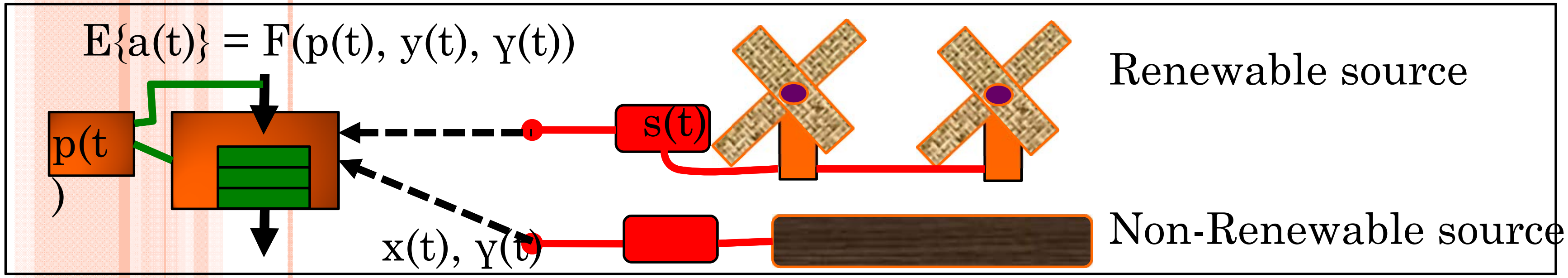
Theorem: Under the above algorithm:
(a) $Q(t) \leq Q_{max}$, $Z(t) \leq Z_{max}$ for all t.
(b) Delay $\leq (Q_{max} + Z_{max})/\varepsilon = O(V)$

Further, if $(s(t), a(t), \gamma(t))$ i.i.d. over slots, and if $\varepsilon \leq \max[E\{a(t)$
Then:

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

[where $B = (s_{max} + x_{ma}$

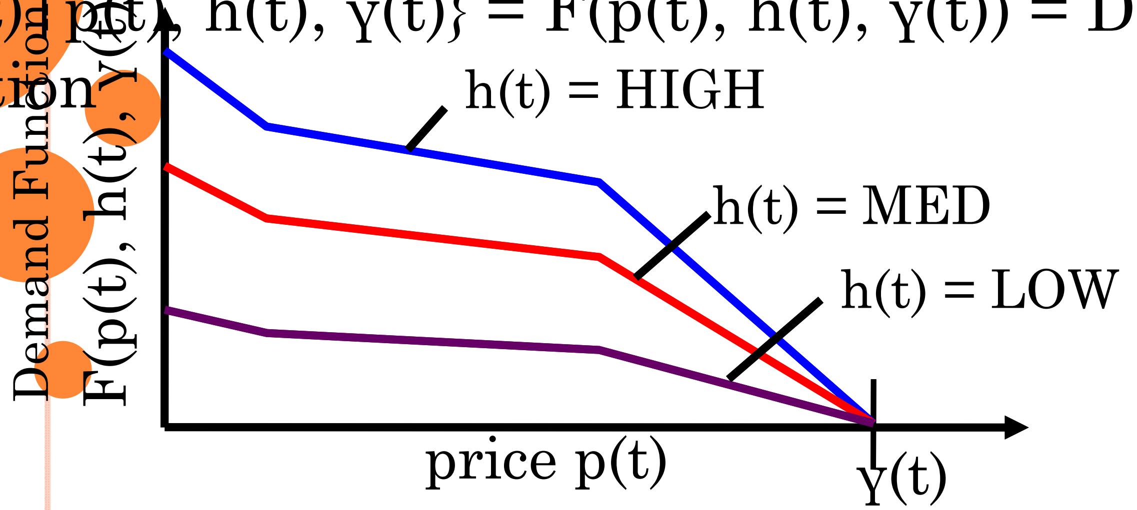
Problem 2: Joint Pricing and Energy Allocation



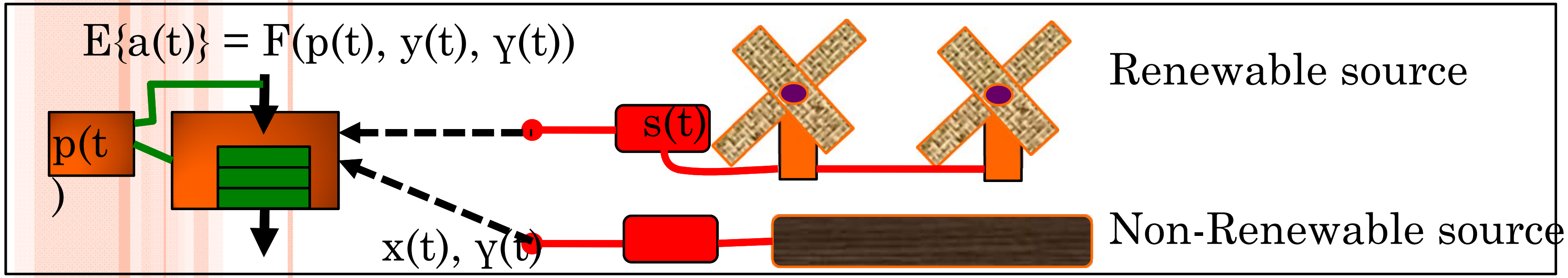
Same system model, with following extensions:

- a(t) = arrivals = Random function of pricing decision p(t)
- h(t) = additional "demand state" (e.g. "HIGH, MED, LOW")
- $E\{a(t) | p(t), h(t), \gamma(t)\} = F(p(t), h(t), \gamma(t))$ = Demand Function

Example:



Problem 2: Joint Pricing and Energy Allocation



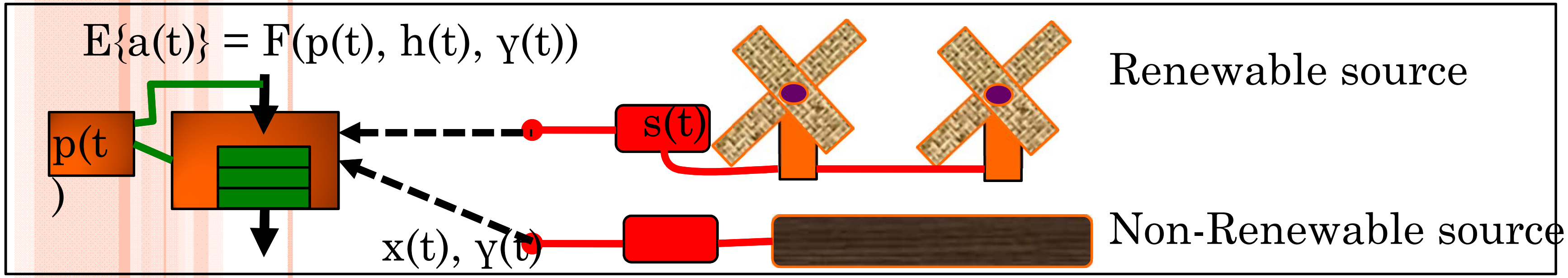
Same system model, with following extensions:

- $a(t)$ = arrivals = Random function of pricing decision $p(t)$
- $h(t)$ = additional "demand state" (e.g. "HIGH, MED, LOW")
- $E\{a(t) | p(t), h(t), \gamma(t)\} = F(p(t), h(t), \gamma(t))$ = Demand Function

New Problem:

- Profit(t) $= a(t)p(t) - x(t)\gamma(t)$
- Maximize Time Average Profit!
- Profit* = Optimal Time Avg. Profit Subject to Stability

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

$$\Delta(t) - VE\{\text{Profit}(t) \mid Z(t), Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t) \mid Z(t)$$

Resulting Algorithm:

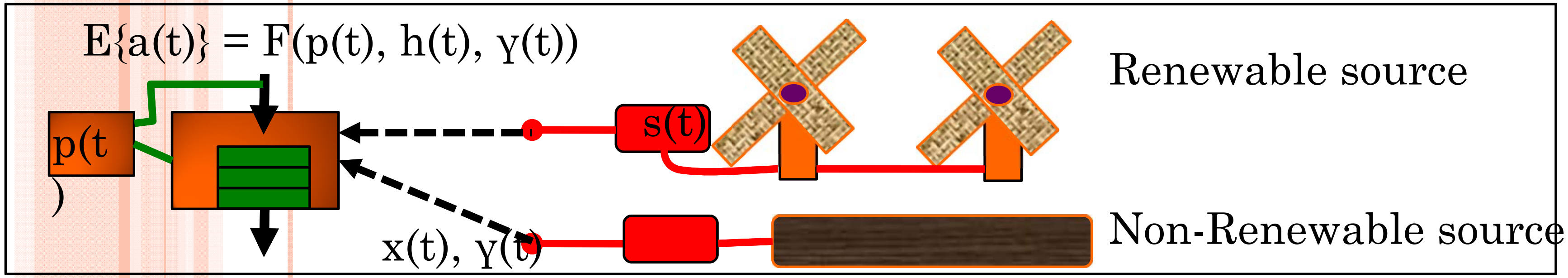
Every slot t, observe $(h(t), Z(t), Q(t), \gamma(t))$. Then:

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

$$\boxed{\begin{aligned} &\text{Maximize:} \quad F(p(t),h(t),\gamma(t))(Vp(t) - Q(t)) \\ &\text{Subject to:} \quad 0 \leq p(t) \leq p_{max} \end{aligned}}$$

- (Purchasing) Choose $x(t)$ same as before.
- Update queues $Q(t)$, $Z(t)$ same as before.

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

$\Delta(t) - VE\{Profit(t) \,|\, Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t) \,|\, Z(t)$

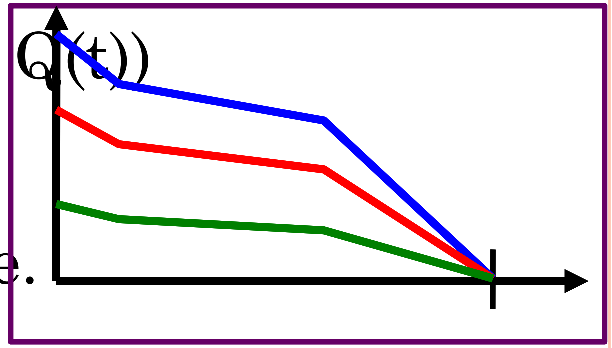
Resulting Algorithm:

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

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

Maximize: $F(p(t),h(t),\gamma(t))(Vp(t) - Q(t))$

Subject to: $0 \leq p(t) \leq p_{max}$

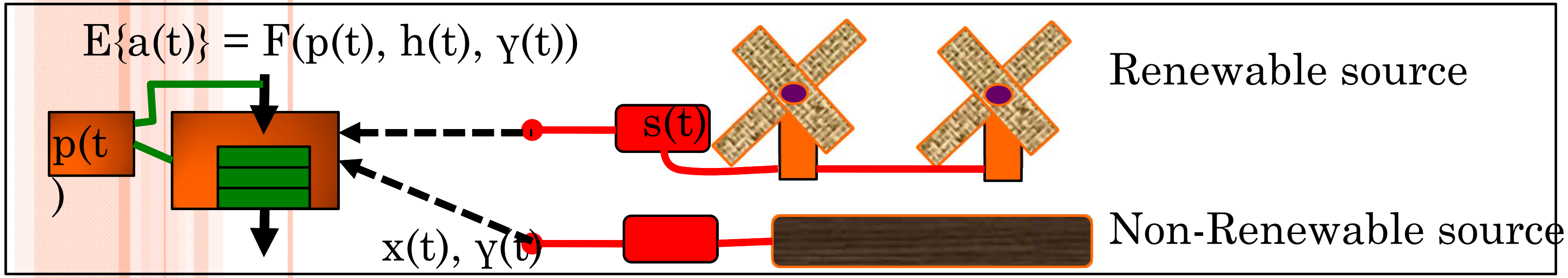


•(Purchasing) Choose x(t) same as before.

•Update queues Q(t), Z(t) same as before.

*If F(p,h,γ) = β(h)G(p,γ), don't need to know demand state

Problem 2: Joint Pricing and Energy Allocation



Drift-Plus-Penalty for New Problem:

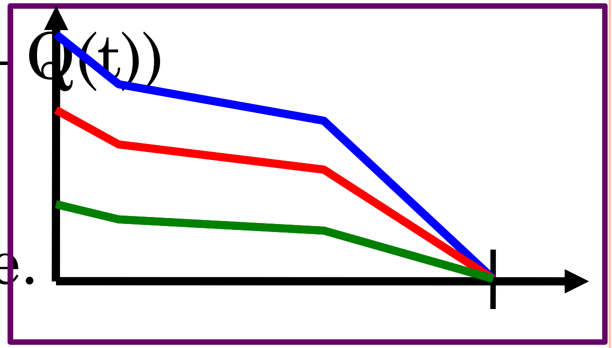
$\Delta(t) - VE\{Profit(t) \mid Z(t),Q(t)\} = \Delta(t) - VE\{a(t)p(t) - x(t)\gamma(t) \mid Z(t)$

Resulting Algorithm:

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

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

> Maximize: $\beta(h(t))G(p(t),\gamma(t))(Vp(t) - Q(t))$
> Subject to: $0 \leq p(t) \leq p_{max}$

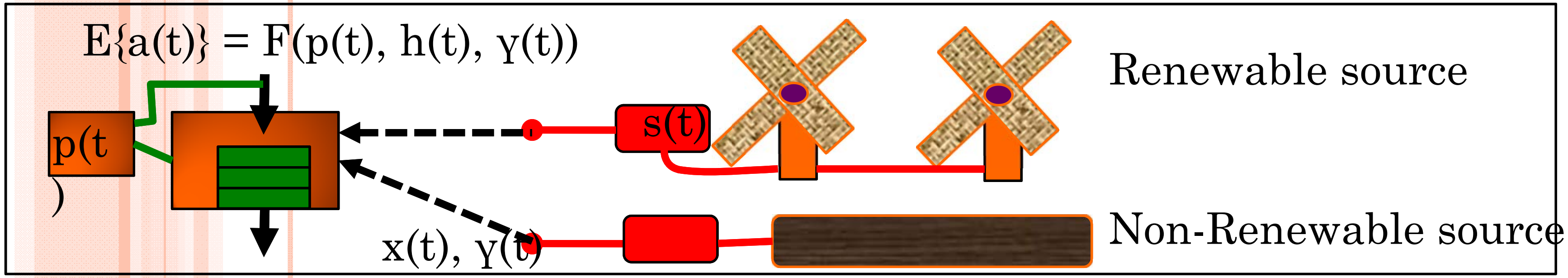


•(Purchasing) Choose x(t) same as before.

•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

Problem 2: Joint Pricing and Energy Allocation



Theorem: Under the joint pricing and energy allocation algor

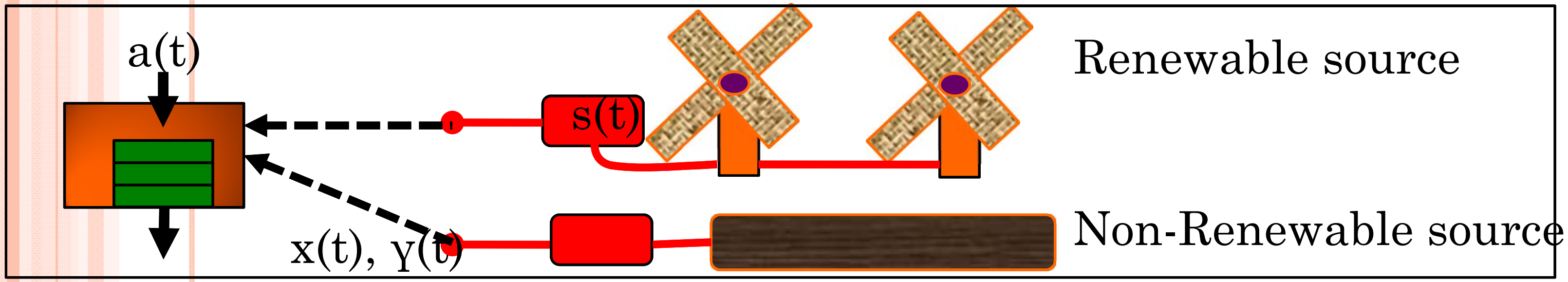
(a) Worst case queue bounds Q_{max}, Z_{max} same as before.

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

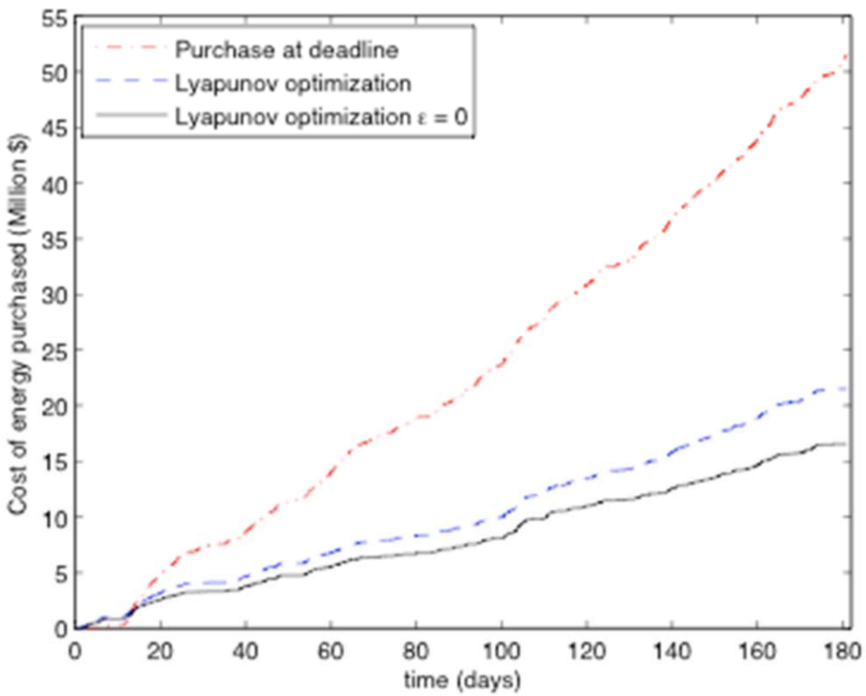
(c) If (s(t), γ(t), h(t)) i.i.d. over slots, and $\varepsilon \leq E\{s(t)\}$, then:

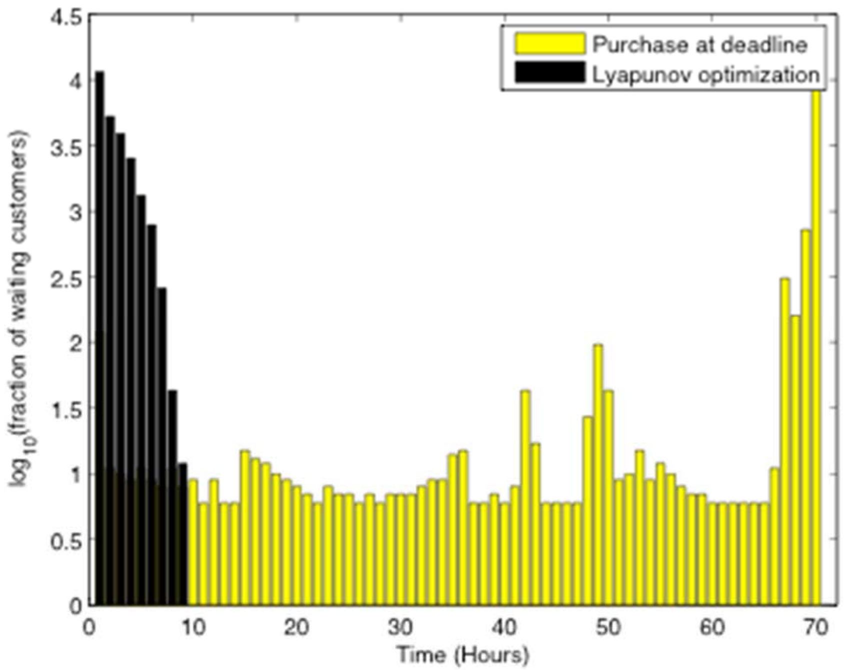
$$E\{profit\} \geq profit^* - O(1/V)$$

Concluding Slide:



•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





Thank you!

Explanation of Why Delay is small even with $\varepsilon=0$…

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

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

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