Distributed Congestion Control and Related Problems in Complex Networks

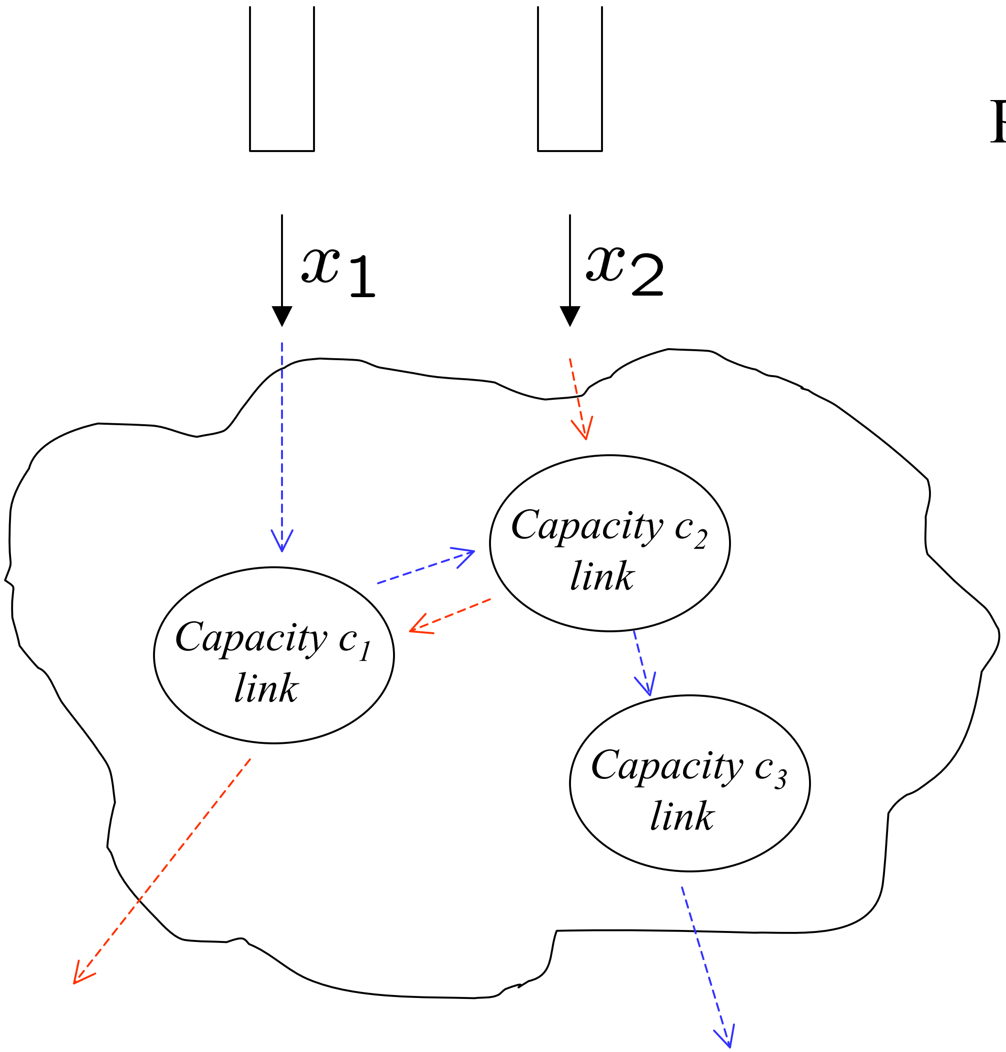
A.L.Stolyar
Bell Labs, Murray Hill, NJ

September 27, 2005

Outline

- Motivation
- General network model, problem statement, GPD algorithm
- Scope of the model: several applications
- Analysis

One Motivation: Congestion control of a complex network



Previous work: Kelly problem

$$\max_x \sum_n U_n(x_n)$$

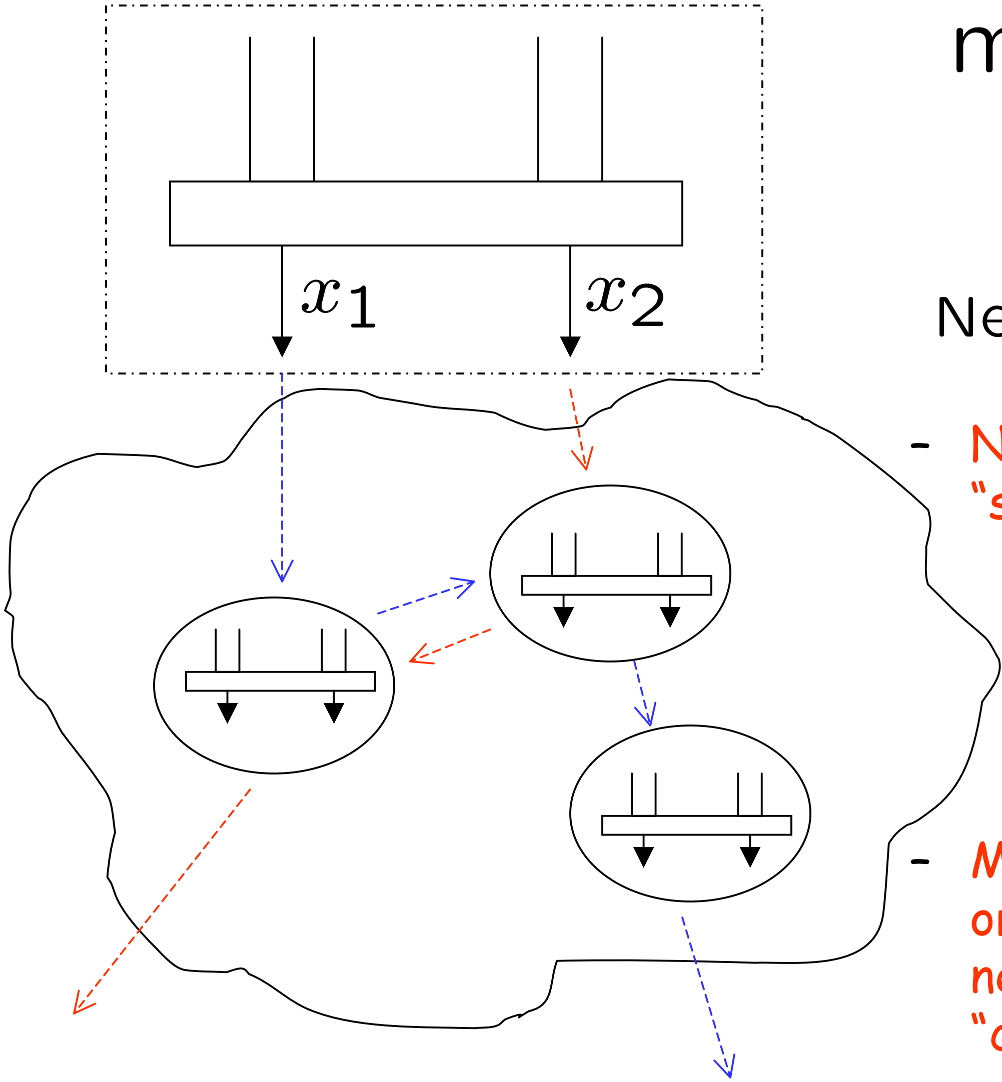
subject to

Each link ℓ is not overloaded:

$$\sum_{n \in F(\ell)} x_n \leq c_\ell$$

- TCP congestion control implicitly tries to solve this problem
- Very large and very active field. F.Kelly et al., S.Low et al., …

One Motivation: Congestion control of a complex network



$$\max_x U(x)$$

subject to

Network queues are stable

- Network nodes are time-varying "switches", which need to be scheduled
  - Sources may be dependent and need to be scheduled jointly
- Maintaining "desired" average injection or service rates may not be practical – need control strategy which uses "current state" only.
- Network nodes may have power usage constraints

Network control problem
and its underlying convex optimization problem

A "STANDARD" CONVEX OPTIMIZATION PROBLEM

No constraints on the underlying domain of x

$$\max_x \sum_n U_n(x_n)$$

subject to

Each link ℓ is not overloaded:

$$\sum_{n \in F(\ell)} x_n \leq c_\ell$$

EXPLICIT (linear) constraints

NETWORK CONTROL PROBLEM

IMPLICIT (linear) constraints on the underlying domain of x

Find CONTROL maximizing $U(x)$

subject to

Network queues are stable

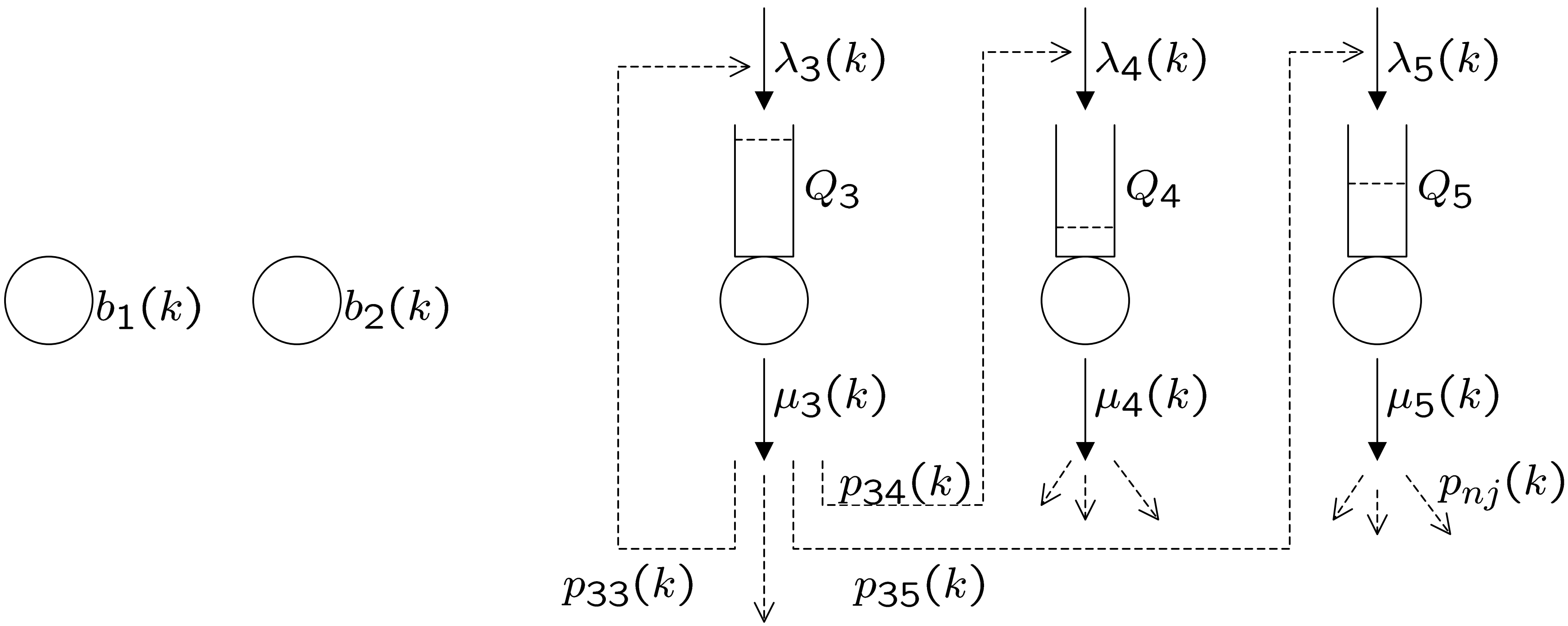
IMPLICIT (linear) constraints on x, making queueing stability feasible

UNDERLYING CONVEX OPTIMIZATION PROBLEM

General network model

"Utility" nodes $\mathcal{N}^u$ $[= \{1,2\}]$

"Processing" nodes $\mathcal{N}^p$ $[= \{3,4,5\}]$



Discrete time *t=0,1,2,…*

Control *k* at *t* is chosen from a finite set *K(m(t))*,

m(t) is underlying random "network mode," finite set of modes.

For any control *k*, it is allowed to "skip" service of any queue

Problem

$$b(k) = (b_n(k), n \in \mathcal{N}^u)$$

$X = E[b(k(t))]$ “Steady-state” average commodity vector, *under a given control strategy*

$$Q(t) = (Q_n(t), n \in \mathcal{N}^p)$$

$$\max U(X)$$

$$\text{s.t. } Q(t) \text{ is stable}$$

Utility function U is continuously differentiable concave (possibly non-strictly concave)

(Asymptotically) Optimal solution: Greedy Primal-Dual (GPD) algorithm

$$k(t) \in \arg\max_k \nabla U(X(t)) \cdot b(k) - \beta Q(t) \cdot \overline{\Delta Q}(k)$$

$$\beta > 0 \quad \text{small parameter}$$

$$X(t+1) = \beta b(k(t)) + (1-\beta) X(t)$$

$$(\overline{\Delta Q})_n(k) \doteq \lambda_n(k) - \mu_n(k) + \sum_{j \in \mathcal{N}^p} \mu_j(k) p_{jn}(k)$$

$\overline{\Delta Q}(k) =$ expected queue drift vector, assuming queues are large enough (i.e. *without worrying about empty queues effects*)

MAIN RESULT (informally):

GPD algorithm is close to optimal when β is small.

GPD algorithm: Preliminary discussion

$$k(t) \in \arg\max_k \; \nabla U(X(t)) \cdot b(k) - \beta Q(t) \cdot \overline{\Delta Q}(k)$$

$$X(t+1) - X(t) = \beta[b(k(t)) - X(t)]$$

$$\overline{\Delta Q}(k) = \text{ expected queue drift vector}$$

GPD rule interpretation: *"Greedily" maximize expected drift of*

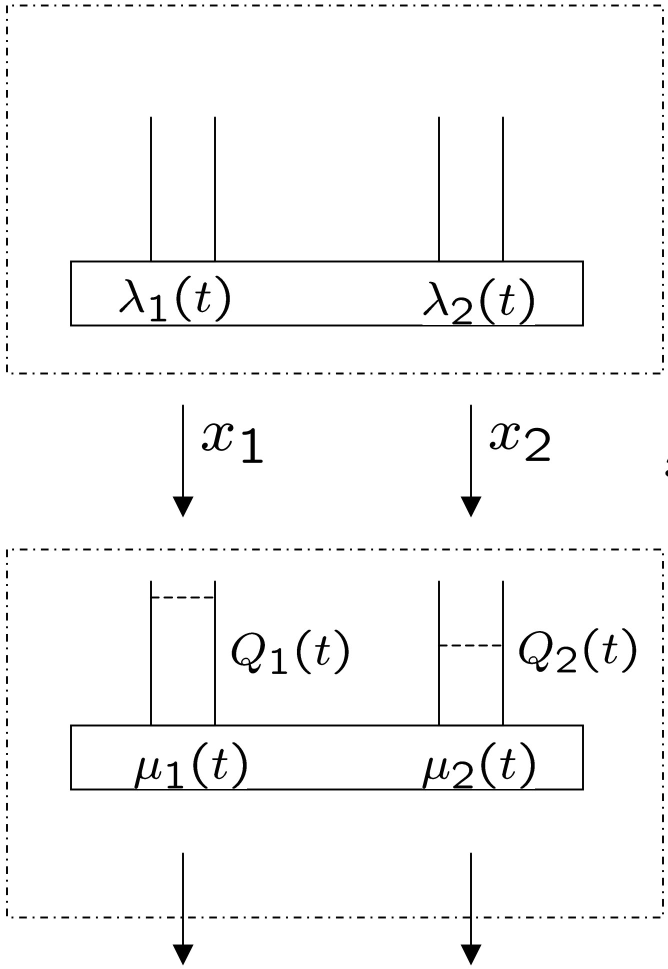
$$F(X(t), Q(t)) = U(X(t)) - \frac{1}{2}\beta \sum_{n \in \mathcal{N}^p} Q_n(t)^2$$

If NO processing nodes: GPD => *"Gradient"* alg., *U(X(t))* is "almost" Lyapunov function

If NO utility nodes: GPD => *"MaxWeight"* alg., $\sum Q_n(t)^2$ is Lyapunov function

GPD may be viewed is a "naïve" combination of Gradient and MaxWeight. Optimality is non-trivial, because for this general model *F(X(t),Q(t))* is NOT a Lyapunov function

Example



Slotted time t=0,1,2,…

Two dependent traffic sources n=1,2.
"Source switch"
$\lambda(t) = (\lambda_1(t), \lambda_2(t))$

$x = (x_1, x_2)$ Average traffic rates

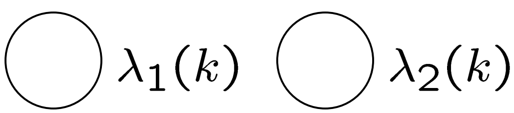
Two dependent servers n=1,2.
"Processing switch"
$(\mu_1(t), \mu_2(t))$

$$\max_x U(x)$$

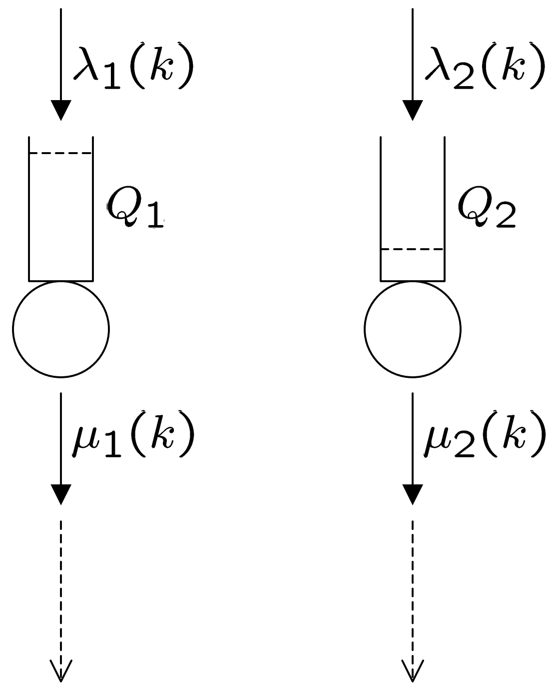
s. t. queues are stable

Example: Mapping to general model

Utility nodes

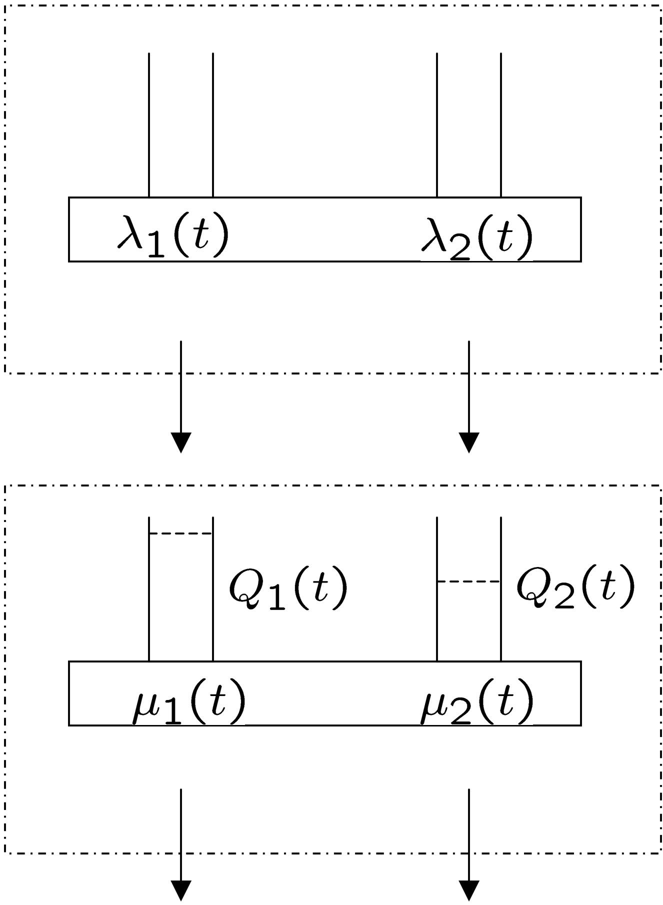


Processing nodes



Control k = Source switch control (λ_1,λ_2) + Processing switch control (μ_1,μ_2)

Example: GPD algorithm instance



Source switch. Knows *its* utility function, keeps track of *its* average traffic injection rates $X(t)=(X_1(t),X_2(t))$, uses queue lengths of the nodes it directly injects traffic into:

$$\lambda(t) \in \arg\max_{\lambda} \; [\nabla U(X(t)) - \beta(Q_1(t), Q_2(t))] \cdot \lambda$$

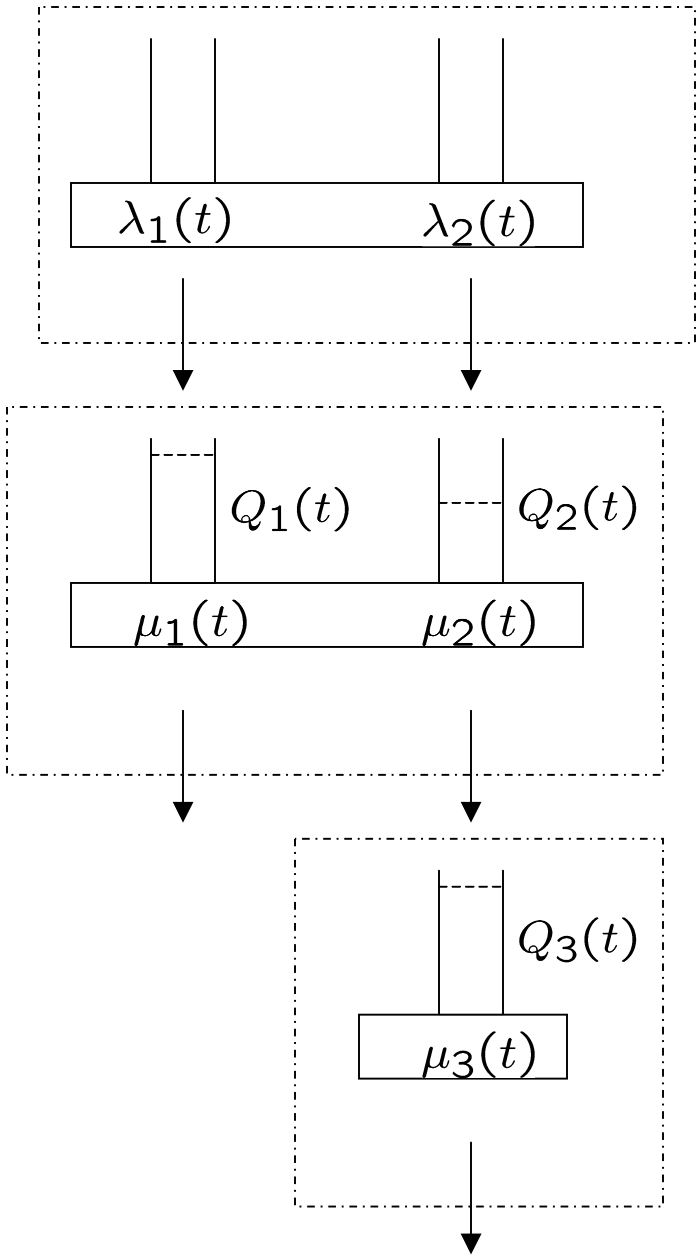
$$X(t+1) = \beta\lambda(t) + (1-\beta)X(t)$$

Processing switch: Uses *its own* queue lengths:

$$(\mu_1(t), \mu_2(t)) \in$$

$$\arg \max_{(\mu_1,\mu_2)} Q_1(t)\mu_1 + Q_2(t)\mu_2$$

Slightly more general example



Source switch:

$$\lambda(t) \in \arg\max_{\lambda} \ [\nabla U(X(t)) - \beta(Q_1(t), Q_2(t))] \cdot \lambda$$

$$X(t+1) = \beta\lambda(t) + (1-\beta)X(t)$$

Processing switch. Uses its own queue lengths + queue lengths of the nodes it directly forwards traffic to:

$$(\mu_1(t), \mu_2(t)) \in$$

$$\arg\max_{(\mu_1,\mu_2)} Q_1(t)\mu_1 + Q_2(t)\mu_2 - Q_3(t)\mu_2$$

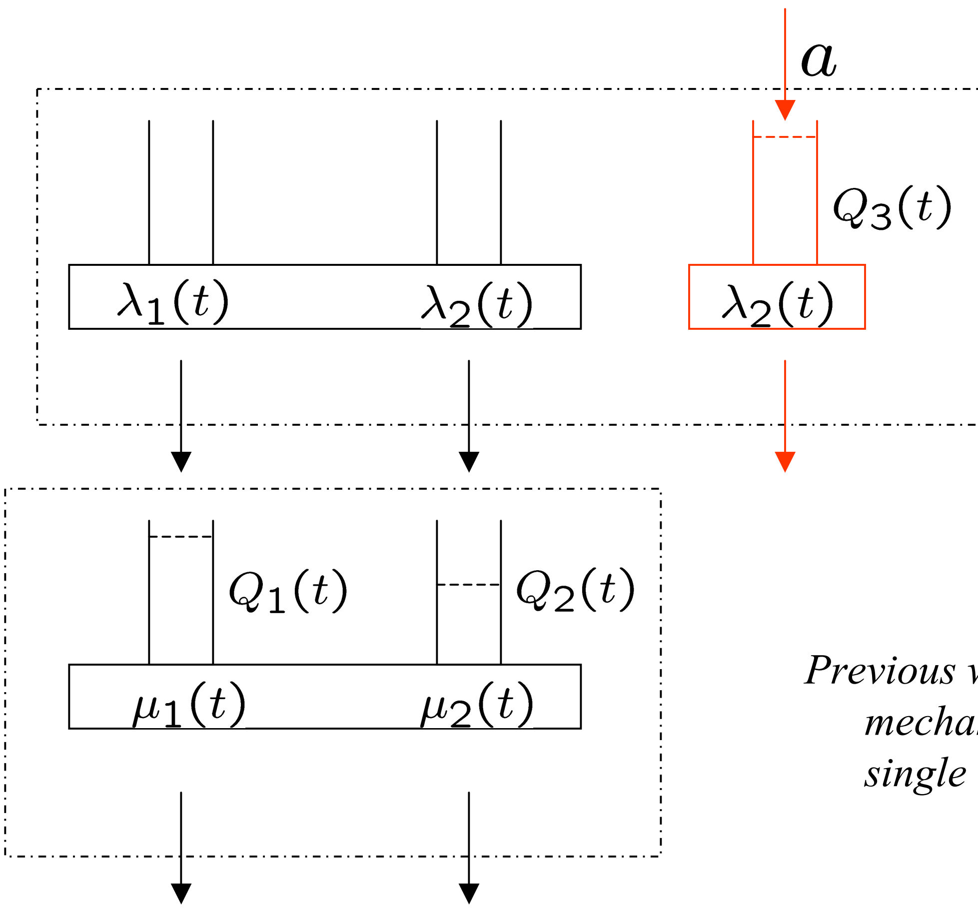
Another processing switch:

$$\text{maximize } \mu_3(t)$$

Application: Adding average rate constraints

$$\lambda(t) \in \arg\max_{\lambda} [\nabla U(X(t)) - \beta(Q_1(t), Q_2(t))] \cdot \lambda + \beta Q_3(t) \lambda_2(t)$$

$$X(t+1) = \beta\lambda(t) + (1-\beta)X(t); \quad Q_3(t+1) = [Q_3(t) + a - \lambda_2(t)]^+$$



$$\max_{x} U(x)$$

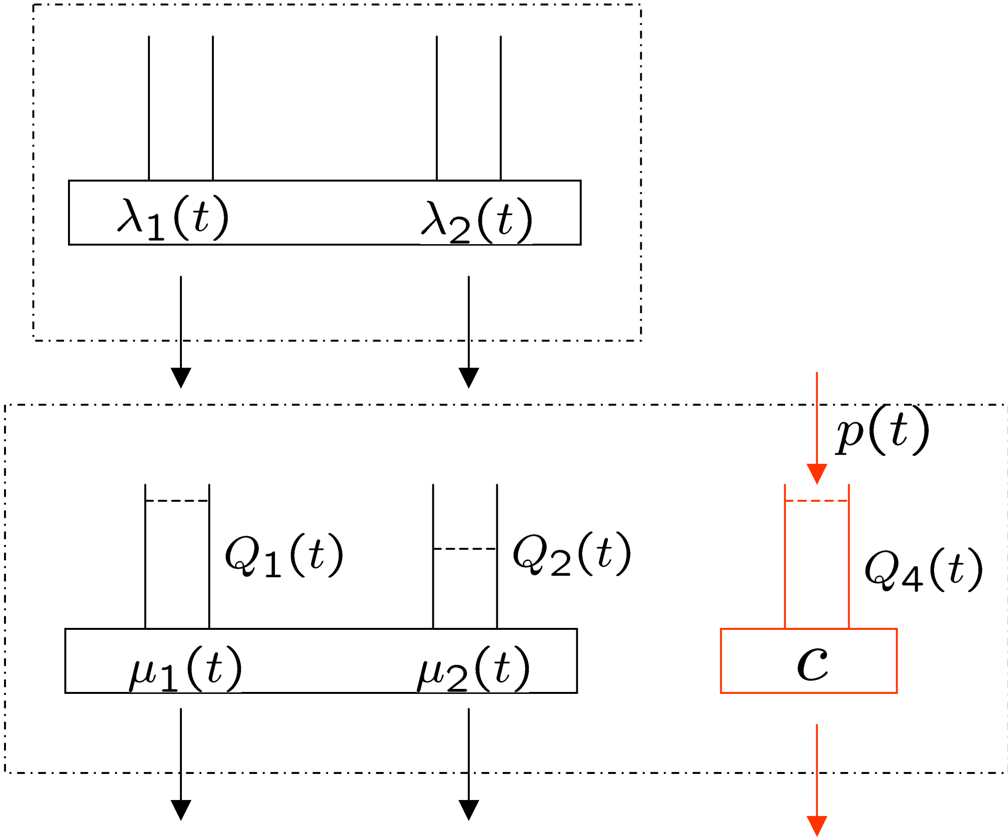
s. t. queues are stable

Additional constraint:

$$x_2 = E[\lambda_2(t)] \geq a$$

Previous work: Similar "virtual token queue" mechanism for rate constraint(s) enforcement for a single "source switch" [Andrews-Qian-Stolyar'05].

Application: Adding average power usage constraints



$$\max_x U(x)$$

s. t. queues are stable

Additional constraint:

$$E[p(t)] \leq c$$

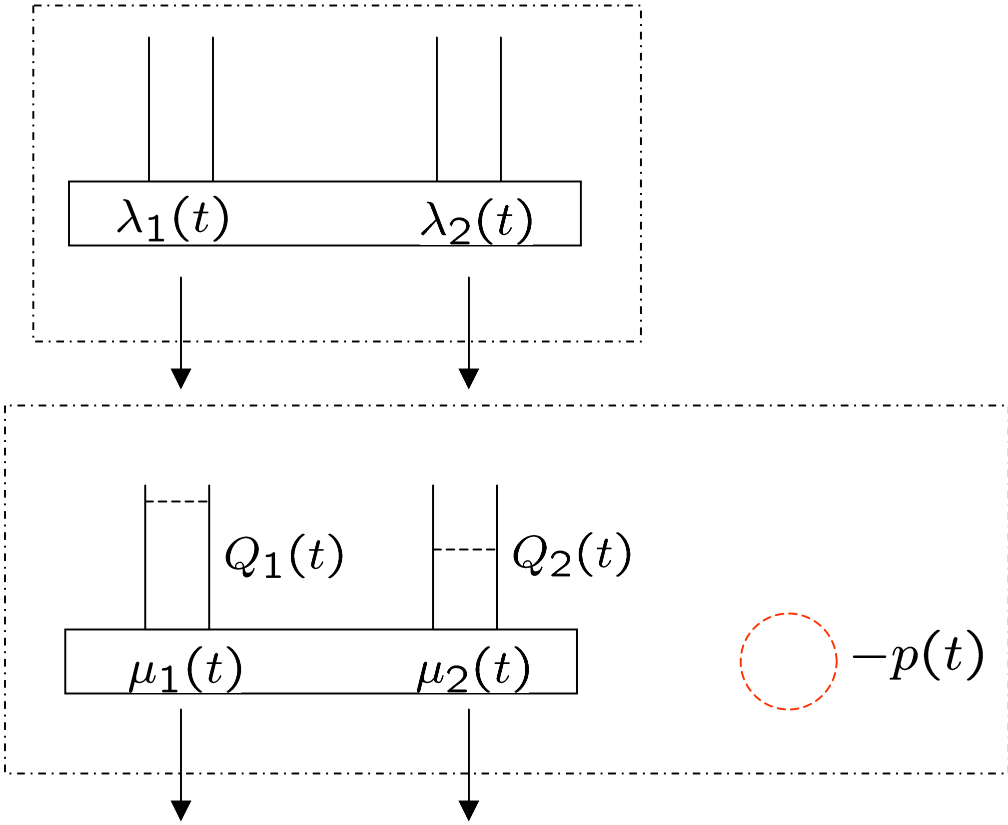
$$(\mu_1(t), \mu_2(t), p(t)) \in \arg \max_{(\mu_1, \mu_2, p)} Q_1(t)\mu_1 + Q_2(t)\mu_2 - Q_4(t)p$$

$$Q_4(t+1) = [Q_4(t) + p(t) - c]^+$$

Some previous work on models with power constraints (special models, alg's different from GPD):

Tse-Hanly'98, Klein-Viswanathan'03, Yeh-Cohen'03.

Application: Minimizing average power usage



$$\min E[p(t)]$$

s. t. queues are stable

$$(\mu_1(t), \mu_2(t), p(t)) \in \arg \max_{(\mu_1, \mu_2, p)} -p + \beta Q_1(t)\mu_1 + \beta Q_2(t)\mu_2$$

Some previous work on minimizing average power usage (different alg's):

Cruz-Santhanam'03, Giaccone-Prabhakar-Shah'03

Application: "Distributed" algorithm for linear programs

$$\max\ c_1x_1 + c_2x_2$$

$$a_{11}x_1 + a_{12}x_2 + a_{13} \leq 0$$

$$a_{21}x_1 + a_{22}x_2 + a_{23} \leq 0$$

$$x_1, x_2 \in [\ell, h]$$

x_j is the commodity rate, $b_j(t) \in \{\ell, h\}$

Each inequality constraint *i* has assoc. processing node with queue length $Q_i(t)$

Arbitrary $X_1(0), X_2(0) \in R$, Arbitrary $Q_1(0), Q_2(0) \geq 0$

$$b_j(t) = \arg\max_{y\in\{\ell,h\}} [c_j - \beta a_{1j}Q_1(t) - \beta a_{2j}Q_2(t)]y,\ j = 1,2$$

$$X_j(t+1) = \beta b_j(t) + (1-\beta)X_j(t),\ j = 1,2$$

$$Q_i(t+1) = [Q_i(t) + a_{i1}b_1(t) + a_{i2}b_2(t) + a_{i3}]^+,\ i = 1,2$$

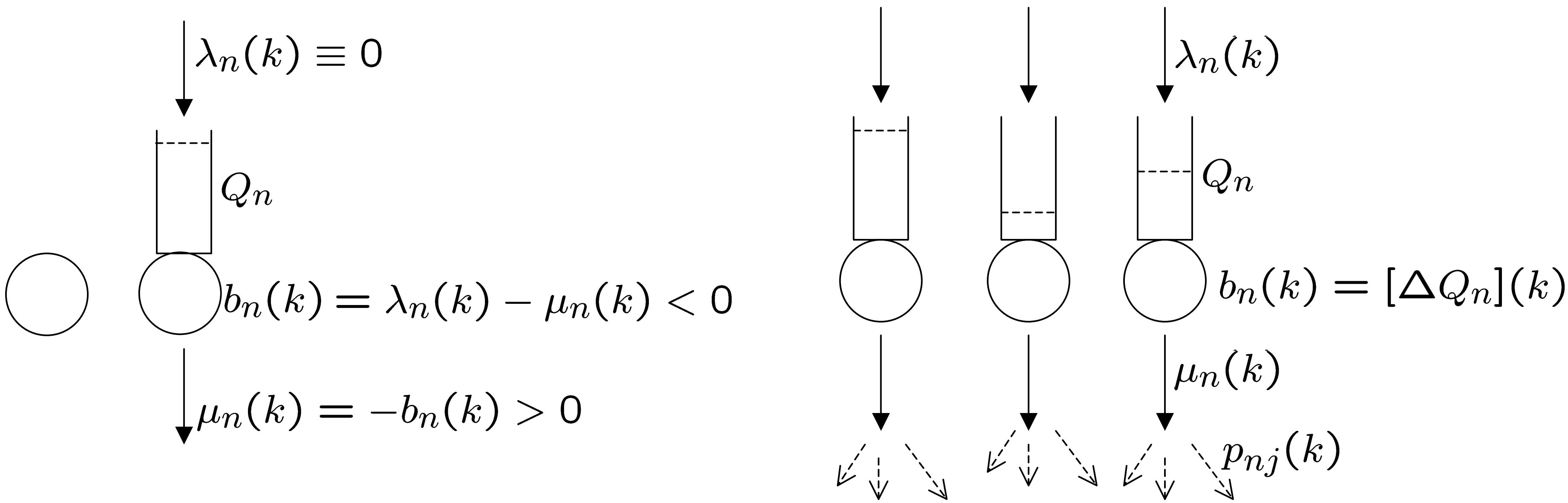
$(X_1(t), X_2(t)) \to$ neighborhood of optimal set

$(\beta Q_1(t), \beta Q_2(t)) \to$ neighborhood of opt. set of the dual

General model analysis: Unified treatment of all nodes

“Utility” nodes $\mathcal{N}^u$

“Processing” nodes $\mathcal{N}^p$



Set of all nodes $\mathcal{N} = \mathcal{N}^u \cup \mathcal{N}^p$

Problem

$$b(k) = (b_n(k), n \in \mathcal{N})$$

$X = E[b(k(t))]$ "Steady-state" average commodity vector, *under a given control strategy*

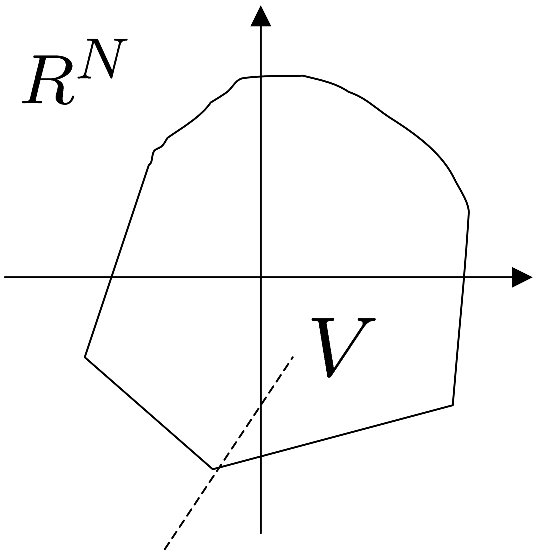
$$Q(t) = (Q_n(t), n \in \mathcal{N})$$

$$\max U(X)$$

$$\text{s.t. } Q(t) \text{ is stable}$$

Utility function U is cont. diff. concave (possibly non-strictly)

Underlying convex optimization problem



Convex compact

Rate region V = {Set of all possible long-term "commodity rate" vectors $X = E[b(k(t))]$ }

equivalently

Rate region V = {Set of all possible long-term "queue drift" vectors $E[\Delta Q(k(t))]$, *assuming queues are large*}

$$\max_{x \in V} U(x)$$

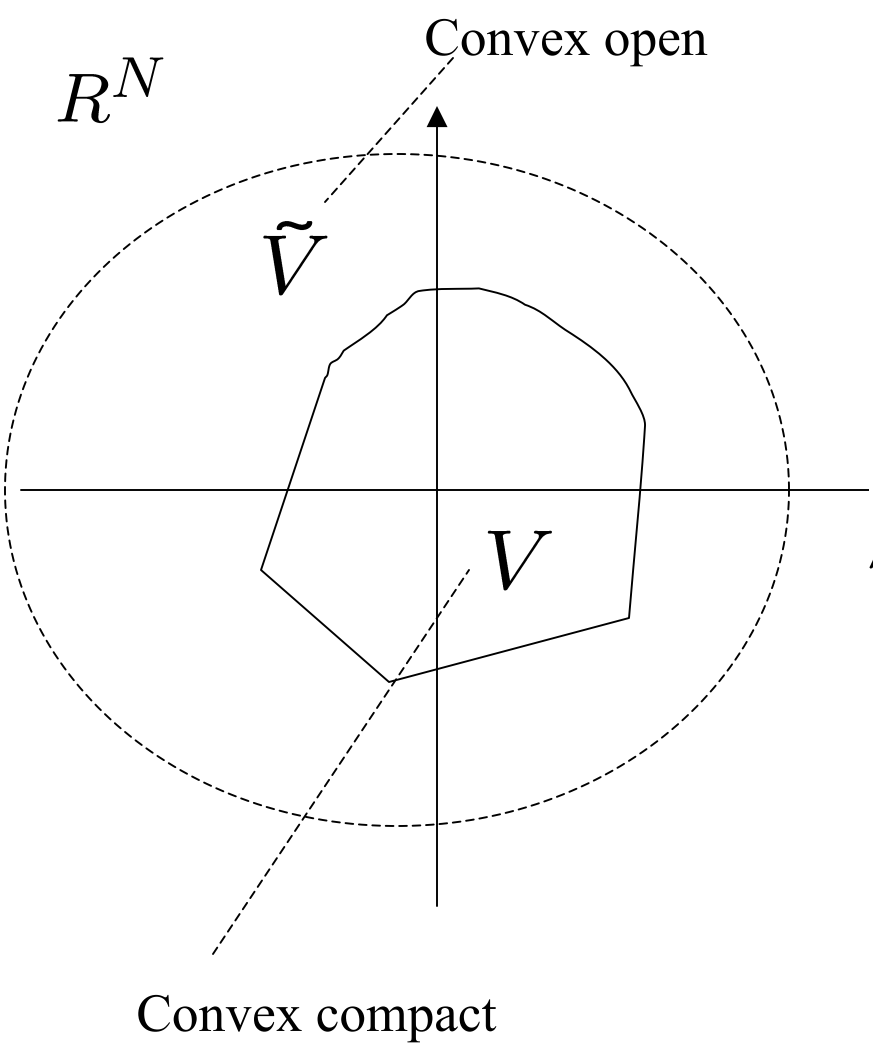
subject to

$$x_n \leq 0, \ \ n = 1, \ldots, N$$

V^* optimal set

Q^* optimal set for the dual

Convergence to a greedy primal-dual dynamic system



Concave cont. diff.

$$U(x),\ \ x \in \tilde{V}$$

THEOREM 1:

Consider GPD alg., and let $\beta \downarrow 0$.

Assume $(X(0), \beta Q(0)) \to (x(0), q(0)) \in \tilde{V} \times R_+^N$.

Then, $(X(t/\beta), \beta Q(t/\beta)) \to (x(t), q(t))$

such that

$$x'(t) = v(t) - x(t)$$

$$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$$

$q'(t) = v(t)$, and $q(t)$ must stay in R_+^N

Greedy primal-dual dynamic system

$$(x(t), q(t)), t \geq 0,$$

$$x'(t) = v(t) - x(t)$$

$$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$$

$$q'(t) = v(t), \text{ and } q(t) \text{ must stay in } R_+^N$$

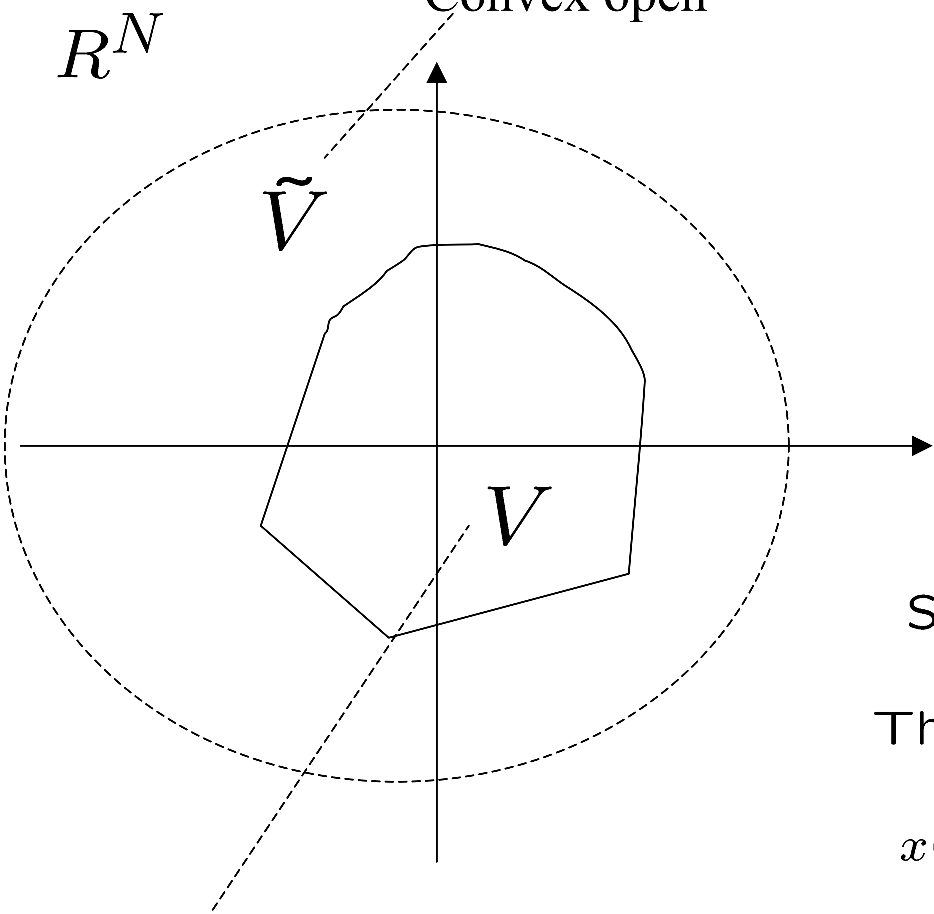
FIRST INTERPRETATION (LAGRANGIAN):

Greedily maximize $[\nabla_x [U(x(t)) - q(t) \cdot x(t)]] \cdot x'(t)$

SECOND INTERPRETATION:

Greedily maximize $\frac{d}{dt}[U(x(t)) - \frac{1}{2} q(t) \cdot q(t)]$

Main result: attraction property of the dynamic system



Concave cont. diff.

$U(x),\ x \in \tilde{V}$

Dynamic system: $(x(t), q(t)), t \geq 0$,

$x'(t) = v(t) - x(t)$

$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$

$q'(t) = v(t)$, and $q(t)$ must stay in R_+^N

<u>THEOREM 2</u>:

Suppose $\exists v \in V,\ v_n < 0\ \forall n$.

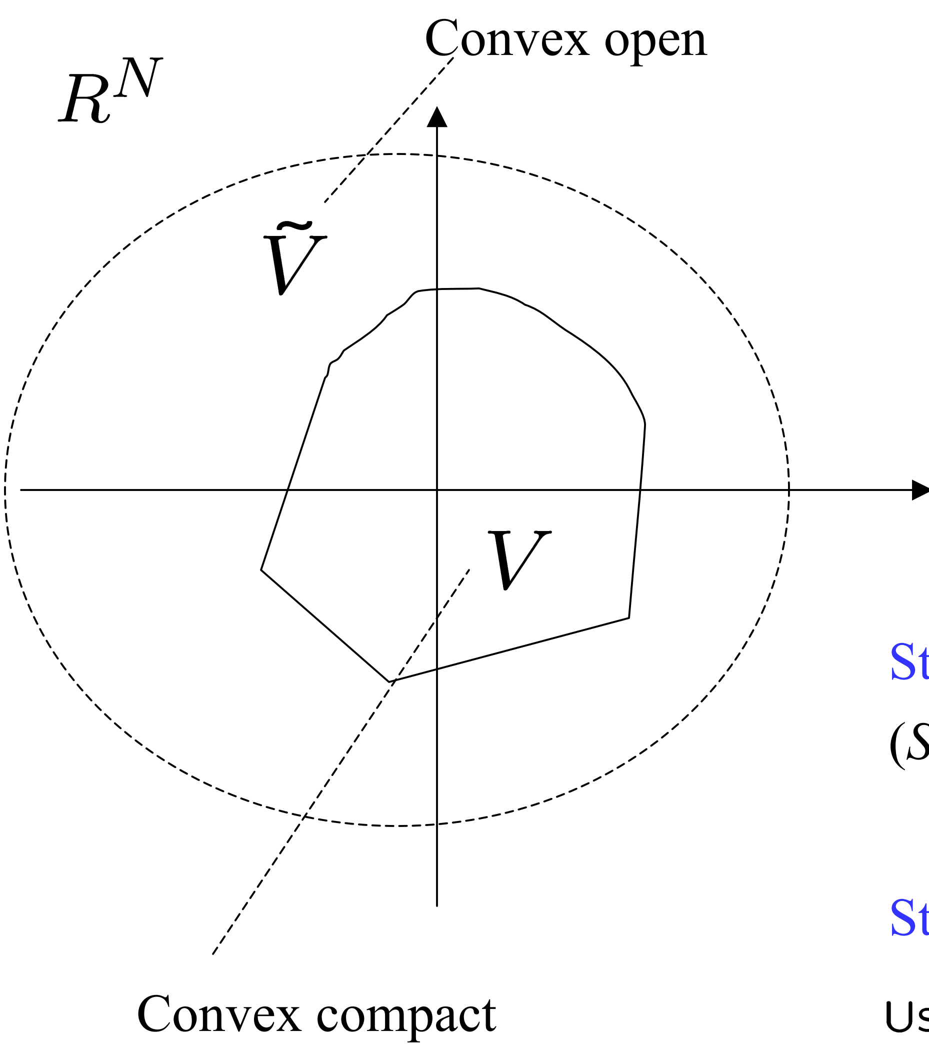
Then, $\forall (x(0), q(0)) \in \tilde{V} \times R_+^N$,

$x(t) \to V^*, \quad q(t) \to q^* \in Q^*$.

Conv. $(x(t), q(t)) \to V^* \times Q^*$ uniform,

if $(x(0), q(0))$ within a compact.

Proof outline



Dynamic system: $(x(t), q(t)), t \geq 0,$

$$x'(t) = v(t) - x(t)$$

$$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$$

$$q'(t) = v(t), \text{ and } q(t) \text{ must stay in } R_+^N$$

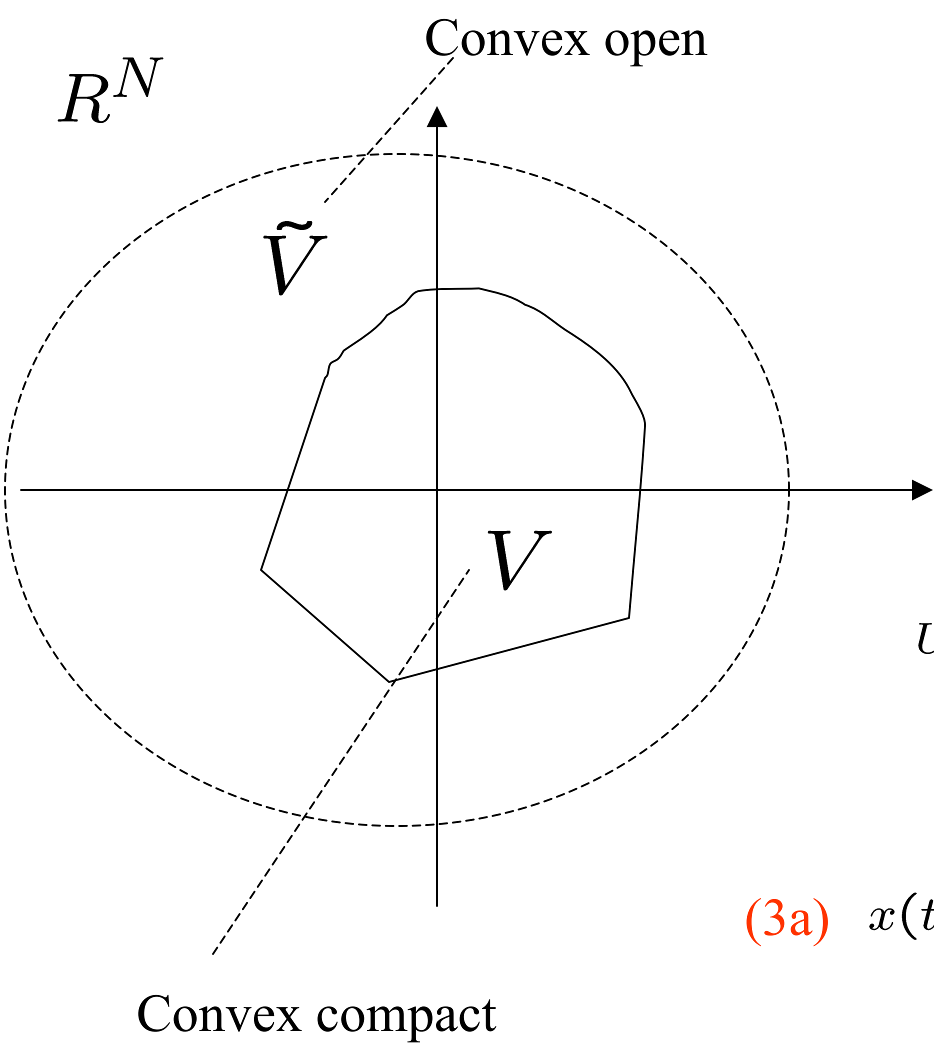
Step (1) $\quad x(t) \to V$

(*Same as for Gradient alg.* [S'05])

Step (2) $\quad q(t)$ is bounded

Use $U(x(t)) - \frac{1}{2} q(t) \cdot q(t)$

Proof outline



Dynamic system: $(x(t), q(t)), t \geq 0,$

$$x'(t) = v(t) - x(t)$$

$$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$$

$q'(t) = v(t)$, and $q(t)$ must stay in R_+^N

Step (3) $q^* \in Q^*$ is fixed.

If $x(t) \in V$, then

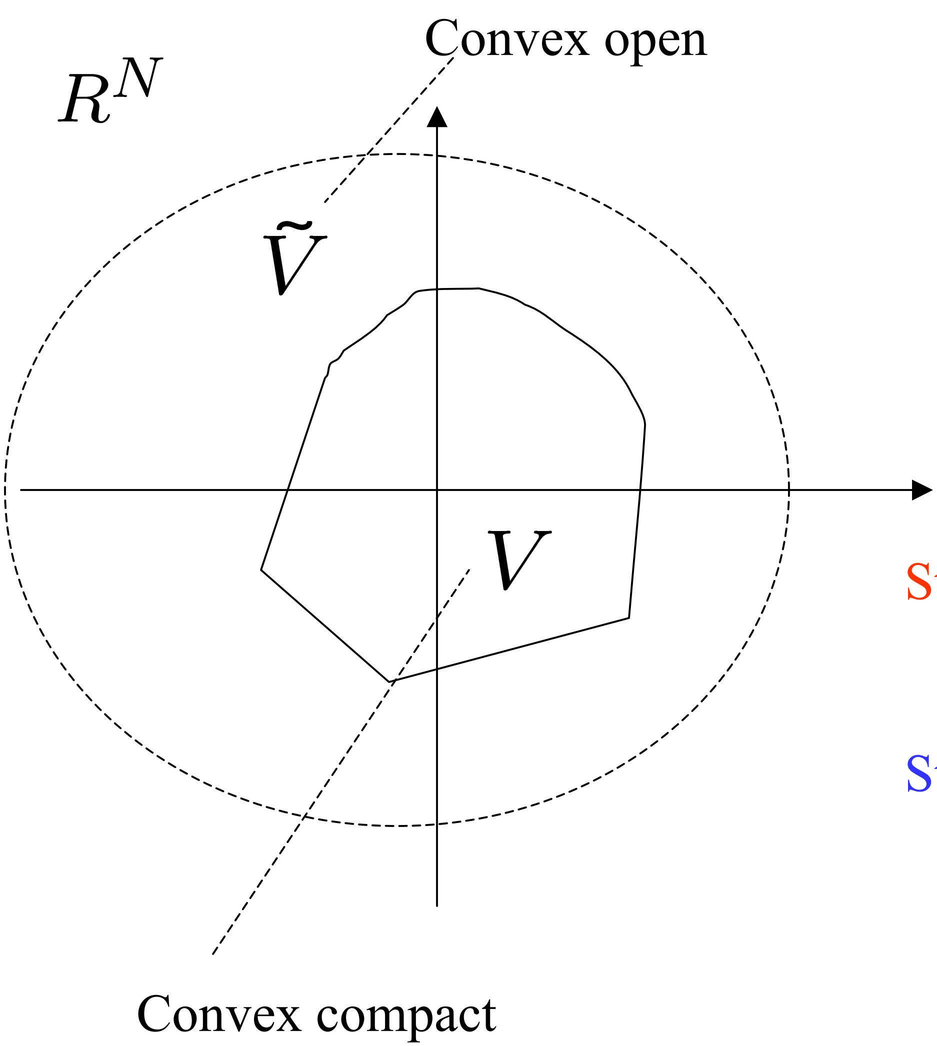
$$U(x(t)) - q^* \cdot x(t) - \frac{1}{2}[q(t) - q^*] \cdot [q(t) - q^*]$$

is non-decreasing.

(3a) $x(t) \to V^{**} = \arg\max_{x \in V} [U(x) - q^* \cdot x] \supseteq V^*$

(3b) $q(t) \to q^{**} \in Q^*$

Proof outline



Dynamic system: $(x(t), q(t)), t \geq 0,$

$$x'(t) = v(t) - x(t)$$

$$v(t) \in \arg\max_{v \in V} [\nabla U(x(t)) - q(t)] \cdot v$$

$$q'(t) = v(t), \text{ and } q(t) \text{ must stay in } R^N_+$$

Step (4) $\quad (3b) \;\Rightarrow\; x(t) \to R^N_-$

Step (5) If $x(t) \in V \cap R^N_-$, then

$$U(x(t)) - \frac{1}{2} q(t) \cdot q(t) \text{ non-decreasing}$$

Step (6) $\quad (3b), (5) \;\Rightarrow\; x(t) \to V^*$

Paper

- "Maximizing Queueing Network Utility subject to Stability: Greedy-Primal Dual Algorithm," *Queueing Systems*, 2005, Vol. 50, No.4, pp.401-457. http://cm.bell-labs.com/cm/ms/who/stolyar/pub.html

Related parallel work

- Erylmaz-Srikant, *INFOCOM'2005*.
- Lin-Shroff, *INFOCOM'2005*.
- Neely-Modiano-Li, *INFOCOM'2005*.

Network congestion control;
strictly concave increasing traffic source utility functions U_i *;*
$U = \Sigma U_i$ *;*
dual algorithms

Conclusions

- GPD = Naïve combination of Gradient and MaxWeight algorithms
- Applies to a wide range of models
- Provably (asymptotically) optimal
- Quite simple and often easy to implement
- Can be used in many cases where standard primal-dual algorithms (e.g., Arrow-Hurwicz-Uzawa) are not implementable