Greedy Primal-Dual Algorithm for Dynamic Resource Allocation in Complex Networks

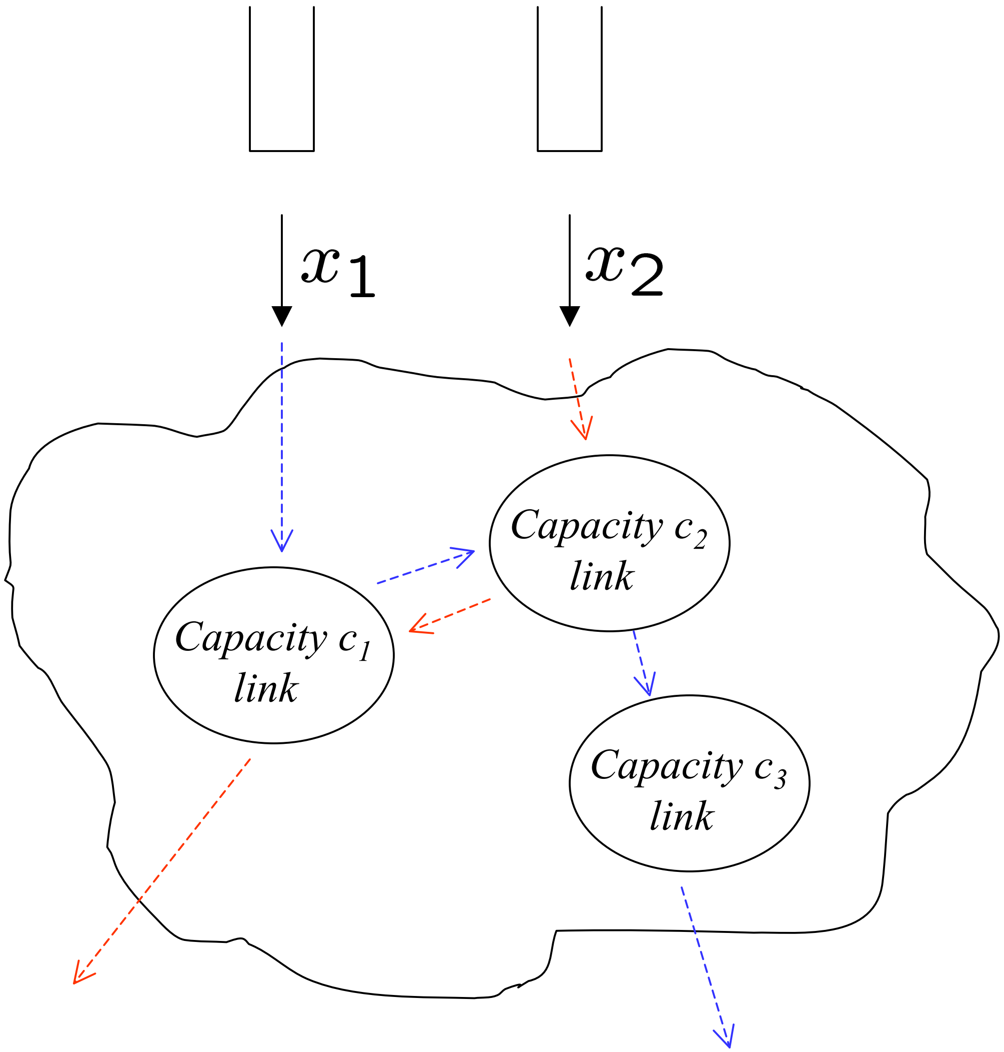
A.L.Stolyar
Bell Labs, Lucent Technologies

June 19, 2006

Outline

- GPD algorithm for the problem of maximizing queueing network utility s.t. stability
- Need for a more general algorithm, with additional convex constraints
- Generalized model and GPD algorithm
- Results
- Comparison to Arrow-Hurwicz-Uzawa primal-dual algorithm
- Key points of the analysis

Distributed congestion control of a "network of links"



Basic problem:

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

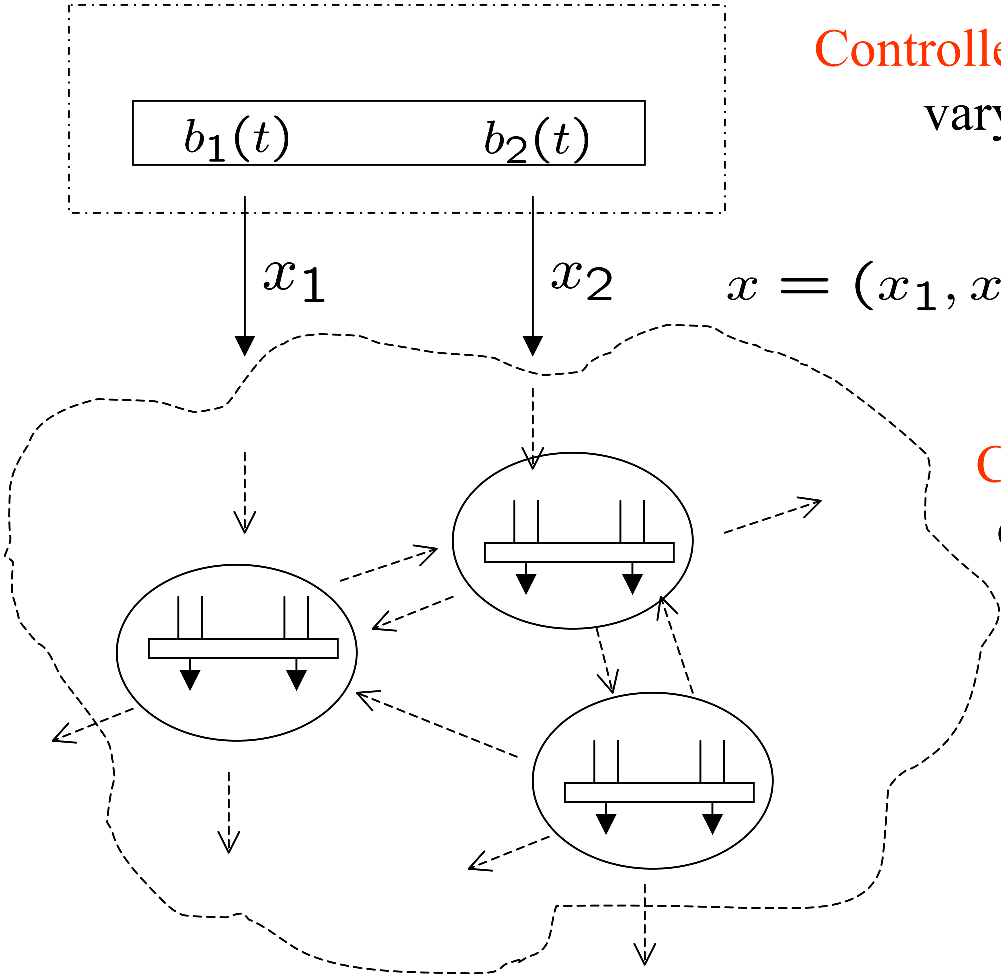
subject to

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
- Large and active field: F.Kelly et al., S.Low et al., …

Problem of maximizing "flow" utility subject to stability of a "complex time-varying" network: GPD algorithm



Controlled constrained interdependent time-varying traffic (commodity) sources

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

Controlled queueing network of constrained interdependent time-varying servers

GPD algorithm (S'2005) solves:

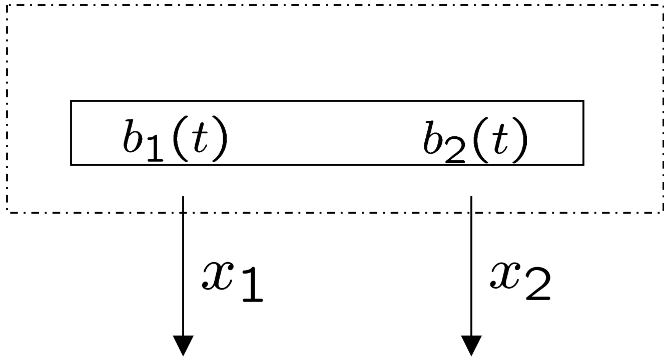
$$\max U(x)$$

concave smooth utility function

$$\text{s. t. queues are stable}$$

Related parallel work: Erylmaz-Srikant'2005, Lin-Shroff'2005, Neely-Modiano-Li'2005

Interdependent time-varying “flow” (commodity) sources



Discrete time

Mode (state) *m* of the aggregate source is random, follows ergodic finite-state Markov chain

Depending on current mode *m,* there is a finite set of choices (control decisions) for the commodity amounts (b_1,b_2) to be “produced”

Mode $m=1$: (b_1,b_2) can be chosen to be (3,5) or (0,6) or (12,0)

Mode $m=2$: (b_1,b_2) can be chosen to be (7,7) or (2,19) or (1,35) or (0,50)

$x=(x_1,x_2)$ = Average rates at which commodities are produced, under a given strategy;

$x \in$ convex compact region *V* – not given/known explicitly to network controller

GPD algorithm interpretation

$$X(t+1) = X(t)+\beta[b(t)-X(t)], \quad \beta > 0 \text{ small parameter}$$

GPD rule interpretation: Among currently available control actions (including choice of *b*), "greedily" choose control action maximizing expected increment of

$$F(X(t),Q(t)) = U(X(t)) - \frac{1}{2}\beta \sum_j Q_j(t)^2$$

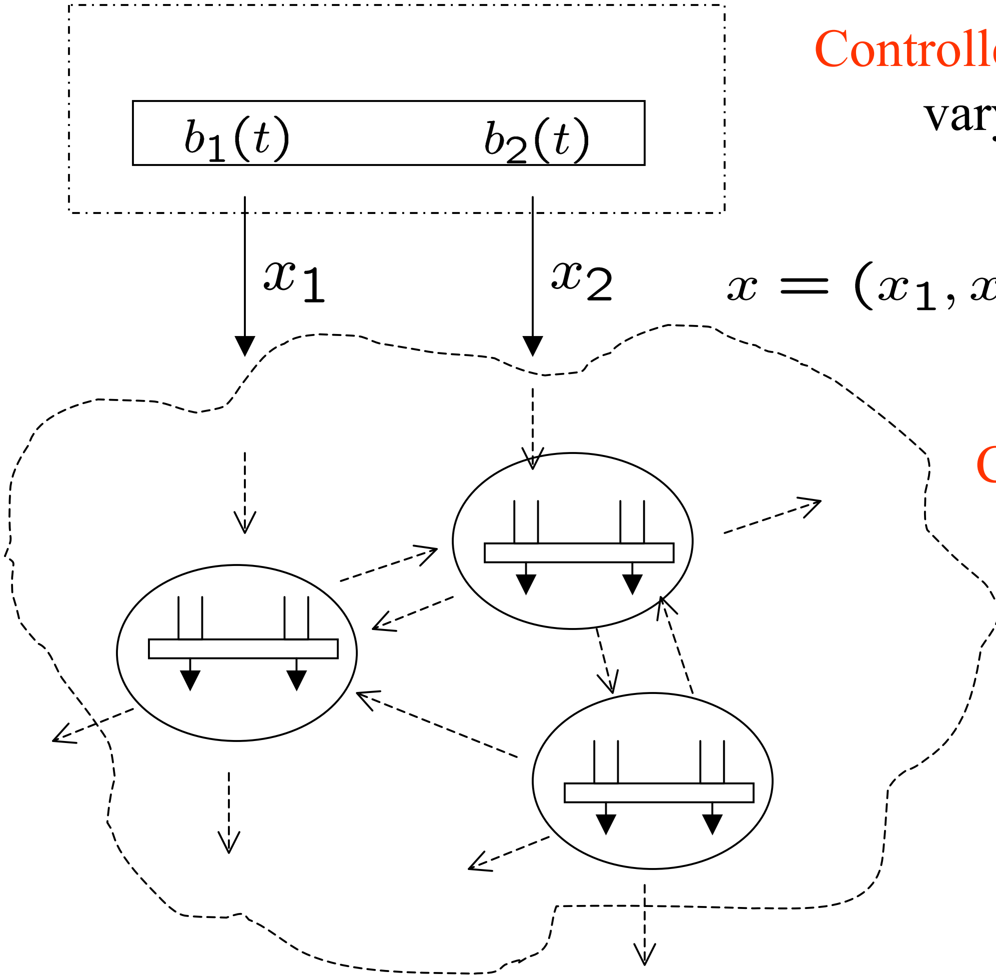
GPD is (close to) optimal when β is small

"No network" case: GPD => *"Gradient"* alg. (Agrawal-Subramanian'2002, S'2002), *U(X(t))* is "almost" Lyapunov function

"No utility" case: GPD => *"MaxWeight"* alg. (orig. Tassiulas-Ephremides'92), $\sum Q_j(t)^2$ is Lyapunov function

GPD may be viewed is a "naïve" combination of Gradient and MaxWeight. Optimality is harder to prove, because *F(X(t),Q(t))* is NOT a Lyapunov function

Goal: Generalization of GPD for the case of additional smooth convex constraints



Controlled constrained interdependent time-varying traffic (commodity) sources

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

Controlled queueing network of constrained interdependent time-varying servers

GPD Algorithm (S'2005) solves:

$$\max U(x)$$

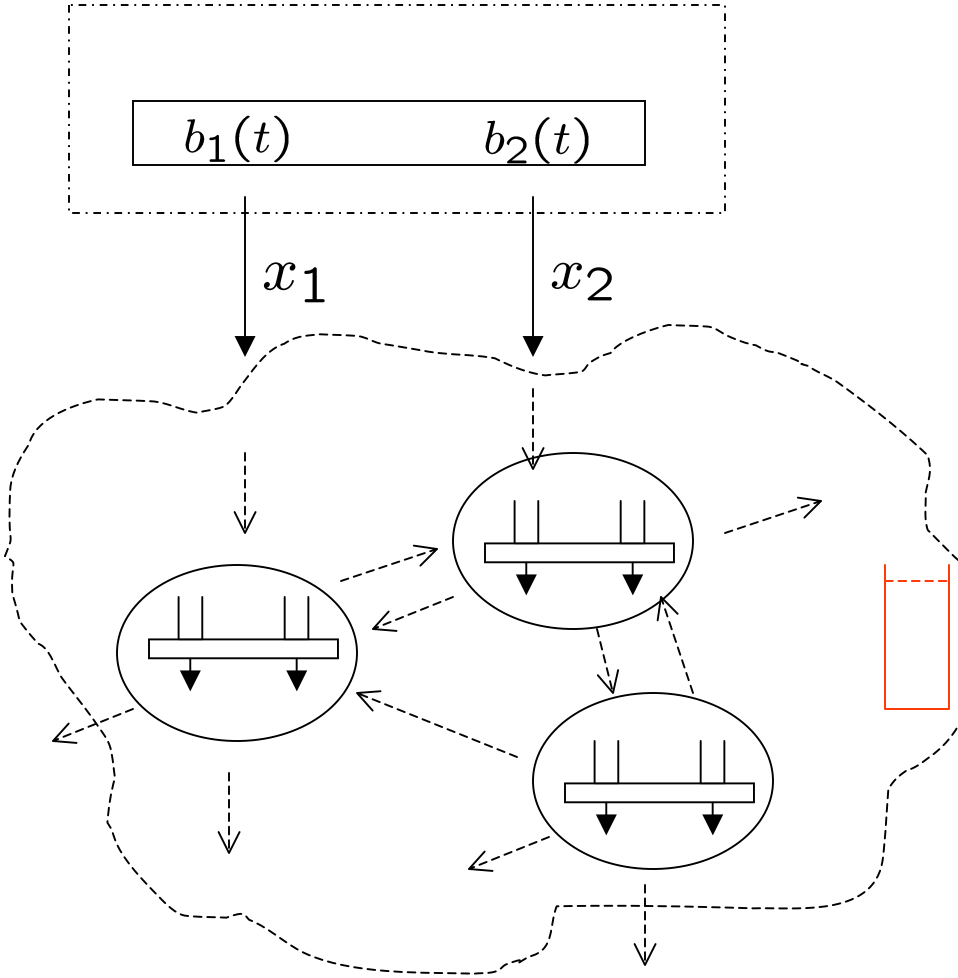
concave smooth utility function

s. t. queues are stable

$$\text{s.t. } G_j(x) \leq 0, \;\; j \in \mathcal{J}$$

Convex smooth constraint functions

Additional linear constraints ⇔ Stability of addl. virtual queues



$$\max U(x)$$

s. t. queues are stable

s.t. $c_1x_1 + c_2x_2 + c \leq 0$

$$\Delta Q(t) = c_1b_1(t) + c_2b_2(t) + c$$

Goal: Generalization of GPD for the case of additional smooth convex constraints

Network control problem:

$$\max U(x)$$

~~s.t. queues are stable~~

$$\text{s.t. } G_j(x) \leq 0, \ \ j \in \mathcal{J}$$

IN THIS TALK

Underlying convex optimization problem:

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

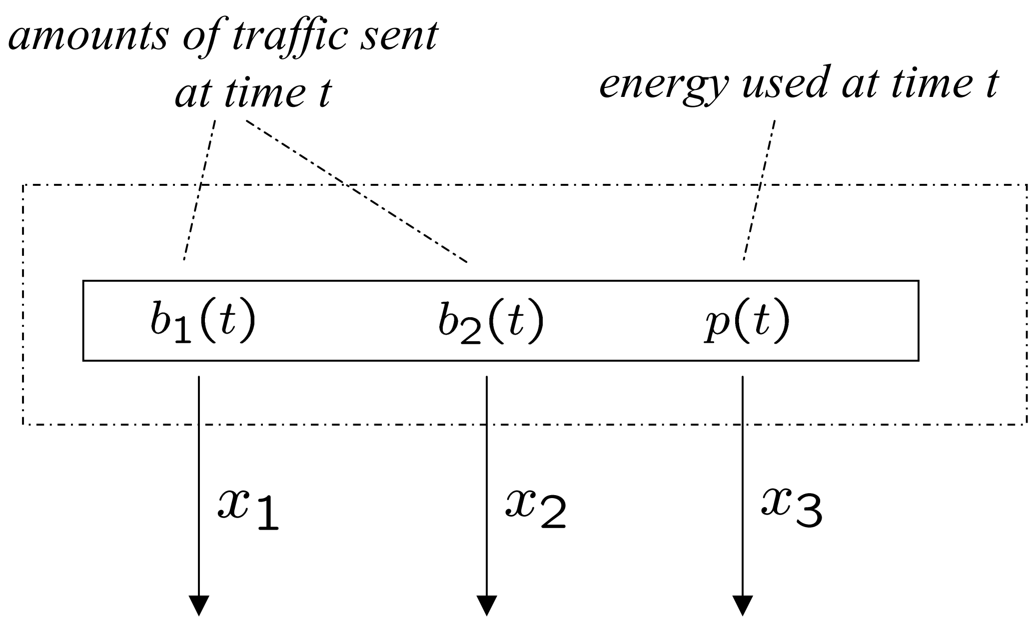
convex compact region

~~s.t. set of linear constraints~~

$$\text{s.t. } G_j(x) \leq 0, \ \ j \in \mathcal{J}$$

Motivation for convex constrains:
1. Min average power usage subject to utility constraint

Data traffic sent to two wireless users over a shared time-varying channel



$$\max \quad \log(x_1) + \log(x_2)$$

$$\text{s. t.} \quad x_3 \le c$$

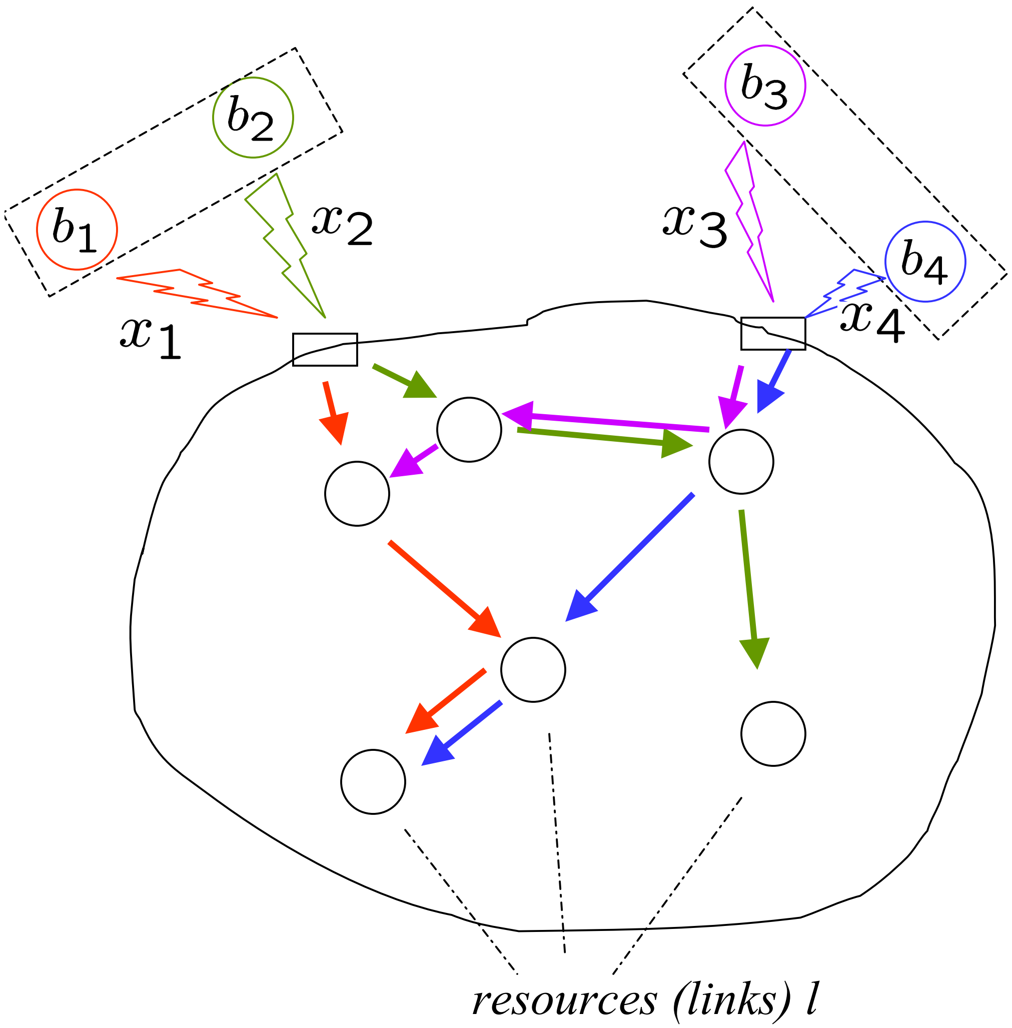
CAN BE SOLVED BY GPD ALGORITHM

NEEDS GENERALIZATION OF GPD ALGORITHM

$$\min x_3$$

$$\text{s. t.} \quad \log(x_1) + \log(x_2) \ge c$$

Motivation for convex constrains :
2. Max utility s.t. average end-to-end delay bounds



$$\max \sum_{n} U_n(x_n)$$

s. t.

$$\sum_{\ell \in \mathcal{R}(n)} C_\ell(x^{(\ell)}) \leq d_n$$

$$x^{(\ell)} \doteq \sum_{n \in \mathcal{R}^{-1}(\ell)} x_n$$

total traffic through link l

convex "congestion measure" (average delay) of link l

example: $$C_\ell(x^{(\ell)}) = \frac{1}{c^{(\ell)} - x^{(\ell)}}$$

Model (without the queueing network part)

Discrete time *t=0,1,2,…*

At any *t* we can choose control *k* from a finite set *K(m(t))*,
m(t) is underlying random (ergodic) "system mode," finite set of modes.

Control *k* has associated vector of "commodities": $b(k) = (b_n(k), n \in \mathcal{N})$

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

Problem: Find control strategy solving

$$\max U(x)$$

$$\text{s.t. } G_j(x) \leq 0, \; j \in \mathcal{J}$$

[-U] and all G_j are continuously differentiable convex (possibly non-strictly convex)

Generalized GPD algorithm

Maintain dual variable ("virtual queue length") βQ_j for each constraint j

$$k(t) \in \arg\max_k \; [\nabla U(X(t)) - \sum_{j \in \mathcal{J}} \beta Q_j(t) \nabla G_j(X(t))] \cdot b(k)$$

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

$$Q_j(t+1) = [Q_j(t) + G_j(X(t)) + \nabla G_j(X(t)) \cdot (b(k(t)) - X(t))]^+$$

$\beta > 0$ small parameter

Interpretation: given these update rules for primals X and duals Q_j, "greedily" maximize the (first order) increment of

$$U(X) - (1/2)\beta \sum_j Q_j^2$$

The algorithm with a small fixed β is close to optimal ("becomes" optimal as β -> 0)

Comparison: Arrow-Hurwicz-Uzawa Gradient method

convex smooth $-U(x),\ G_j(x),\ j \in \mathcal{J},\ \ x \in R_+^N$

$$\max\ U(x) \ \text{ s.t. } \ G_j(x) \leq 0,\ \forall j$$

$$X(t+1) = \left[X(t) + \beta[\nabla U(X(t)) - \sum_{j \in \mathcal{J}} \beta Q_j(t) \nabla G_j(X(t))] \right]^+$$

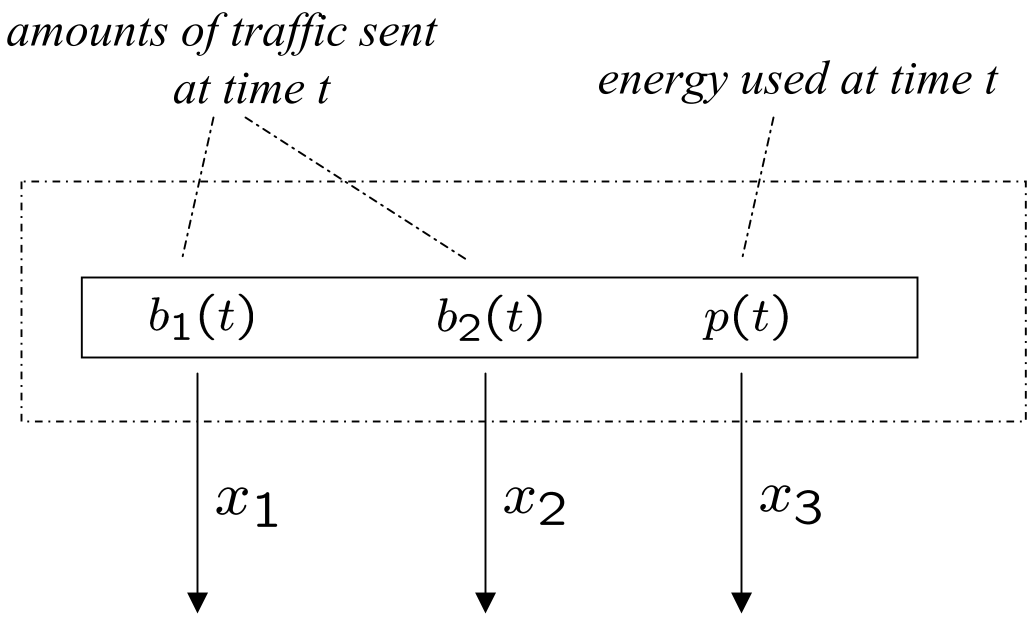
$$Q_j(t+1) = [Q_j(t) + G_j(X(t))]^+$$

Interpretation:

$$L(x,q) \doteq U(x) - \sum_j q_j G_j(x)$$

$$\Delta X = \beta \nabla_x L(X, \beta Q), \quad \Delta(\beta Q) = -\beta \nabla_q L(X, \beta Q)$$

Min average power usage subject to utility constraint



$$\min x_3$$

$$\text{s. t.} \quad U(x_1, x_2) \geq c$$

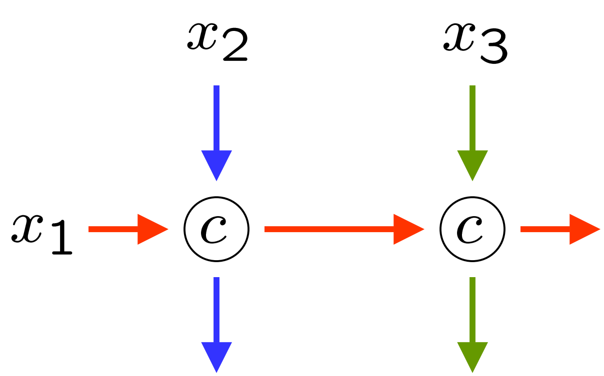
$$X = (X_1, X_2), \qquad b = (b_1, b_2)$$

$$k(t) \in \arg\max_k \; -p(k) + \beta Q(t) \nabla U(X(t)) \cdot b(k)$$

$$Q(t+1) = [Q(t) - U(X(t)) - \nabla U(X(t)) \cdot (b(k(t)) - X(t)) + c]^+$$

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

Max utility subject to mean end-to-end delays:
Naïve algorithms cannot possibly work



$$\max \sum_n \log x_n$$

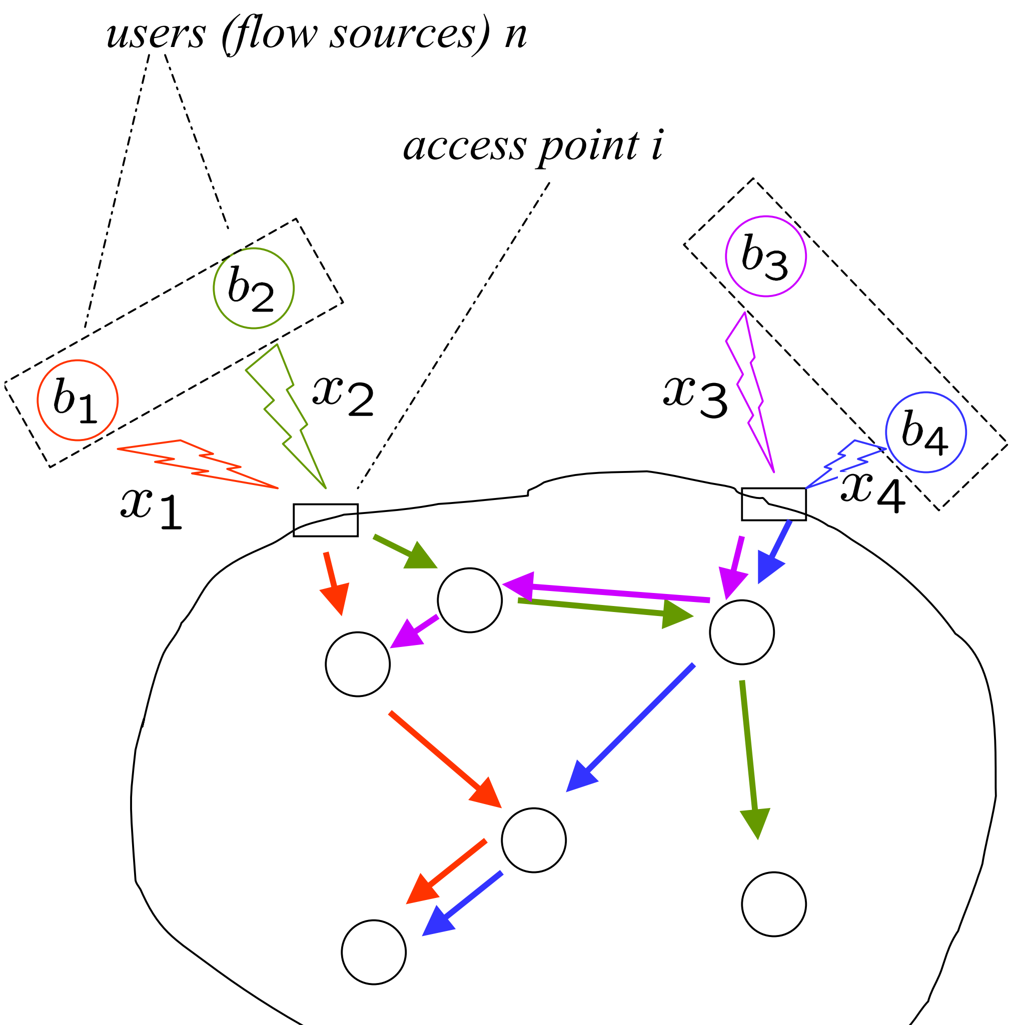
$$D_1 = \frac{1}{c-(x_1+x_2)} + \frac{1}{c-(x_1+x_3)} \leq d$$

$$D_2 = \frac{1}{c-(x_1+x_2)} \leq d$$

$$D_3 = \frac{1}{c-(x_1+x_3)} \leq d$$

An algorithm increasing x_n as long as $D_n<d$ cannot be optimal

Max utility subject to mean end-to-end delays



$$\max \sum_n U_n(x_n)$$

$$\sum_{\ell \in \mathcal{R}(n)} C_\ell(x^{(\ell)}) \le d_n, \quad \forall n$$

access point i scheduling decisions

$$k_i(t) \in \arg\max_{k_i \in K_i} \sum_{n \in \mathcal{N}_i} b_n(k_i) \left[U_n'(X_n(t)) - \beta W_{1,n}(t) \right]$$

maintained by access points

$$X_n(t+1) = \beta b_n(t) + (1-\beta) X_n(t)$$

$$Q_n(t+1) = \left[Q_n(t) + W_{2,n}(t) - d_n \right]^+$$

"collected" by messages along flow routes

$$W_{1,n}(t) \doteq \sum_{\ell \in \mathcal{R}(n)} W_1^{(\ell)}(t)$$

$$W_{2,n}(t) \doteq \sum_{\ell \in \mathcal{R}(n)} W_2^{(\ell)}(t)$$

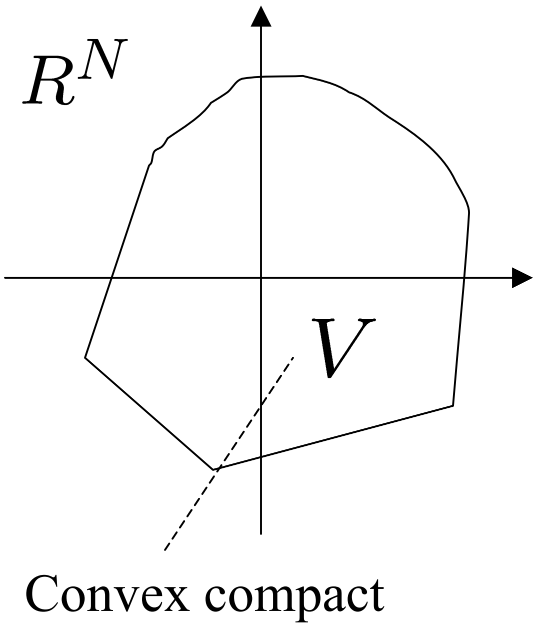
$$X^{(\ell)}(t+1) = \beta b^{(\ell)}(t) + (1-\beta) X^{(\ell)}(t)$$

$$W_1^{(\ell)}(t) \doteq Q^{(\ell)}(t) C_\ell'(X^{(\ell)}(t))$$

$$W_2^{(\ell)}(t) \doteq C_\ell(X^{(\ell)}(t)) + C_\ell'(X^{(\ell)}(t))(b^{(\ell)}(t) - X^{(\ell)}(t))$$

aggregates maintained by links

Underlying convex optimization problem



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

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

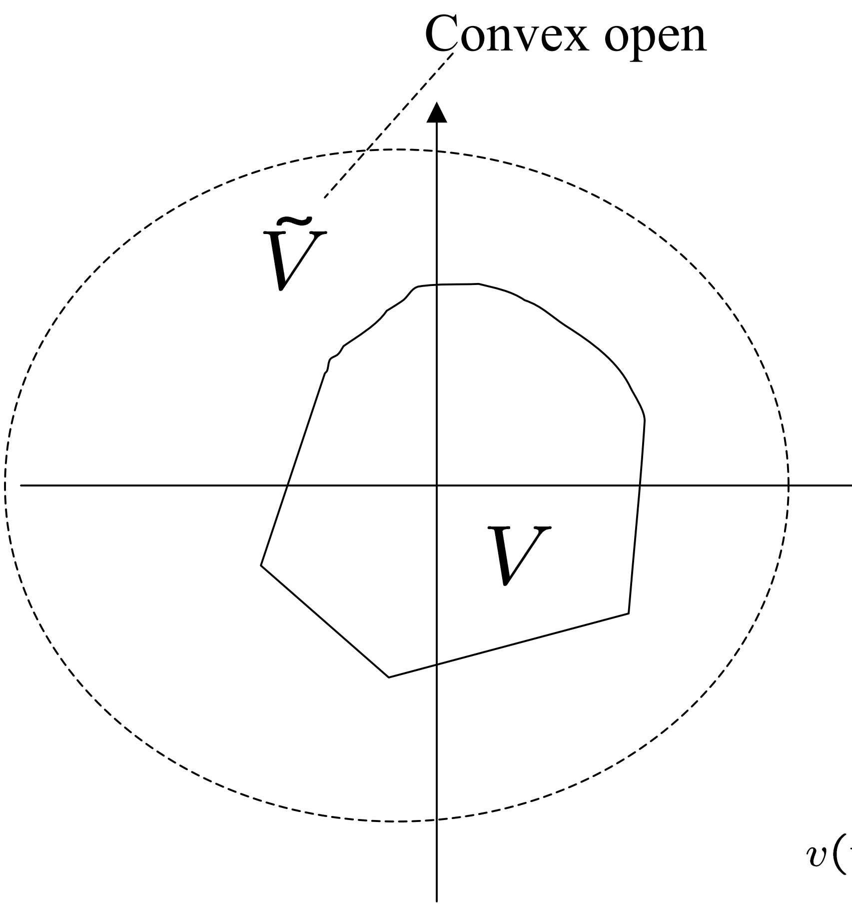
subject to

$$G_j(x) \leq 0, \ \ j \in \mathcal{J}$$

V^* optimal set

Q^* optimal set for the dual

Convergence to a greedy primal-dual dynamic system



$$-U(x),\ G_j(x),\ \forall j$$

are convex cont. diff. for $x \in \tilde{V}$

<u>THEOREM 1</u>:

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

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

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

such that

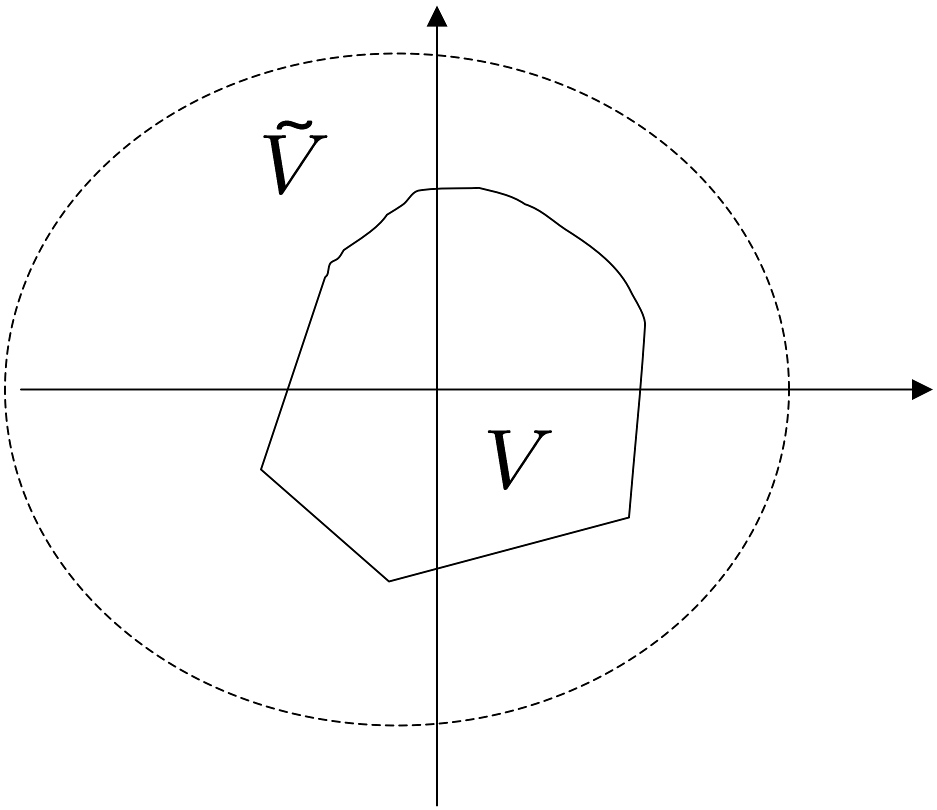
$$x'(t) = v(t) - x(t)$$

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

$$q_j'(t) = G_j(x(t)) + \nabla G_j(x(t)) \cdot (v(t) - x(t)),$$

and each $q_j(t)$ is reflected at 0

Main result: attraction property of the dynamic system



THEOREM 2:

Suppose $\exists v \in V,\ G_j(v) < 0\ \forall j$.

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

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

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

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

System dynamics: Arrow-Hurwicz-Uzawa Vs GPD

AHU	GPD

Domain of x

AHU	GPD
Defined explicitly (positive orthant)	Compact set V, implicitly defined by model structure

Interpretation

AHU:

$$L(x,q) \doteq U(x) - \sum_j q_j G_j(x)$$

$$x' = \nabla_x L(x,q), \quad q' = -\nabla_q L(x,q)$$

GPD:

"Greedily" maximize the derivative of

$$U(x) - (1/2)\|q\|^2$$

given the equations ("update rules") for x and q

Lyapunov function(s)

AHU:

$x^* \in V^*,\ q^* \in Q^*$ are fixed.

$$\|x - x^*\|^2 + \|q - q^*\|^2$$

GPD:

$q^* \in Q^*$ is fixed.

KEY: $U(x) - \sum_j q_j^* G_j(x) - (1/2)\|q - q^*\|^2$

$$U(x) - (1/2)\|q\|^2$$

Conclusions

- Additional convex constraints naturally arise in many applications
- Generalized GPD algorithm is formulated as a dynamic network control mechanism, but can be viewed as a dynamic algorithm for general convex optimization problems (including those implicitly given by network structure)
  - Specializes to parsimonious distributed algorithms
  - Can be used in many cases where standard primal-dual algorithms (e.g., Arrow-Hurwicz-Uzawa) are not implementable

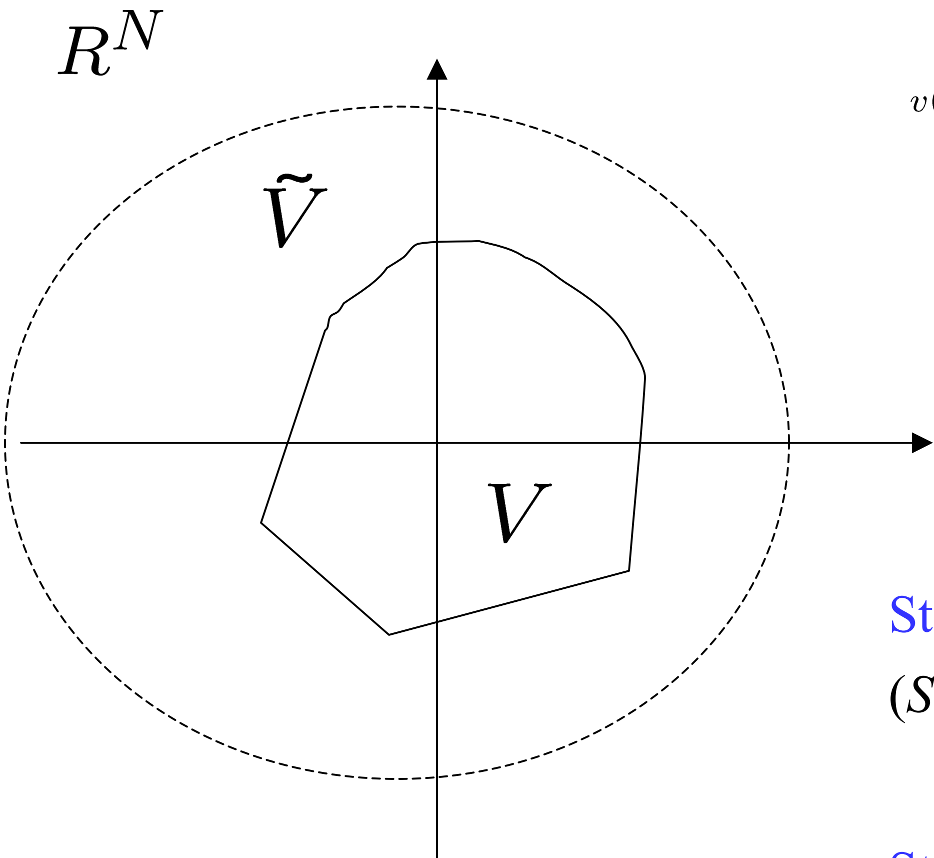
Papers

- A.L.Stolyar, "Maximizing Queueing Network Utility subject to Stability: Greedy-Primal Dual Algorithm," *Queueing Systems*, 2005.

GPD with additional smooth convex constraints:

- A.L.Stolyar,"Greedy-Primal Dual Algorithm for Dynamic Resource Allocation in Complex Networks," 2005, submitted. (To appear.)

Proof outline



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

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

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

$$q_j'(t) = G_j(x(t)) + \nabla G_j(x(t)) \cdot [v(t) - x(t)]$$

each $q_j(t)$ reflected at 0

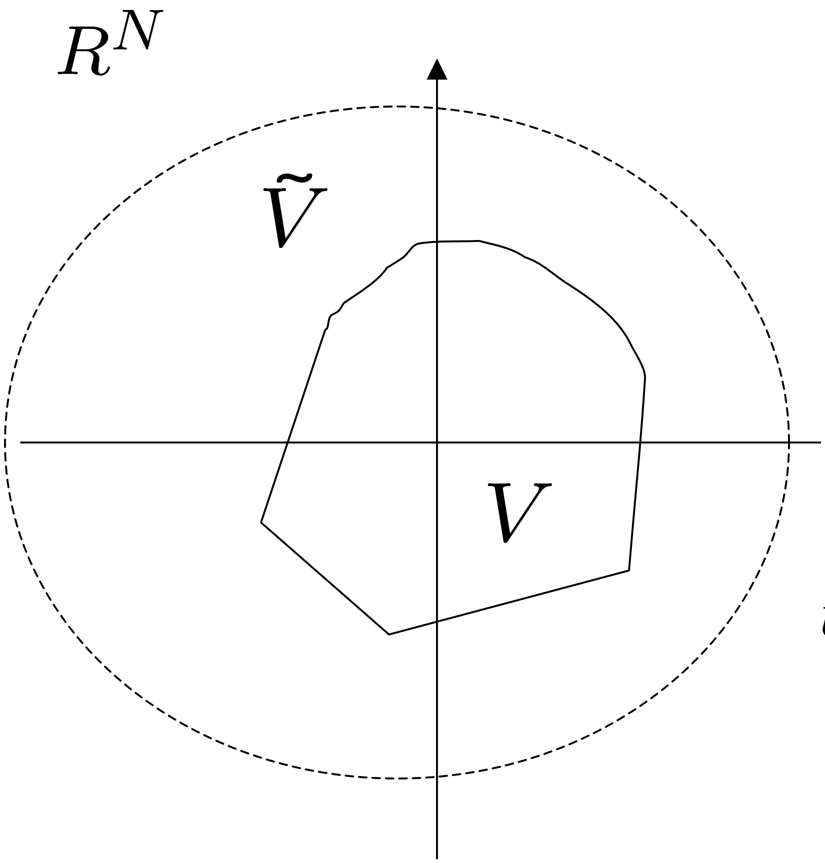
Step (1) $x(t) \to V$

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

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

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

Proof outline



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

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

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

$$q_j'(t) = G_j(x(t)) + \nabla G_j(x(t)) \cdot [v(t) - x(t)]$$

each $q_j(t)$ reflected at 0

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

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

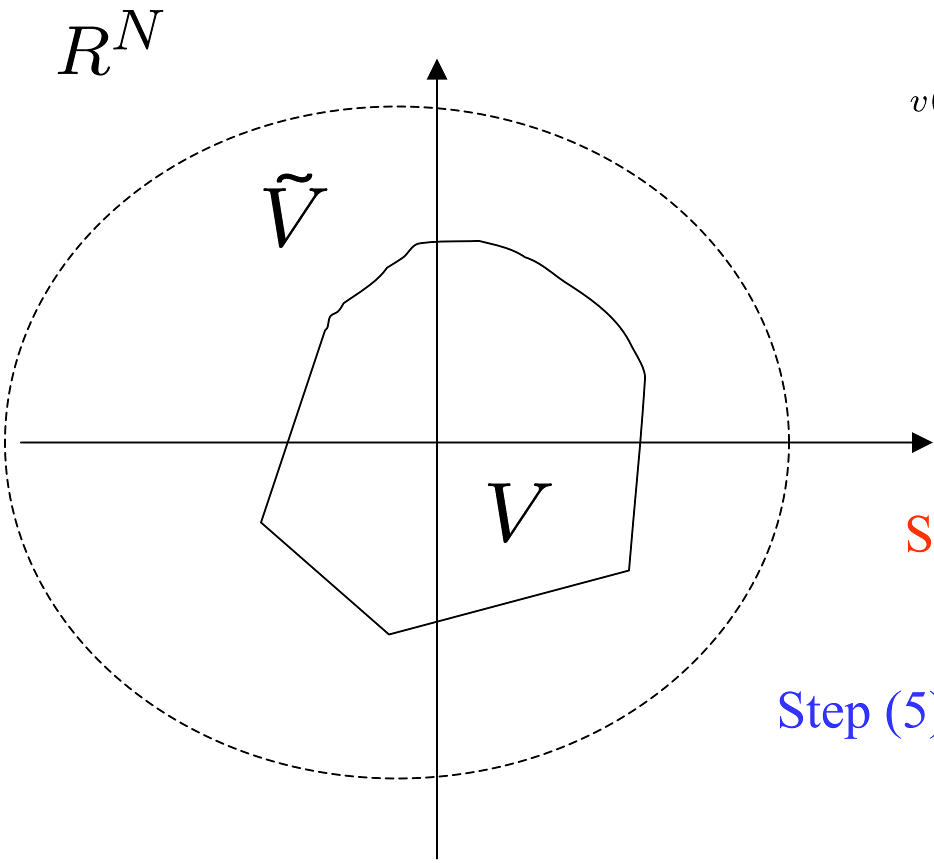
$$U(x(t)) - \sum_j q_j^* G_j(x(t)) - \frac{1}{2}[q(t) - q^*] \cdot [q(t) - q^*]$$

is non-decreasing.

(3a) $x(t) \rightarrow V^{**} = \arg\max_{x \in V} [U(x) - \sum_j q_j^* G_j(x)] \supseteq V^*$

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

Proof outline



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

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

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

$$q_j'(t) = G_j(x(t)) + \nabla G_j(x(t)) \cdot [v(t) - x(t)]$$

each $q_j(t)$ reflected at 0

Step (4) $\quad (3b) \;\Rightarrow\; \limsup G_j(x(t)) \leq 0$

Step (5) $\quad$ If $x(t) \in V \cap [\cap_j \{G_j(v) \leq 0\}]$, then

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

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