Primal-Dual Algorithm for Utility Maximization

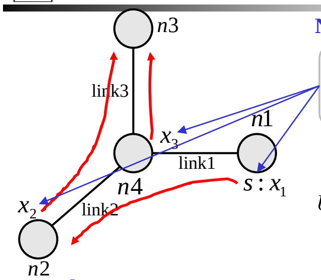
Yosuke Hoshi
FL 10 – 06 – 01
7th,June,2010

Outline

Network Optimization and Control
Srinivas Shakkottai and R. Srikant
Foundations and Trends in Networking,
Vol.2,No.3,2007.

- Background
- Primal Algorithm
- Dual Algorithm
- Primal-Dual Algorithm
- Power network

Background - Network



Network

s : traffic source
each source is associated with a route *r*

x_r : rate of source route *r*'s transmission

$U_r(x_r)$: the utility that the source obtains from transmitting data on route *r* at rate x_r

example

Utility $U_r(x_r) = \omega_r \log x_r$

Constraints $x_1 \le c_1$: link 1

$x_1 + x_2 \le c_2$: link 2

$x_2 + x_3 \le c_2$: link 3

Utility Maximization in Networks

Utility $\max_{x_r} \sum_{r \in S} U_r(x_r)$

Constraints $\sum_{r: l \in r} x_r \le c_l \quad \forall l \in L$

$x_r \ge 0 \quad \forall r \in S$

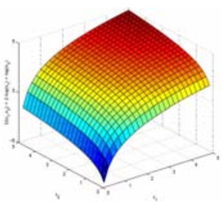
Background – Lagrangian

Concave function ⟺ Convex function

$f(tx + (1-t)y) \ge t f(x) + (1-t) f(y) \quad \alpha \in [0,1]$

Strictly concave function

$f(tx + (1-t)y) > t f(x) + (1-t) f(y) \quad \alpha \in (0,1)$



a strictly concave function has a unique maximum

Lagrangian

Lagrangian — Utility — constraints

$$L(x,\lambda) = \sum_{r \in S} U_r(x_r) - \sum_{r: l \in r} \lambda_r (x_r - c_r) \qquad \lambda_r \ge 0$$

strictly concave — strictly concave — convex

Background – Karush-Kuhn-Tucker heorem

example

min $f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 2)^2$ ...nvex

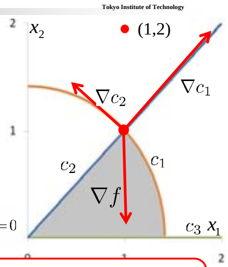
constraints $c_1(\mathbf{x}) = x_1^2 + x_2^2 - 2 \le 0$

$c_2(\mathbf{x}) = -x_1 + x_2 \le 0$

$c_3(\mathbf{x}) = -x_2 \le 0$

$\nabla f(\mathbf{x}^*) - u_1^* \nabla c_1(\mathbf{x}^*) - u_2^* \nabla c_2(\mathbf{x}^*) = 0$

$$\begin{pmatrix} 2(x_1^*-1) \\ 2(x_2^*-2) \end{pmatrix} + u_1^* \begin{pmatrix} 2x_1^* \\ 2x_2^* \end{pmatrix} + u_2^* \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 2 \\ 2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} = 0$$



Karush-Kuhn-Tucker theorem

$$L(x,\lambda) = \sum_{r \in S} U_r(x_r) - \sum_{r: l \in r} \lambda_r (x_r - c_r) \qquad \lambda_r \ge 0$$

the optimal value x^* and constraint's grad is Linear independence, then there exists constants λ_r such that

$$\partial L(x^*)/\partial x = 0 \qquad \lambda_r (x_r - c_r) = 0$$

Outline

- Background
- Primal Algorithm
- Dual Algorithm
- Primal-Dual Algorithm
- Power network

Primal Algorithm - example

Primal Algorithm - rate-update functions at source side

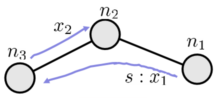
example

Utility $U_r(x_r) = \log x_r$

link Penalty $B_1(x_1) = x_1^2$: link 1

$B_2(x_1 + x_2) = (x_1 + x_2)^2$: link 2

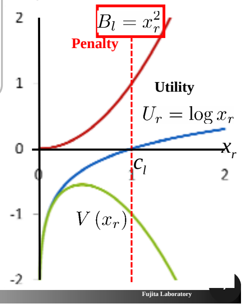
not to exceed link capacity c_l



$$V(x_r) = \log x_1 + \log x_2 \quad \text{:Utility}$$

$$-\left\{x_1^2 + (x_1 + x_2)^2\right\} \quad \text{:Penalty}$$

We want to maximize $V(x_r)$



Primal Algorithm - example

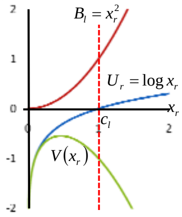
$$V(x_r) = \boxed{\log x_1 + \log x_2} - \boxed{\left\{x_1^2 + (x_1 + x_2)^2\right\}}$$

strictly concave — strictly concave — convex

when $V(x_r)$ is maximum at point x_1^* and x_2^*,

$$V_1'(x_1^*) = 1/x_1^* - (4x_1^* + 2x_2^*) = 0$$

$$V_2'(x_2^*) = 1/x_2^* - 2(x_1^* + x_2^*) = 0$$

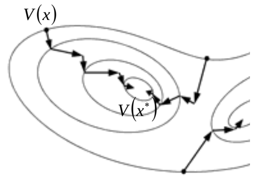


Primal algorithm

$$\dot{x}_1 = k_1(x_1)\{1/x_1 - (4x_1 + 2x_2)\}$$

$$\dot{x}_2 = k_2(x_2)\{1/x_2 - 2(x_1 + x_2)\}$$

$k_r(x_r)$: non-negative,increasing and continuous



Primal Algorithm - example

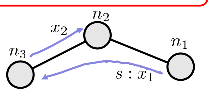
Primal algorithm

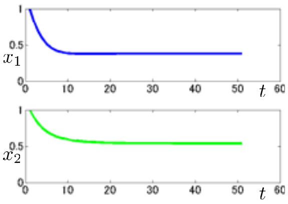
$$\dot{x}_1 = k_1(x_1)\{1/x_1 - (4x_1 + 2x_2)\}$$

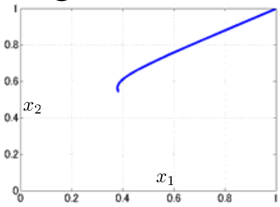
$$\dot{x}_2 = k_2(x_2)\{1/x_2 - 2(x_1 + x_2)\}$$

$k_1(x_1) = 0.01$
$k_2(x_2) = 0.01$

$x_1 = 0.3827$
$x_2 = 0.5412$







Primal Algorithm – Penalty function

Network utility maximization problem

$$V(x) = \sum_{r \in S} U_r(x_r) - \sum_{l \in L} B_l(y)$$

$U_r(x_r)$: utility (strictly concave)

$y = \sum_{s:l \in s} x_s$: the arrival rate on a link l

Penalty function

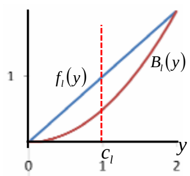
$B_l(.)$ increases when the arrival rate on a link l approaches the link capacity c_l

$$B_l(y) = \int_0^y f_l(y)\,dy$$

congestion price function

$f_l(y) = f_l\left(\sum_{s:l \in s} x_s\right)$: increasing, continuous function

B_l : convex function



Primal Algorithm

$$\boxed{\sum_{r \in S} U_r(x_r)} - \boxed{\sum_{l \in L} B_l(y)} = \boxed{V(x)} \qquad y = \sum_{s:l \in s} x_s$$

strictly concave — convex — strictly concave

the condition of miximization $V(x)$

$$V'(x_r) = U_r'(x_r) - \sum_{l:l \in r} f_l\left(\sum_{l:l \in s} x_s\right) = 0, \quad r \in \mathcal{S}$$

drive x towards the solution of $V(x)$

Primal algorithm

$$\dot{x}_r = k_r(x_r)\left\{U_r'(x_r) - \sum_{l:l \in r} f_l\left(\sum_{s:l \in s} x_s\right)\right\}$$

$k_r(x_r)$: non-negative,increasing and continuous

drive x_r towards the direction of ascent

Outline

- Background
- Primal Algorithm
- Dual Algorithm
- Primal-Dual Algorithm
- Power network

Dual Algorithm - example

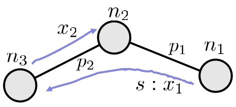
Dual Algorithm - price-update functions on each of links

example

Utility $U_r(x_r) = \omega_r \log x_r$

Link capacity $x_1 \le c_1$: link 1

$x_1 + x_2 \le c_2$: link 2



Lagrange dual

$$D(p) = \max_{\{x_r>0\}} (\omega_1 \log x_1 + \omega_2 \log x_2) - p_1(x_1 - c_1) - p_2(x_1 + x_2 - c_2)$$

constraints

p_l : Lagrange multiplier

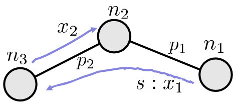
The Dual problem

$$\min_{p \ge 0} D(p)$$

Dual Algorithm - example

Lagrange dual

$$D(p) = \max_{\{x_r>0\}} (\omega_1 \log x_1 + \omega_2 \log x_2) - p_1(x_1 - c_1) - p_2(x_1 + x_2 - c_2)$$



The Dual problem $\min_{p \ge 0} D(p)$

calculate $\dfrac{\partial D}{\partial p_l}$

$$\frac{\partial D}{\partial p_1} = -(x_1 - c_1) \ge 0$$

$$\frac{\partial D}{\partial p_2} = -(x_1 + x_2 - c_2) \ge 0$$

We need to **descend down** to gradient $D(p)$!

Dual algorithm

$$\dot{p}_1 = (x_1 - c_1)^+_{p_1}$$

$$\dot{p}_2 = (x_1 + x_2 - c_2)^+_{p_2}$$

$$x_r = \frac{\omega_r}{\sum_l p_l}$$

$$(y_l - c_l)^+_{p_l} = \begin{cases} y_l - c_l & p_l > 0 \\ \max(y_l - c_l, 0) & p_l = 0 \end{cases}$$

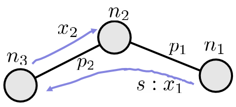
Dual Algorithm - example

Dual algorithm

$$\dot{p}_1 = (x_1 - c_1)^+_{p_1}$$

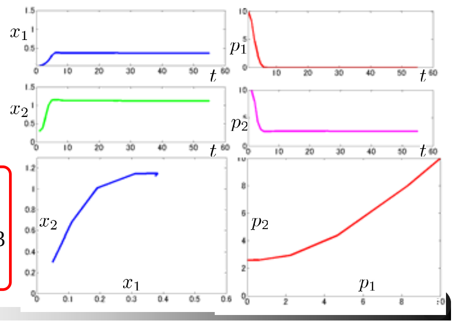
$$\dot{p}_2 = (x_1 + x_2 - c_2)^+_{p_2}$$

$$x_r = \frac{\omega_r}{\sum_l p_l}$$



$c_1 = 1.0$
$c_2 = 1.5$
$\omega_1 = 1.0$
$\omega_2 = 3.0$

$x_1 =$ 0.3750
$x_2 =$ 1.1250
$p_1 =$ -1.2723e-013
$p_2 =$ 2.6667



Dual Algorithm

resource allocation problem

utility $\max_{x_r} \sum_{r \in S} U_r(x_r)$ constraints $\sum_{r:l \in r} x_r \le c_l \quad \forall l \in L$

$x_r \ge 0 \quad \forall r \in S$

Each link has a capacity

The Lagrange dual

$$D(p) = \max_{\{x_r>0\}} \sum_r U_r(x_r) - \sum_l p_l \left(\sum_{s:l \in s} x_s - c_l \right)$$

costraints

The Dual problem $\min_{p \ge 0} D(p)$

p_l : Lagrange multiplier

price

To find the direction of a gradient descent, we need to know $\dfrac{\partial D}{\partial p_l}$

Dual Algorithm

Before derive $\dfrac{\partial D}{\partial p_l}$, we calculate $\dfrac{\partial D}{\partial x_r}$ and $\dfrac{\partial U^*}{\partial p_l}$

$$D(p) = \max_{\{x_r>0\}} \sum_r U_r(x_r) - \sum_l p_l \left(\sum_{s:l \in s} x_s - c_l \right)$$

$$\frac{\partial D(p_r)}{\partial x_r} = U'(x_r) - q_r = 0 \quad U''(x_r) \frac{\partial x_r}{\partial p_l} = 1$$

$\because \ q_r = \sum_{l:l \in r} p_l$ The price of route r

$$\frac{\partial D}{\partial p_l} = \sum_{r:l \in r} \frac{U'(x_r)}{U''(x_r)} - (y_l - c_l) - \sum_l p_l \sum_{r:l \in r} \frac{1}{U''(x_r)}$$

$\because \ y_l = \sum_{s:l \in s} x_s$ The load on link l

$$= c_l - y_l > 0$$

We need to **descend down** to gradient $D(p)$!

Dual algorithm

$$\dot{p}_l = h_l (y_l - c_l)^+_{p_l} \quad h_l > 0$$

$$x_r = U_r'^{-1}(q_r)$$

$$(y_l - c_l)^+_{p_l} = \begin{cases} y_l - c_l & p_l > 0 \\ \max(y_l - c_l, 0) & p_l = 0 \end{cases}$$

p_l increases when the arrival rate is larger than the capacity

Outline

- Background
- Primal Algorithm
- Dual Algorithm
- Primal-Dual Algorithm
- Power network

Primal-Dual Algorithm

Tokyo Institute of Technology

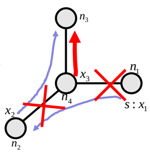
Primal
rate-update functions
at source side

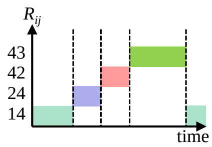
Dual
price-update functions
on each of links

Exact form of the update were different in the two cases!

in wireless network,
interference among various links necessitates scheduling of links

scheduling
R_{ij}: time of rate





Primal-Dual Algorithm

Primal-Dual Algorithm

Tokyo Institute of Technology

Primal-Dual Algorithm - flow base functions

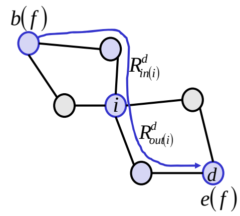
each flow has { a beginning node $b(f)$ / an ending node $e(f)$

$R^d_{in(i)}$: inflow rate (destination d at node i)

$$\sum_d R^d_{in(i)} = \sum_j R_{ji}$$

$R^d_{out(i)}$: outflow rate (destination d at node i)

$$\sum_d R^d_{out(i)} = \sum_k R_{ik}$$



the arrival rate is less than the departure rate

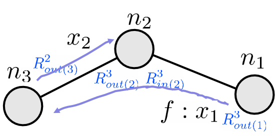
$$R^d_{in(i)} \le R^d_{out(i)}$$

Primal-Dual Algorithm example

Tokyo Institute of Technology

Utility $\quad U_r(x_r) = \omega_r \log x_r$

Constraints
1→3 $\quad x_1 \le R^3_{out(1)}$
3→2 $\quad x_2 \le R^2_{out(3)}$
2→3 $\quad R^3_{in(2)} \le R^3_{out(2)}$



Lagrange function

$$\max_{x,R\ge 0} \{\omega_1 \log x_1 + \omega_2 \log x_2 \quad \textbf{Utility}$$
$$-p_{13}\left(x_1 - R^3_{out(1)}\right)$$
$$-p_{32}\left(x_2 - R^2_{out(3)}\right) \quad \textbf{constraints}$$
$$-p_{23}\left(R^3_{in(2)} - R^3_{out(2)}\right)\}$$

Decoupled

Primal-Dual Algorithm example

Tokyo Institute of Technology

$$\max_{x\ge 0} \{\omega_1 \log x_1 + \omega_2 \log x_2 - p_{13}x_1 - p_{32}x_2\}$$

the congestion control problem

$$+\max_{R\ge 0} \left\{p_{13}R^3_{out(1)} + p_{32}R^2_{out(3)} + p_{23}\left(R^3_{out(2)} - R^3_{in(2)}\right)\right\}$$

the scheduling problem

Primal-Dual Algorithm

the primal algorithm
at each source f

$$\begin{cases} \dot{x}_1(t) = \omega_1/x_1 - p_{13} \\ \dot{x}_2(t) = \omega_2/x_2 - p_{32} \end{cases}$$

the dual algorithm
at each { node i / destination d

$$\begin{cases} \dot{p}_{13} = \left(x_1 - R^3_{out(1)}\right)^+_{p_{13}} \\ \dot{p}_{32} = \left(x_2 - R^2_{out(3)}\right)^+_{p_{32}} \\ \dot{p}_{23} = \left(R^3_{in(2)} - R^3_{out(2)}\right)^+_{p_{23}} \end{cases}$$

calculated at each time instant by solving the scheduling problem

Primal-Dual Algorithm

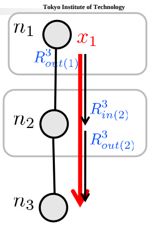
Tokyo Institute of Technology

the optimization problem

$$\max_{x_f,R\ge 0} \sum_f U_f(x_f)$$

each flow constraints

$i \to d \quad R^d_{in(i)} + \sum_{f:b(f)=i,e(f)=d} x_f \le R^d_{out(i)} \quad \forall \text{nodes } i, d \ne i$



Lagrange function

$$\max_{x_f,R\ge 0} \sum_f U_f(x_f) - \sum_i \sum_{d\ne i} p_{id}\left(R^d_{in(i)} + \sum_{f:b(f)=d} x_f - R^d_{out(i)}\right)$$

Decoupled

p_{id}: Lagrange multiplier

$$\max_{x_f\ge 0}\left\{\sum_f U_f(x_f) - \sum_f p_{b(f)i}x_f\right\} + \max_{R\ge 0}\left\{\sum_i \sum_{d\ne i} p_{id}\left(R^d_{in(i)} - R^d_{out(i)}\right)\right\}$$

x_f's terms $\qquad$ R's terms

Primal-Dual Algorithm

Tokyo Institute of Technology

Lagrange function

$$\max_{x_f\ge 0}\left\{\sum_f U_f(x_f) - \sum_f p_{b(f)i}x_f\right\} + \max_{R\ge 0}\left\{\sum_i \sum_{d\ne i} p_{id}\left(R^d_{in(i)} - R^d_{out(i)}\right)\right\}$$

the congestion control problem $\qquad$ **the scheduling problem**

Primal-Dual Algorithm to solve the congestion control problem

the primal algorithm
at each source f

$$\dot{x}_f(t) = U'_r(x_f) - p_{b(f)e(f)}$$

the dual algorithm
at each { node i / destination d

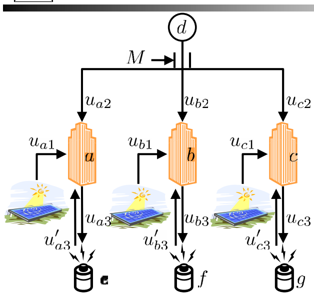
$$\dot{p}_{id} = \left\{\sum_{f:b(f)=i,e(f)=d} x_f(t) + R^d_{in(i)} - R^d_{out(i)}\right\}^+_{p_{id}}$$

calculated at each time instant by solving the scheduling problem

Outline

- Background
- Primal Algorithm
- Dual Algorithm
- Primal-Dual Algorithm
- Power network

Power network



purpose

to supply each building demands for energy r_i

constraints

$$u_{a2} + u_{b2} + u_{c2} + u_{a3} + u_{b3} + u_{c3} \leq M$$

battery

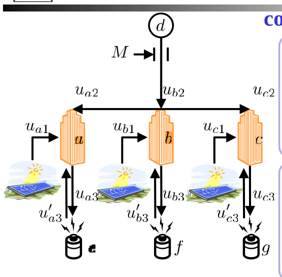
$$\dot{x}_i = -u'_{i3} + u'_{i3}, \quad |x_i| \leq m_i$$

Utility function

purpose **battery**

$$\min_{u_{i2}\geq 0, u_{i3}} \sum_{i=a}^{c} \left\{ q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right) + q_{2i} \exp\left(m_i - \int_{t_0}^{T} u_{i3} dt + \int_{t_0}^{T} u'_{i3} dt\right) \right\}$$

Power network



constraints

energy to building

$$u_{a2} + u_{b2} + u_{c2} + u_{a3} + u_{b3} + u_{c3} \leq M$$

line to charge battery

a→e	$R^e_{in(a)} \leq R^e_{out(a)}$
b→f	$R^f_{in(b)} \leq R^f_{out(b)}$
c→g	$R^g_{in(c)} \leq R^g_{out(c)}$

battery to building

e→a	$u'_{a3} \leq R^e_{out(a)}$
f→b	$u'_{b3} \leq R^b_{out(f)}$
g→c	$u'_{c3} \leq R^c_{out(g)}$

energy to charge battery

d→e	$u_{a3} \leq R^e_{out(d)}$
d→f	$u_{b3} \leq R^f_{out(d)}$
d→g	$u_{c3} \leq R^g_{out(d)}$

Lagrange function

$$\min_{u_{i2}\geq 0, u_{i3}} \sum_{i=a}^{c} \left\{ q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right) + q_{2i} \exp\left(m_i - \int_{t_0}^{T} u_{i3} dt + \int_{t_0}^{T} u'_{i3} dt\right) \right\}$$
$$- p_1\left(\sum_{i=a}^{c} u_{i2} + \sum_{i=a}^{c} u_{i3} - M\right) - \sum_{i=a,j=e}^{c,g} \left\{ p_{2i}\left(R^j_{in(i)} - R^j_{out(i)}\right) + p_{3i}\left(u'_{i3} - R^j_{out(j)}\right) \right\} - \sum_{i=a,j=e}^{c,g} p_{4i}\left(u_{i3} - R^j_{out(d)}\right)$$

Power network

$$\min_{u_{i2}\geq 0, u_{i3}} \sum_{i=a}^{c} \Big(q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right)$$
$$+ q_{2i} \exp\left(m_i - \int_{t_0}^{T} u_{i3} dt + \int_{t_0}^{T} u'_{i3} dt\right) - \left\{p_1(u_{i2} + u_{i3}) + p_{3i} u'_{i3} + p_{4i} u_{i3}\right\}\Big)$$

the congestion control problem

$$\min_{R} \sum_{i=a,j=e}^{c,g} \left\{ p_{2i}\left(R^j_{in(i)} - R^j_{out(i)}\right) + p_{3i} R^j_{out(j)} + p_{4i} R^j_{out(d)} \right\}$$

the scheduling problem

Primal-Dual Algorithm

the primal algorithm

$$\dot{u}_{i2} = q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right) - p_1$$
$$\dot{u}_{i3} = q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right) - p_1 - p_{4i} + q_{2i}\left(u_{i3}|_{t_0} - u_{i3}|_T\right) \exp\left(m_i - \int_{t_0}^{T} u_{i3} dt + \int_{t_0}^{T} u'_{i3} dt\right)$$
$$\dot{u}'_{i3} = q_{1i} \exp\left(u_{i1} + u_{i2} + u_{i3} - u'_{i3} - r_i\right) - p_{3i} + q_{2i}\left(u'_{i3}|_{t_0} - u'_{i3}|_T\right) \exp\left(m_i - \int_{t_0}^{T} u_{i3} dt + \int_{t_0}^{T} u'_{i3} dt\right)$$

the dual algorithm

$$\dot{p}_1 = \left(\sum_{i=a}^{c} u_{i2} + \sum_{i=a}^{c} u_{i2} - M\right)^+_{p_1}$$
$$\dot{p}_{2i} = \left(R^j_{in(i)} - R^j_{out(i)}\right)^+_{p_{2i}}$$
$$\dot{p}_{3i} = \left(u'_{i3} - R^i_{out(j)}\right)^+_{p_{3i}}$$
$$\dot{p}_{4i} = \left(u_{i3} - R^i_{out(j)}\right)^+_{p_{4i}}$$