Comparing Distributed Primal-Dual Algorithms for Discrete Variational Mechanics and Internet Congestion Control

Andrew Lamperski

June 3, 2005

Abstract

We review Lall and West's work connecting primal-dual algorithms used in variational mechanics with those used by TCP/AQM. We demonstrate that the algorithms follow the same basic structure. It is hoped that these methods might provide a basis for systematic construction and analysis of large and/or multi-scale systems.

Introduction As science moves toward tackling larger and more complex problems in both design and analysis, we immediately run into two problems. First, attempting to perform all computations on a single machine may be too slow for practicality. Second, and possibly more importantly, if we need large scale robust computations, then distributing the load over many computers may be a sensible alternative to relying on a single unit.

The following is primarily a review of work by Lall and West [2]. Their work developed the link between variational integration schemes and TCP/AQM (Transmission Control Protocol/Active Queue Management) protocols employed for internet congestion control. They found that both optimization problems could be solved by distributed primal dual algorithms.

While not much original work is presented here, we do make a few clarifications. Propositions 1 and 2 have simple converses that to our knowlege have not been presented before. Further, a distributed algorithm for solving the dual problem in discrete multisymplectic mechanics is presented. Both of these results, especially the multisymplectic dual algorithm are hinted at in Lall and West's work. In short, everything presented here is probably already well understood.

Duality First we give a brief introduction to the main definitions and facts from optimization that we will use in the finite dimensional problems.

Our finite dimensional treatment follows largely from Boyd and Vandenberge [1]. The common form for optimization problems over $\mathbb{R}^n$ is as follows

$$\begin{array}{l} \quad minimize \quad f_0(x) \quad s.t. \\ f_i(x) \leq 0 \quad \forall i = 1, \dots, m \\ h_i(x) = 0 \quad \forall i = 1, \dots, p. \end{array} \tag{1}$$

If x^* solves the optimization problem, that is, x^* minimizes f_0 subject to the constraints, we call x^* an optimal solution. We often refer to f_0 as the objective function. We call $p^* = f_0(x^*)$ the optimal value. We call the primal problem convex if f_i are all convex and $h_i(x) = a_i x - b_i$, (i.e. the equality constraints are affine).

We construct the Lagrangian for the optimization problem as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_i h_i(x). \tag{2}$$

This Lagrangian is related to the Lagrangian with constraints that we find in mechanics [6]. In mechanics however, we are not interested in extremizing the Lagrangian itself, but rather we want to extremize the action integral of the Lagrangian.

The dual function is

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) \tag{3}$$

where the infimum is taken over all x in the domain of the primal problem (i.e. x for which f_i and h_i are defined). The dual problem is

$$\begin{array}{l} maximize \quad g(\lambda, \nu) \quad s.t. \\ \lambda_i \geq 0 \quad \forall i = 1, \ldots, m. \end{array} \tag{4}$$

Similar to the case of the primal problem, if (λ^*, ν^*) maximizes g with $\lambda_i \geq 0$, we call (λ^*, ν^*) the optimal solution. We call $d^* = g(\lambda^*, \nu^*)$ the optimal value of the dual problem.

One should note that if x satisfies all the constraints (inequalities and equalities in (1)) and $\lambda_i \geq 0$, then the Lagrangian and hence the dual provide a lower bound on the value of $f_0(x)$.

We say an optimization problem has strong duality if the optimal value of the primal problem equals the optimal value of the dual problem. That is $p^* = d^*$.

The Karush-Kuhn-Tucker (KKT) conditions provide important conditions for when we have strong duality.

Fact 1 *Let x^* be a primal optimal solution, and let (λ^*, ν^*) be a dual optimal solution. If we have strong duality, $f_0(x^*) = g(\lambda^*, \nu^*)$, then*

$$\begin{array}{rcl} f_i(x^*) & \leq & 0, \quad i = 1, \ldots, m \\ h_i(x^*) & = & 0, \quad i = 1, \ldots, p \\ \lambda_i^* & \geq & 0, \quad i = 1, \ldots, m \\ \lambda_i^* f_i(x^*) & = & 0, \quad i = 1, \ldots, m \\ \nabla f_0(x^*) + \sum_{i=1}^m \nabla f_i(x^*) \lambda_i^* + \sum_{i=1}^p \nabla h_i(x^*) \nu_i^* & = & 0. \end{array} \tag{5}$$

Further, if the primal problem is convex (f_i convex and h_i affine), then any (x, λ, ν) satisfying the KKT conditions (5) are primal-dual optimal with strong duality.

The dual algorithms detailed below share the basic scheme:

- 1. Given dual variables (λ, ν), compute x to minimize the Lagrangian.
- 2. Now given x, perform gradient ascent on the dual function. In all cases, the gradient will just be the vector of constraints.

Discrete Mechanics We detail some basic notions in discrete mechanics and present a distributed variational integration scheme which can be implemented in a manner similar to the methods used in internet congestion control. The results here largely follow Lall and West's work [2].

In continuous mechanics, we study Lagrangians over the tangent bundle of some manifold, $L : TQ \to \mathbb{R}$. In discrete variational mechanics, we look at Lagrangians of the form $L_d : Q \times Q \to \mathbb{R}$. Instead of extremizing the action integral, in discrete mechanics, we extremize the action sum

$$S_d = \sum_{i=0}^{N-1} L_d(q(i), q(i+1)) \tag{6}$$

with endpoints fixed as $q(0) = q_0$ and $q(N) = q_N$. We can derive the discrete Euler-Lagrange (DEL) equations by setting the differential of the action sum equal to zero for all variations [5].

$$dS_d \cdot \delta q = \sum_{i=1}^{N-1} \left[D_2 L_d(q(i-1), q(i)) + D_1 L_d(q(i), q(i+1)) \right] \cdot \delta q(i) = 0. \tag{7}$$

Since this must be true for all variations we get the discrete Euler-Lagrange equations

$$D_2 L_d(q(i-1), q(i)) + D_1 L_d(q(i), q(i+1)) = 0 \tag{8}$$

for $i = 1, \cdots, N-1$.

We call a set of points on Q, $\{q(i)\}$ a trajectory. We say a trajectory satisfies the DEL equations if the trajectory has $N+1$ points, the DEL equations hold for $i = 1, \ldots, N-1$ and the endpoints the trajectory are fixed at $q(0) = q_0$ and $q(N) = q_N$.

We define the conjugate momenta [5] as

$$\begin{aligned} p^-_{k,k+1} &= -D_1 L_d(q_k, r_{k+1}) \\ p^+_{k-1,k} &= D_2 L_d(q_{k-1}, r_k). \end{aligned} \tag{9}$$

Note that the first and second arguments come from different trajectories in general.

Now we prove the simple result that makes the allows us to make distributed duality based algorithms for mechanical integration.

Proposition 1 *A trajectory* $\{q(i)\}_{i=0}^N$ *with endpoints* q_0 *and* q_N *satisfies the DEL equations iff* $p^+_{k-1,k} = p^-_{k,k+1}$ *for* $k = 1, \ldots, N-1$.

Proof:(⇒) Let $\{q(i)\}_{i=0}^N$ satisfy the DEL equations.Then for $k = 1, \ldots, N-1$

$$\begin{aligned} 0 &= D_2 L_d(q(k-1), q(k)) + D_1 L_d(q(k), q(k+1)) \\ &= p^+_{k-1,k} - p^-_{k,k+1}. \end{aligned} \tag{10}$$

Therefore, $p^+_{k-1,k} = p^-_{k,k+1}$ for $k = 1, \ldots, N-1$.

(⇐) Let $p^+_{k-1,k} = p^-_{k,k+1}$ for $k = 1, \ldots, N-1$. Then define

$$p_k := p^+_{k-1,k} = p^-_{k,k+1}. \tag{11}$$

Then the derivative with respect to q and r is given by

$$\begin{aligned} 0 &= \begin{bmatrix} D_1 L_d(q(0), r(1)) + p_0 \\ \vdots \\ D_1 L_d(q(N-1), r(N)) + p_{N-1} \\ D_2 L_d(q(0), r(1)) - p_1 \\ \vdots \\ D_2 L_d(q(N-1), r(N)) - p_N \end{bmatrix} \\ &= D_{(q,r)} \Big[\textstyle\sum_{i=0}^{N-1} L_d(q(i), r(i+1)) + \sum_{i=1}^{N-1} p_i(q(i) - r(i)) + \\ &\qquad p_0(q(0) - q_0) + p_n(r(N) - q_N) \Big] \end{aligned} \tag{12}$$

Thus, by the Lagrange Multiplier Theorem [6], $\{q(i)\}$ and $\{r(i)\}$ make a critcal point of

$$\sum_{i=0}^{N-1} L_d(q(i), r(i+1)) \tag{13}$$

subject to the constraints

$$\begin{aligned} q(i) &= r(i), \quad i = 1, \ldots, N-1 \\ q(0) &= q_0 \\ r(N) &= q_N. \end{aligned} \tag{14}$$

We define $q(N) := r(N)$. To see that the DEL equations are satisfied for $i = 1, \ldots, N-1$, add the terms in the matrix from (12) and substitute $q(i)$ for $r(i)$ to get $D_1 L_d(q(i), q(i+1)) + D_2 L_d(q(i-1), q(i)) = 0$. Exactly the DEL equations. Thus $\{q(i)\}_{i=1}^N$ is a trajectory satisfying the DEL equations ■

One immediate result of Proposition 1 is that if L_d is convex in both entries and p is defined as in the proof, we have shown that the KKT conditions hold and thus we have strong duality.

We can perform the variational integration using both "primal" and "dual" methods. This is really an abuse of terminology, since our primal method and dual method are actually based on different (though mathematically equivalent) optimization problems.

In the following algorithms, we assume that $L_d : \mathbb{R}^n \times \mathbb{R}^n$ is convex in both of its arguments. The optimization problem for the primal scheme is

$$\begin{aligned} \textit{minimize} \quad & \textstyle\sum_{i=0}^{N-1} L_d(q(i), q(i+1)) \quad \textit{s.t.} \\ & q(0) = q_0 \\ & q(N) = q_N. \end{aligned} \tag{15}$$

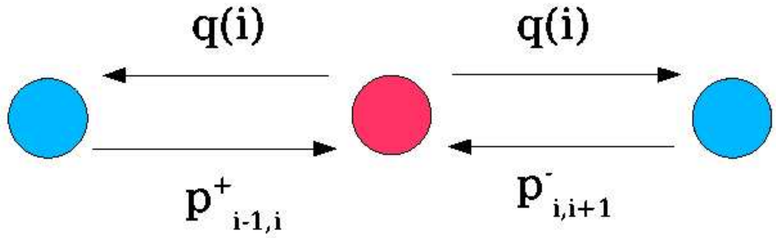

Figure 1: Example nodes in the primal scheme graph

The distributed algorithm is really nothing more than gradient descent on the objective function, which is the action integral. That is, for $i = 1, \ldots, N-1$, we update $q(i)$ as

$$q(i) := q(i) - \gamma(D_2 L_d(q(i-1), q(i)) + D_1 L_d(q(i), q(i+1))). \tag{16}$$

Here $\gamma > 0$ is some rate constant controlling how fast we perform the gradient descent.

We solve this as a distributed algorithm by setting up a graph of primal nodes and dual nodes (see Figure 1).

- 1. At the dual node i, given $q(i-1)$ and $q(i)$, we compute

$$\begin{array}{rcl} p^+_{i-1,i} & = & D_2 L_d(q(i-1), q(i)) \\ p^-_{i-1,i} & = & -D_1 L_d(q(i-1), q(i)). \end{array} \tag{17}$$

  Then dual node $i+1$ passes $p^-_{i,i+1}$ to primal node i and dual node i passes $p^+_{i-1,i}$ to primal node i.

- 2. Primal node i computes

$$q(i) := q(i) - \gamma(p^+_{i-1,i} - p^-_{i,i+1}). \tag{18}$$

  The primal node then passes $q(i)$ to dual nodes i and $i-1$.

The dual scheme solves the problem

$$\begin{array}{c} minimize \quad \sum_{i=0}^{N-1} L(q(i), r(i+1)) \quad s.t. \\ q(i) = r(i), \quad i = 1, \ldots, N-1 \\ q(0) = q_0 \\ r(N) = q_N. \end{array} \tag{19}$$

As mentioned in the Duality section, the dual algorithm involves finding q and r to minimize the Lagrangian

$$\begin{array}{rcl} K(q, r, p) & = & \sum_{i=0}^{N-1} L_d(q(i), r(i+1)) + \\ & & \sum_{i=1}^{N-1} p_i(q(i) - r(i)) + p_0(q(0) - q_0) + p_n(r(N) - q_N). \end{array} \tag{20}$$

Then use q and r to compute the gradient of the dual function. Note that if $q(i)$ and $r(i)$ are points in trajectories minimizing the Lagrangian, then the partial derivatives of the dual function ($g(p) = \inf_{(q,r)} K(q, r, p)$) are given as

$$\frac{\partial g}{\partial p_i} = q(i) - r(i). \tag{21}$$

We can solve the dual scheme on a graph as follows.

- 1. At primal node i, given p_{i-1} and p_i, find $q(i-1)$ and $r(i)$ satisfying

$$\begin{array}{rcl} p_{i-1} & = & -D_1 L_d(q(i-1), r(i)) \\ p_i & = & D_2 L_d(q(i-1), r(i)). \end{array} \tag{22}$$

  Then pass $r(i)$ to dual node i and pass $q(i-1)$ to dual node $i-1$.

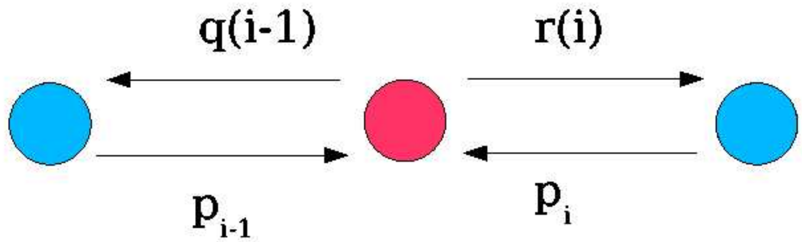

Figure 2: Nodes of the graph for the dual scheme.

- 2. At dual node i perform gradient ascent as

$$p_i := p_i + \gamma(q(i) - r(i)). \tag{23}$$

Solving the dual problem in this manner gives us q, r and p that satisfy the KKT conditions. By convexity of L_d, this implies that we find both the primal and dual optimal solutions.

Discrete Multisymplectic Mechanics The generalization of Proposition 1 and the two algorithms for integration are straightforward but notationally cumbersome. The main difference is that in the original case, the momenta were Lagrange multipliers associated with the constraint of holding the line elements of a trajectory together. In the multisymplectic case, the momenta correspond to Lagrange multipliers enforcing the constraint of holding the mesh elements of a spacetime discretization together (see Figures 3 and 4). Much of this section also follows from Lall and West's work [2]. Our treatment sacrafices precision for easier notation.

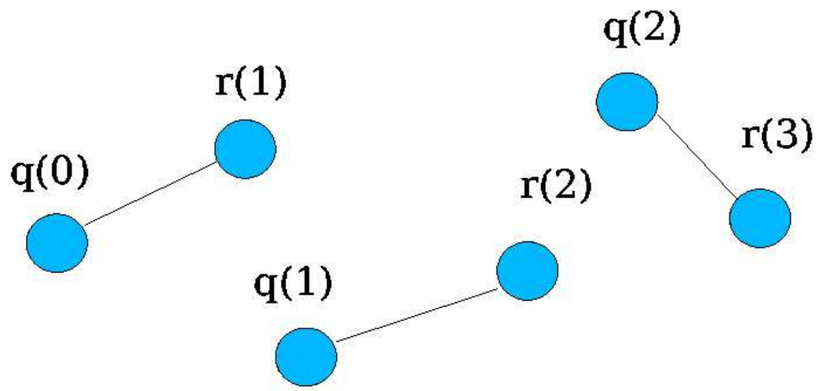

Figure 3: Line Elements of finite dimensional discrete mechanics. Momenta correspond to constraint $q(i) = r(i)$

We descretize a continuum problem with a space-time mesh (we use a simplified version of notation presented [7]). We let $\mathcal{E}_d$ be the set of elements of the space-time mesh and let $\mathcal{X}_d$ be the set of nodes in the space-time mesh. We let $\mathcal{E}_d(X)$ be the set of elements E such that node $X \in E$. The action sum we intend to minimize (assuming convexity of the discrete Lagrangian) is

$$S_d = \sum_{E \in \mathcal{E}_d} L_d(E). \tag{24}$$

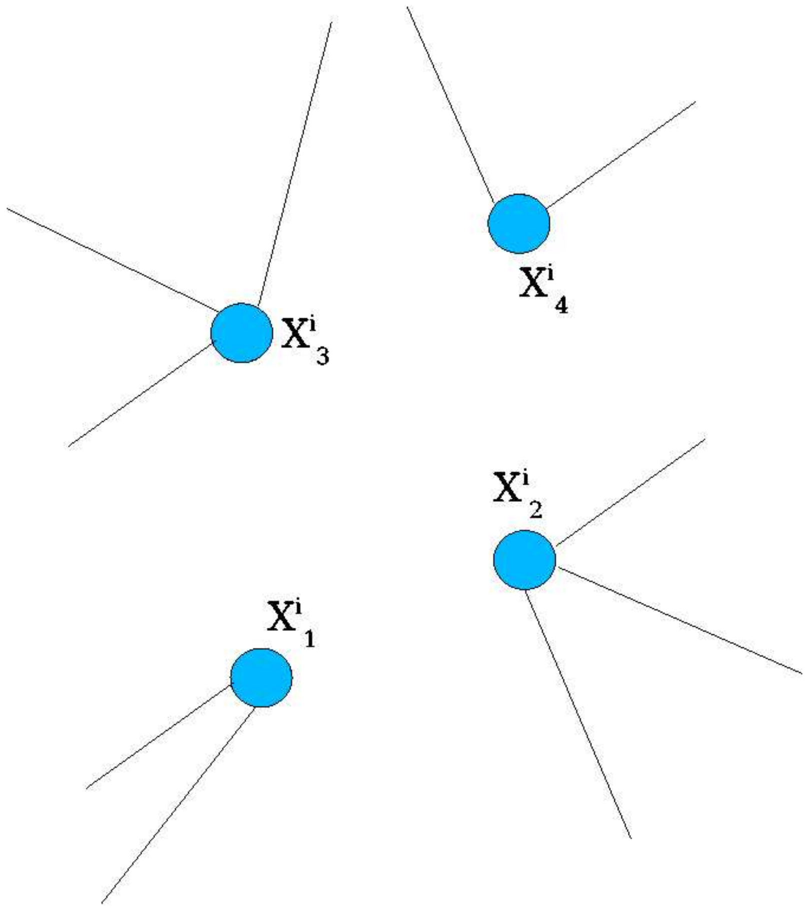

Figure 4: Nodes in space-time mesh for multi-symplectic mechanics. Momenta correspond to the constraint $X^i_j = X^i_k$.

We enumerate all the nodes as X^i for $i = 0, \ldots, N_{tot}$. We enumerate the elements in $\mathcal{E}_d(X^i)$ as E^i_j for $j = 0, \ldots, N_i$. Then the discrete Euler-Lagrange equations for multisymplectic mechanics are

$$\sum_{j=0}^{N_i} \frac{\partial L_d(E^i_j)}{\partial X^i} = 0. \tag{25}$$

We define the conjugate momenta as

$$p^i_j = \frac{\partial L_d(E^i_j)}{\partial X^i_j}. \tag{26}$$

Note that in general, as in the finite dimensional case, we allow a single node break apart and have multiple representations (see Figure 4).

Now we generalize Proposition 1 to the multisymplectic case.

Proposition 2 *For $0 \le i \le N_{tot}$, there exists a node X^i that satisfies the corresponding DEL equation iff $\sum_{j=0}^{N_i} p^i_j = 0$.*

Proof: $(\Rightarrow)$ Let X^i satisfy the corresponding DEL equation. Then using definition of the conjugate momenta

$$\begin{array}{rcl} 0 & = & \sum_{j=0}^{N_i} \frac{\partial L_d(E_j^i)}{\partial X^i} \\ & = & \sum_{j=0}^{N_i} p_j^i. \end{array} \tag{27}$$

($\Leftarrow$) Let $\sum_{j=0}^{N_i} p_j^i = 0$. Then first note

$$p_{N_i}^i = -\sum_{j=0}^{N_i-1} p_j^i. \tag{28}$$

Therefore we have

$$\begin{array}{rcl} \sum_{j=0}^{N_i} p_j^i X_j^i & = & \left(\sum_{j=0}^{N_i-1} p_j^i X_j^i\right) - \left(\sum_{j=0}^{N_i-1} p_j^i X_{N_i}^i\right) \\ & = & \sum_{j=0}^{N_i-1} p_j^i \left(X_j^i - X_{N_i}^j\right). \end{array} \tag{29}$$

Now we plug the result in to the definition of p, to get

$$\begin{array}{rcl} 0 & = & \begin{bmatrix} \frac{\partial L_d(E_0^i)}{\partial X_0^i} - p_0^i \\ \vdots \\ \frac{\partial L_d(E_{N_i}^i)}{\partial X_{N_i}^i} - p_{N_i}^i \end{bmatrix} \\ & = & \frac{\partial}{\partial X_j^i}\left[\sum_{j=0}^{N_i}\left(L_d(E_j^i) - p_j^i X_j^i\right)\right] \\ & = & \frac{\partial}{\partial X_j^i}\left[\left(\sum_{j=0}^{N_i} L_d(E_j^i)\right) - \sum_{j=0}^{N_i-1} p_j^i \left(X_j^i - X_{N_i}^i\right)\right]. \end{array} \tag{30}$$

Therefore, by the Lagrange Multiplier Theorem [6], we have that the X_j^i minimize the sum $\sum_{j=0}^{N_i} L_d(E_j^i)$ with X_j^i constrained to $X_j^i = X_{N_i}^i$. That, is all the different representations of X^i reduce to a single node $X_{N_i}^i$. Plugging in the constraint gives us a single node that satisfies the DEL equation ■

Having the momenta sum to zero enforces the constraint that all representations of a single node are the same. That implies that our mesh does not break apart. Assuming we have a convex discrete Lagrangian $L_d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, if our momenta sum to zero, then X and p satisfy the KKT conditions and strong duality holds.

The primal algorithm can be solved in a decentralized manner by once again just doing a simple gradient descent over the action sum. The algorithm works as follows:

- 1. At the elements $E_j^i \in \mathcal{E}_d(X^i)$, given E_j^i compute the momenta

$$p_j^i = \frac{\partial L_d(E_j^i)}{\partial X^i}. \tag{31}$$

  Then pass the momenta to node X^i.

- 2. At the node perform gradient descent

$$X^i := X^i - \gamma \sum_{j=0}^{N_i} p_j^i. \tag{32}$$

The dual algorithm solves the optimization problem

$$\begin{array}{c} minimize \quad \sum_{i=0}^{N_{tot}} \sum_{j=0}^{N_i} L_d(E_j^i) \quad s.t. \\ X_j^i = X_k^i. \end{array} \tag{33}$$

We must note that E_j^i has implicit dependence on X^i.

This problem has a Lagrangian function given by

$$K(X,p) = \sum_{i=0}^{N_{tot}} \left[\sum_{j=0}^{N_i} L_d(E_j^i) - \sum_{j=0}^{N_i-1} p_j^i \left(X_j^i - X_{N_i}^i\right)\right] \tag{34}$$

We compute the dual algorithm is computed as:

- 1. At each element, given p^i_j, compute the element that is the solution to

$$p^i_j = \frac{\partial L_d(E^i_j)}{\partial X^i}. \tag{35}$$

  Note that this the same as minimizing the Lagrangian function (34), by convexity.

- 2. At each node representation, we perform gradient ascent on the dual function by

$$p^i_j := p^i_j - \gamma \left(X^i_j - X^i_{N_i} \right). \tag{36}$$

For a convex discrete Lagrangian, as noted above, we get strong duality so the dual algorithm converges to a primal-dual optimal solution.

Network Congestion Control This section provides a brief sketch of the similarities between the internet congestion control algorithm and the distributed algorithms presented above. For a more detailed treatment on the congestion control problem see [3].

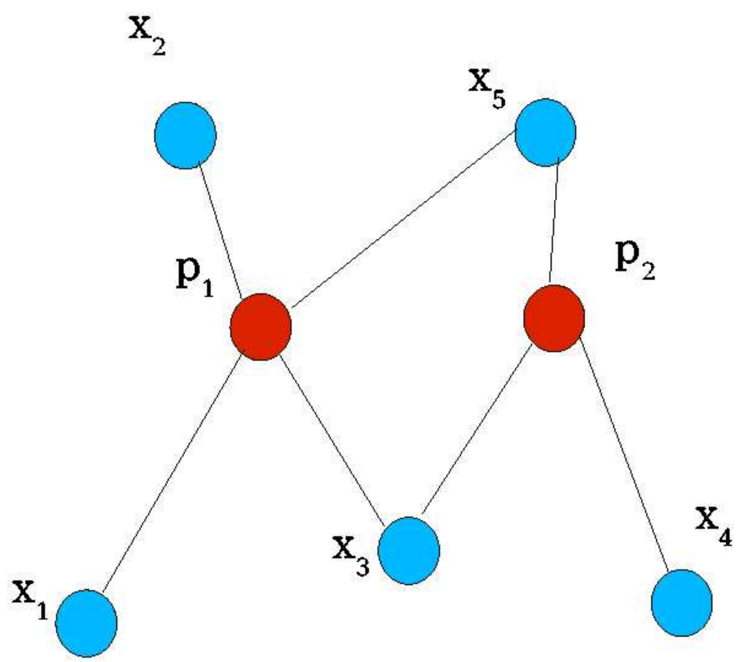

Figure 5: An example of a small internet.

We model the internet as sources that transmit information with rate x_s over a set of links. Each link has a measure of congestion p_l. We can represent the network as a graph as seen in Figure 5. In turn we can represent the topology of this graph with the routing matrix

$$R_{ls} = \begin{cases} 1 & source \quad s \quad uses \quad link \quad l \\ 0 & otherwise. \end{cases} \tag{37}$$

We assume that each source knows the total congestion of the links connected to it. That is, at each s, we know

$$\sum_l R_{ls} p_l. \tag{38}$$

Likewise, each link has information about the total flow into it

$$\sum_s R_{ls} x_s. \tag{39}$$

Network congestion control is solved by the optimization problem

$$\begin{array}{c} maximize \quad \sum_s U_s(x_s) \quad s.t. \\ \sum_s R_{ls} x_s \leq c_l. \end{array} \tag{40}$$

Here U_s is a concave non-decreasing utility function for each source s. This is a concave maximization, which is equivalent to the convex minimization of the negative objective function. An implicit constraint on the system is that $x_s \geq 0$.

Notice that the coupling of the source rates through the constraint prevents the primal problem from being solved in a distributed fashion.

The Lagrangian of the system is

$$L(x,p) = \sum_s \left[U_s(x_s) - \sum_l (R_{ls} x_s p_l - p_l c_l) \right]. \tag{41}$$

One should note that since each source has access to the total congestion at the source $\sum_l R_{ls} p_l$, the term $U_s(x_s) - x_s \sum_l R_{ls} p_l$ can be maximized at each source separately. Thus our dual problem becomes

$$\min_{p \geq 0} \sum_s \max_{x_s \geq 0} \left(U_s(x_s) - x_s \sum_l R_{ls} p_l \right) + \sum_l p_l c_l. \tag{42}$$

One should note the partial derivatives of the dual function are given by

$$\frac{\partial g}{\partial p_l} = c_l - \sum_s R_{ls} x_s. \tag{43}$$

We note that active queue management (AQM) at the links is designed to enforce the KKT conditions and thus give strong duality, since this is concave maximization.

One possible distributed algorithm is as follows:

- 1. At the sources, given $\sum_l R_{ls} p_l$ compute

$$x_s^* = \arg\max_{x_s \geq 0} U_s(x_s) - x_s \sum_l R_{ls} p_l \tag{44}$$

  Pass x_s to the corresponding links.

- 2. At the links, given $\sum_s R_{ls} x_s^*$, perform gradient descent on the dual problem as

$$p_l := p_l + \gamma \left(\sum_s R_{ls} x_s^* - c_l \right). \tag{45}$$

Conclusion As stated many times throughout the paper, it is clear that variational integrators have a similar mathematical structure to the congestion control problem. One could see this work as a starting point to designing large scale systems around this mathematical structure. Notably, one might use this methodology to for large scale mechanical control design using asynchronous variational integration. Future work could also include generalizing the mathematical structure to account for changes in graph topology such as verteces appearing or being removed. [4].

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] Sanjay Lall. Duality and asynchronous distributed algorithms for variational mechanics and feedback control of networks. Slide presentation of joint work with Matthew West, 2004.

[3] Steven H. Low. A duality model of tcp and queue management algorithms. *IEEE/ACM Trans. on Networking*, 2003.

[4] J. E. Marsden. Personal Communication.

[5] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 2001.

[6] Jerrold E. Marsden and Tudor Ratiu. *Introduction to Mechanics and Symmetry*. Springer-Verlag, 1999.

[7] A. Lew J. E. Marsden M. Ortiz and M. West. Asynchronous variational integrators. *Arch. Rational Mech. Anal.*, 2003.