Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications

Daniel P. Palomar

Princeton University

Guest Lecture in ELE539A: Optimization of Communication Systems (Prof. Mung Chiang), Princeton University

March 15, 2006

Outline

- Introduction
- Decomposition Techniques: primal/dual, direct/indirect, multilevel, algorithms.
- Applications
- Conclusions

- Material of this lecture from

Daniel Palomar and Mung Chiang, "Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications," in *Proc. IEEE Infocom, Barcelona*, Spain, April 23 – 29, 2006.

Introduction (I)

- Communication networks which are ubiquitous in our modern society.

- We need to design and optimize network. How?

- Two extreme approaches naturally arise:
  - competitive networks: game-theoretic approach (distributed algorithms but not the best of the network)
  - cooperative networks: global optimization problem (best of the network but centralized algorithms)

- We want both features: i) best of the network and ii) distributed algorithms. Can we achieve that??

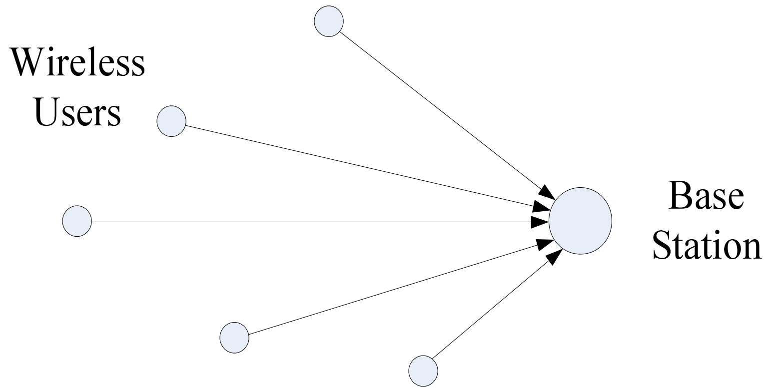
Introduction (II): NUM

- To design the network as a whole:
  - we will measure the "happiness" of a user through a utility function of the optimization variables: $U_i(\mathbf{x}_i)$
  - we will measure the "happiness" of the network with the aggregate utility: $\sum_i U_i(\mathbf{x}_i)$

- We will formulate the design of the network as the maximization of the aggregate utility of the users subject to a variety of constraints:

Network Utility Maximization (NUM)

Introduction (III)

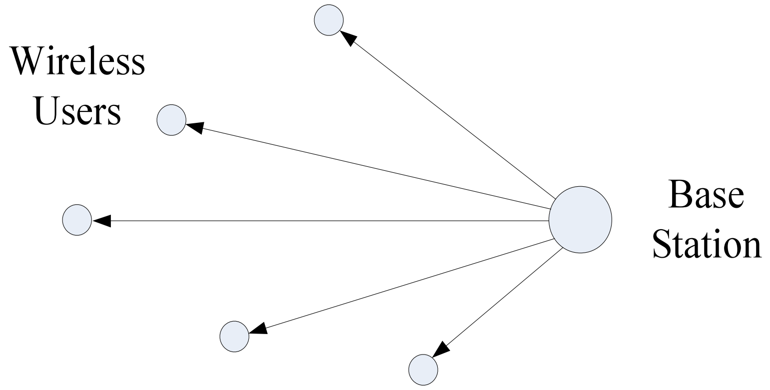


- Consider the uplink problem formulation

$$\begin{array}{lll} \underset{\{r_i,p_i\}}{\text{maximize}} & \sum_i U_i\left(r_i\right) & \\ \text{subject to} & r_i \leq \log\left(1+g_i p_i\right) & \forall i \\ & p_i \leq P_i. & \end{array}$$

- It naturally decouples into parallel subproblems for each of the users, with solution: $r_i^{\star} = \log\left(1+g_i P_i\right)$.

Introduction (IV)



- Consider now the downlink problem formulation

$$\begin{array}{ll} \underset{\{r_i,p_i\}}{\text{maximize}} & \sum_i U_i\left(r_i\right) \\ \text{subject to} & r_i \leq \log\left(1+g_i p_i\right) \qquad \forall i \\ & \sum_i p_i \leq P_T. \end{array}$$

- It does not decouple into parallel subproblems because of the coupling constraint: $\sum_i p_i \leq P_T$.

Introduction (V)

- Real problems do not decouple naturally.

- Centralized algorithms are theoretically possible but not desirable in practice (not scalable, not robust, too much signalling, not adaptive).

- Can we still obtain distributed algorithms to solve such coupled problems ??

- The classical approach is to use a dual-decomposition to obtain a distributed solution.

- However, there are many other alternatives and we can obtain a variety of distributed implementations.

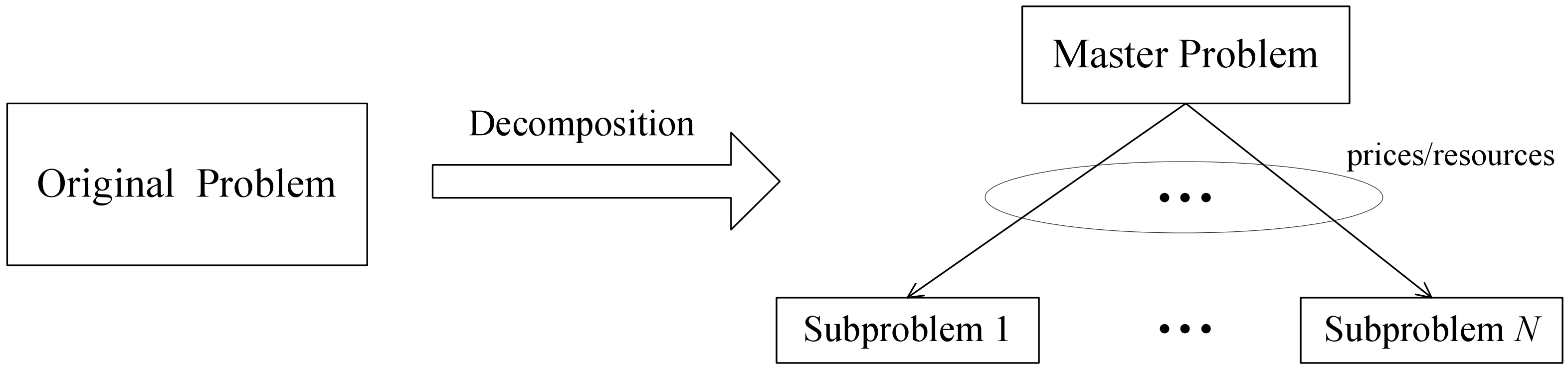
Introduction (VI)

- We will see a systematic way to obtain a variety of different distributed algorithms combining:

  i) direct primal/dual decompositions,
  ii) indirect primal/dual decompositions,
  iii) multilevel decomposition, and
  iv) use of different optimization methods (subgradient, Gauss-Seidel, Jacobi).

Decomposition Techniques: General Idea (I)

- The idea is to decompose the original large and coupled problem into subproblems (which can be locally solved) and a master problem (which is in charge of coordinating the subproblems):



- There has to be some signalling between master problem and subproblems.
- Alternative decomposition leads to different layered protocol architecture in the framework of Layering as Optimization Decomposition.

Decomposition Techniques: General Idea (II)

- Two main classes of decomposition techniques: *primal decomposition* and *dual decomposition*.

- Primal decomposition: *decompose original problem* by optimizing over one set of variables and then over the remaining set.
  - Interpretation: master problem directly allocates the existing resources to subproblems.

- Dual decomposition: *decompose dual problem* (obtained after a Lagrange relaxation of the coupling constraints)
  - Interpretation: master problem sets prices for the resources to subproblems.

Decomposition Techniques: Primal Decomp. (I)

- The following convex problem (with coupling variable $\mathbf{y}$)

$$\begin{array}{lll} \underset{\mathbf{y},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i\right) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{A}_i\mathbf{x}_i \leq \mathbf{y} & \\ & \mathbf{y} \in \mathcal{Y} & \end{array}$$

is decomposed into the subproblems:

$$\begin{array}{ll} \underset{\mathbf{x}_i \in \mathcal{X}_i}{\text{maximize}} & f_i\left(\mathbf{x}_i\right) \\ \text{subject to} & \mathbf{A}_i\mathbf{x}_i \leq \mathbf{y} \end{array}$$

and the master problem

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{maximize}} \quad \sum_i f_i^{\star}\left(\mathbf{y}\right)$$

where $f_i^{\star}\left(\mathbf{y}\right)$ is the optimal value of the ith subproblem.

Decomposition Techniques: Primal Decomp. (II)

- If the original problem is convex optimization, then the subproblems as well as the master problem are all convex.

- To maximize $\sum_i f_i^{\star}(\mathbf{y})$: a gradient/subgradient method, which only requires the knowledge of subgradient of each $f_i^{\star}(\mathbf{y})$ given by

$$\mathbf{s}_i(\mathbf{y}) = \boldsymbol{\lambda}_i^{\star}(\mathbf{y})$$

  where $\boldsymbol{\lambda}_i^{\star}(\mathbf{y})$ is the optimal Lagrange multiplier corresponding to the constraint $\mathbf{A}_i\mathbf{x}_i \leq \mathbf{y}$ in the ith subproblem.

- The global subgradient is then $\mathbf{s}(\mathbf{y}) = \sum_i \mathbf{s}_i(\mathbf{y}) = \sum_i \boldsymbol{\lambda}_i^{\star}(\mathbf{y})$.

- The subproblems can be locally solved with the knowledge of $\mathbf{y}$.

Decomposition Techniques: Dual Decomp. (I)

- The dual of the following convex problem (with coupling constraint)

$$\begin{array}{ll} \underset{\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \qquad \forall i, \\ & \sum_i \mathbf{h}_i(\mathbf{x}_i) \leq \mathbf{c} \end{array}$$

  is decomposed into subproblems:

$$\begin{array}{ll} \underset{\mathbf{x}_i}{\text{maximize}} & f_i(\mathbf{x}_i) - \boldsymbol{\lambda}^T \mathbf{h}_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i. \end{array}$$

  and the master problem

$$\underset{\boldsymbol{\lambda} \geq \mathbf{0}}{\text{minimize}} \quad g(\boldsymbol{\lambda}) = \sum_i g_i(\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T \mathbf{c}$$

  where $g_i(\boldsymbol{\lambda})$ is the optimal value of the ith subproblem.

Decomposition Techniques: Dual Decomp. (II)

- The dual decomposition is in fact solving the dual problem instead of the original primal one.
- The dual problem is always convex but we need convexity of the original problem to have strong duality.
- To minimize the dual function $g(\boldsymbol{\lambda})$: gradient/subgradient method, which only requires the knowledge of subgradient of each $g_i(\boldsymbol{\lambda})$:

$$\mathbf{s}_i(\boldsymbol{\lambda}) = -\mathbf{h}_i(\mathbf{x}_i^\star(\boldsymbol{\lambda})),$$

  where $\mathbf{x}_i^\star(\boldsymbol{\lambda})$ is the optimal solution of the ith subproblem for a given $\boldsymbol{\lambda}$.

Decomposition Techniques: Dual Decomp. (III)

- The global subgradient is then $\mathbf{s}(\boldsymbol{\lambda}) = \sum_i \mathbf{s}_i(\mathbf{y}) + \mathbf{c} = \mathbf{c} - \sum_i \mathbf{h}_i(\mathbf{x}_i^\star(\boldsymbol{\lambda}))$.

- The subproblems can be locally and independently solved with knowledge of $\boldsymbol{\lambda}$.

General Subgradient Results: Primal Decomposition

- **Lemma:** Consider:

$$f^{\star}(\mathbf{y}) \triangleq \sup_{\mathbf{x}: f_i(\mathbf{x},\mathbf{y}) \leq 0} f_0(\mathbf{x},\mathbf{y})$$

where f_0 is (strictly) concave and the f_i's are convex. Then, $f^{\star}(\mathbf{y})$ is (strictly) concave and a subgradient is given by

$$\mathbf{s}_{\mathbf{y}}^{\star}(\mathbf{y}) = \mathbf{s}_{0,\mathbf{y}}(\mathbf{x}^{\star}(\mathbf{y}),\mathbf{y}) - \mathbf{S}_{\mathbf{y}}(\mathbf{x}^{\star}(\mathbf{y}),\mathbf{y})\,\boldsymbol{\lambda}^{\star}(\mathbf{y})$$

where $\mathbf{s}_{0,\mathbf{y}}(\mathbf{x},\mathbf{y})$ is a subgradient of $f_0(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{y}$ and $\mathbf{S}_{\mathbf{y}}(\mathbf{x},\mathbf{y})$ is a matrix containing in the ith column a subgradient of $f_i(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{y}$.

General Subgradient Results: Dual Decomposition

- **Lemma:** Consider the following dual function defined as the supremum of a partial Lagrangian:

$$g(\boldsymbol{\lambda}) \triangleq \sup_{\mathbf{x}:g_i(\mathbf{x})\leq 0} \left\{ f_0(\mathbf{x}) - \sum_i \lambda_i f_i(\mathbf{x}) \right\}.$$

  Then, $g(\boldsymbol{\lambda})$ is convex and a subgradient, denoted by $\mathbf{s}_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$, is given by

$$s_{\lambda_i}(\boldsymbol{\lambda}) = -f_i(\mathbf{x}^{\star}(\boldsymbol{\lambda}))$$

  where $\mathbf{x}^{\star}(\boldsymbol{\lambda})$ is the value of $\mathbf{x}$ that achieves the supremum for a given $\boldsymbol{\lambda}$ (which is obtained 'for free' each time that $g(\boldsymbol{\lambda})$ is evaluated at some point).

Indirect Primal/Dual Decompositions (I)

- Often the problem can be reformulated and more effective primal and dual decompositions can be indirectly applied.
- The introduction of auxiliary variables provides much flexibility in terms of choosing a primal or a dual decomposition and of the resulting distributed algorithm.
- Consider the problem previously decomposed with a primal decomposition:

$$\begin{array}{lll} \underset{\mathbf{y},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i\right) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y} & \\ & \mathbf{y} \in \mathcal{Y}. & \end{array}$$

Indirect Primal/Dual Decompositions (II)

- It can also be solved with an indirect dual decomposition by first introducing the additional variables $\{\mathbf{y}_i\}$:

$$\begin{array}{ll}
\underset{\mathbf{y},\{\mathbf{y}_i\},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i\right) \\
\text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \qquad\qquad \forall i \\
& \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y}_i \\
& \mathbf{y}_i = \mathbf{y} \\
& \mathbf{y} \in \mathcal{Y}.
\end{array}$$

- We have transformed the coupling variable $\mathbf{y}$ into a set of coupling constraints $\mathbf{y}_i = \mathbf{y}$ which can be dealt with using a dual decomposition.

Indirect Primal/Dual Decompositions (III)

- Consider now the problem previously decomposed with a dual decomposition:

$$\begin{array}{ll} \underset{\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i\right) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \qquad\qquad \forall i, \\ & \sum_i \mathbf{h}_i\left(\mathbf{x}_i\right) \leq \mathbf{c}. \end{array}$$

Indirect Primal/Dual Decompositions (IV)

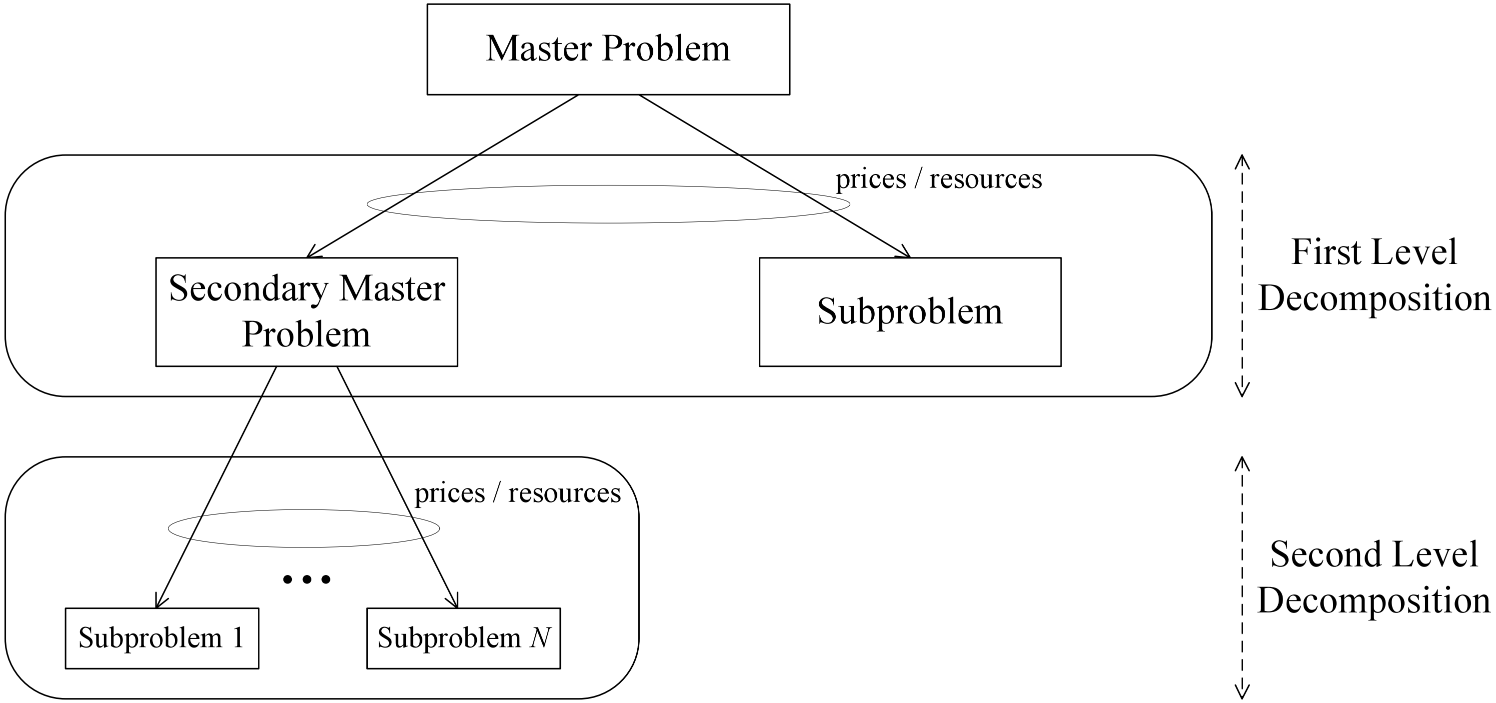
- It can also be solved with an indirect primal decomposition by introducing again additional variables $\{\mathbf{y}_i\}$:

$$\begin{array}{lll} \underset{\{\mathbf{y}_i\},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i\right) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{h}_i\left(\mathbf{x}_i\right) \leq \mathbf{y}_i & \\ & \sum_i \mathbf{y}_i \leq \mathbf{c}. & \end{array}$$

- We have transformed the coupling constraint $\sum_i \mathbf{h}_i\left(\mathbf{x}_i\right) \leq \mathbf{c}$ into a coupling variable $\mathbf{y} = \left[\mathbf{y}_1^T, \cdots, \mathbf{y}_N^T\right]^T$ which can be dealt with using a primal decomposition.

Multilevel Primal/Dual Decompositions (I)

- Hierarchical application of primal/dual decompositions to obtain smaller and smaller subproblems:



- Important technique that leads to alternatives of distributed architectures.

Multilevel Primal/Dual Decompositions (II)

- Example: consider the following problem which includes both a coupling variable and a coupling constraint:

$$
\begin{array}{lll}
\underset{\mathbf{y},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i\left(\mathbf{x}_i, \mathbf{y}\right) & \\
\text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\
& \sum_i \mathbf{h}_i\left(\mathbf{x}_i\right) \le \mathbf{c} & \\
& \mathbf{A}_i \mathbf{x}_i \le \mathbf{y} & \\
& \mathbf{y} \in \mathcal{Y}. &
\end{array}
$$

Multilevel Primal/Dual Decompositions (III)

- Decomposion #1: first take a primal decomposition with respect to the coupling variable $\mathbf{y}$ and then a dual decomposition with respect to the coupling constraint $\sum_i \mathbf{h}_i(\mathbf{x}_i) \leq \mathbf{c}$. This would produce a two-level optimization decomposition: a master primal problem, a secondary master dual problem, and the subproblems.

- Decomposion #2: first take a dual decomposition and then a primal one.

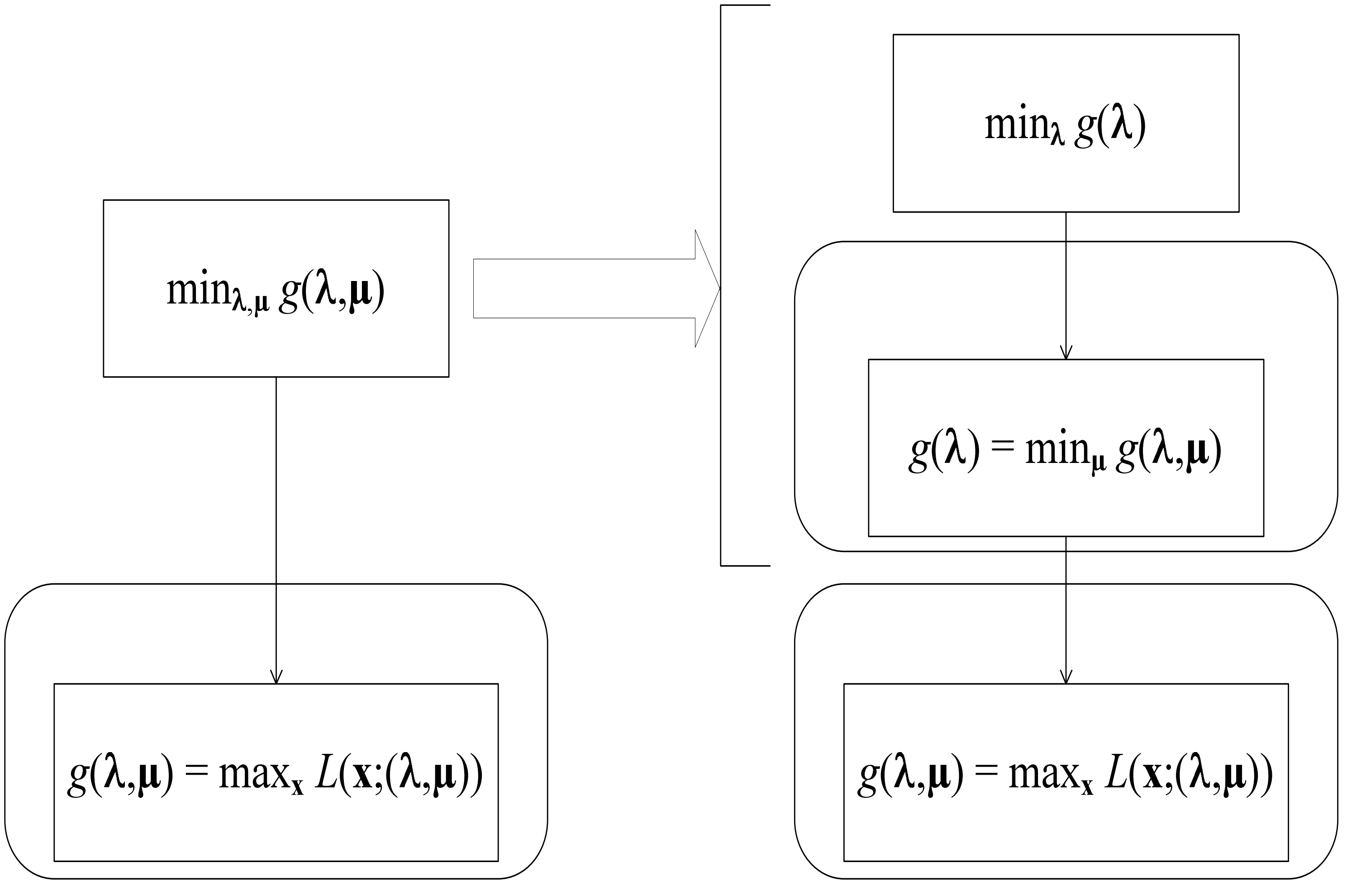
Multilevel Primal/Dual Decompositions (II)

- Example:

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{maximize}} & f_0(\mathbf{x}) & \\ \text{subject to} & f_i(\mathbf{x}) \leq 0 & \forall i \\ & g_i(\mathbf{x}) \leq 0. & \end{array}$$

- Decomposition #1 (dual-primal): first apply a full dual decomposition by relaxing both sets of constraints to obtain the dual function $g(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and then a primal decomposition on the dual problem by minimizing g first over $\boldsymbol{\mu}$ and later over $\boldsymbol{\lambda}$: $\min_{\boldsymbol{\lambda}} \min_{\boldsymbol{\mu}} g(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

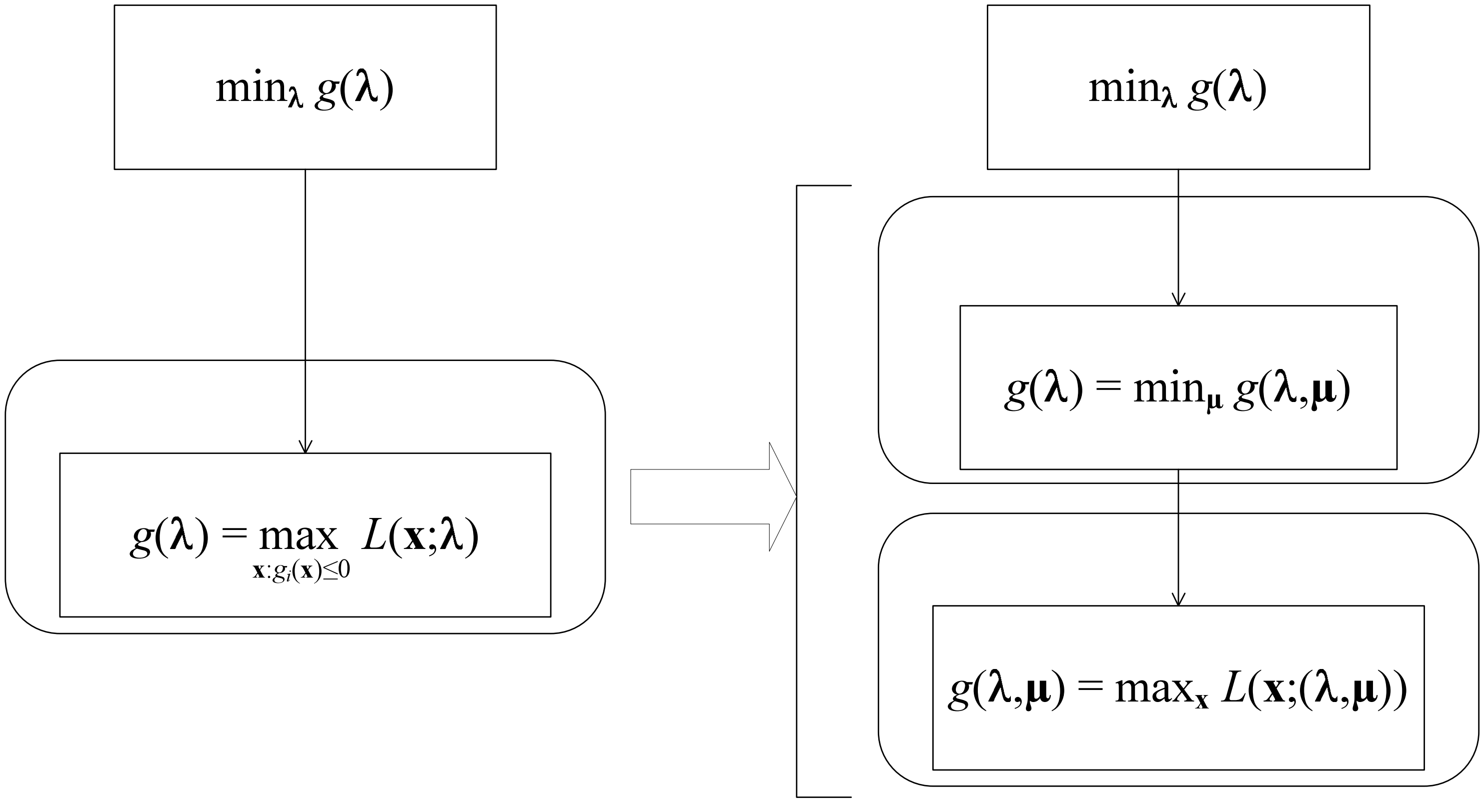
Multilevel Primal/Dual Decompositions (III)



Multilevel Primal/Dual Decompositions (IV)

- Decomposition #2 (dual-dual): first apply a partial dual decomposition by relaxing only one set of constraints, say $f_i(\mathbf{x}) \leq 0, \ \forall i$, obtaining the dual function $g(\boldsymbol{\lambda})$ to be minimized by the master problem. But to compute $g(\boldsymbol{\lambda})$ for a given $\boldsymbol{\lambda}$, the partial Lagrangian has to be maximized subject to the remaining constraints $g_i(\mathbf{x}) \leq 0 \ \forall i$, for which yet another relaxation can be used.

Multilevel Primal/Dual Decompositions (V)



Algorithms: Gradient/Subgradient Methods (I)

- After performing a decomposition, the objective function of the resulting master problem may or may not be differentiable.

- For differentiable/nondifferentiable functions a gradient/subgradient method is very convenient because of its simplicity, little requirements of memory usage, and amenability for parallel implementation.

Algorithms: Gradient/Subgradient Methods (II)

- Consider

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & f_0(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}. \end{array}$$

- Both the gradient and subgradient projection methods generate a sequence of feasible points $\{\mathbf{x}(t)\}$ as

$$\mathbf{x}(t+1) = [\mathbf{x}(t) + \alpha(t)\,\mathbf{s}(t)]_{\mathcal{X}}$$

where $\mathbf{s}(t)$ is a gradient/subgradient of f_0 at $\mathbf{x}(t)$, $[\cdot]_{\mathcal{X}}$ denotes the projection onto $\mathcal{X}$, and $\alpha(t)$ is the stepsize.

Algorithms: Gradient/Subgradient Methods (III)

- Many results on convergence of the gradient/subgradient method with different choices of stepsize:
  - for a diminishing stepsize rule $\alpha(t) = \frac{1+m}{t+m}$, where m is a fixed nonnegative number, the algorithm is guaranteed to converge to the optimal value (assuming bounded gradients/subgradients).
  - for a constant stepsize $\alpha(t) = \alpha$, more convenient for distributed algorithms, the gradient algorithm converges to the optimal value provided that the stepsize is sufficiently small (assuming that the gradient is Lipschitz), whereas for the subgradient algorithm the best value converges to within some range of the optimal value (assuming bounded subgradients).

Algorithms: Gauss-Seidel and Jacobi Methods

- Gauss-Seidel algorithm (block-coordinate descent algorithm): optimize $f(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ sequentially:

$$\mathbf{x}_k^{(t+1)} = \arg\max_{\mathbf{x}_k} f\left(\mathbf{x}_1^{(t+1)}, \cdots, \mathbf{x}_{k-1}^{(t+1)}, \mathbf{x}_k, \mathbf{x}_{k+1}^{(t)}, \cdots, \mathbf{x}_N^{(t)}\right)$$

  where t is the index for a global iteration.

- Jacobi algorithm: optimize $f(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ in parallel:

$$\mathbf{x}_k^{(t+1)} = \arg\max_{\mathbf{x}_k} f\left(\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_{k-1}^{(t)}, \mathbf{x}_k, \mathbf{x}_{k+1}^{(t)}, \cdots, \mathbf{x}_N^{(t)}\right).$$

- If the mapping defined by $T(\mathbf{x}) = \mathbf{x} - \gamma \nabla f(\mathbf{x})$ is a contraction for some γ, then $\left\{\mathbf{x}^{(t)}\right\}$ converges to solution $\mathbf{x}^\star$ geometrically.

Standard Algorithm for Basic NUM (I)

- The standard dual-based algorithm is a one-level full dual decomposition.

- Network with L links, each with capacity c_l, and S sources transmitting at rate x_s. Each source s emits one flow, using a fixed set of links $L(s)$ in its path, and has a utility function $U_s(x_s)$:

$$\begin{array}{ll} \underset{\mathbf{x}\geq\mathbf{0}}{\text{maximize}} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s:l\in L(s)} x_s \leq c_l \quad \forall l \end{array}$$

- This problem is solved with a single-level dual decomposition technique.

Standard Algorithm for Basic NUM (II)

- Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s:l \in L(s)} x_s \right)$$
$$= \sum_s \left[U_s(x_s) - \lambda^s x_s \right] + \sum_l \lambda_l c_l$$

where $\lambda^s = \sum_{l \in L(s)} \lambda_l$.

- Each source maximizes its Lagrangian:

$$x_s^\star(\lambda^s) = \arg\max_{x_s \geq 0} \left[U_s(x_s) - \lambda^s x_s \right] \quad \forall s.$$

Standard Algorithm for Basic NUM (III)

- Master dual problem:

$$\underset{\boldsymbol{\lambda} \geq \mathbf{0}}{\text{minimize}} \;\; g\left(\boldsymbol{\lambda}\right) = \sum_s g_s\left(\boldsymbol{\lambda}\right) + \boldsymbol{\lambda}^T \mathbf{c}$$

where $g_s\left(\boldsymbol{\lambda}\right) = L_s\left(x_s^{\star}\left(\lambda^s\right), \lambda^s\right)$. To minimize the master problem a subgradient method can be used:

$$\lambda_l\left(t+1\right) = \left[\lambda_l\left(t\right) - \alpha\left(c_l - \sum_{s:l \in L(s)} x_s^{\star}(\lambda^s\left(t\right))\right)\right]^+ \quad \forall l.$$

Applic. 1: Power-Constrained Rate Allocation (I)

- Basic NUM problem but with variable link capacities $\{c_l(p_l)\}$:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{p}\geq\mathbf{0}}{\text{maximize}} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s:l\in L(s)} x_s \leq c_l(p_l) \quad \forall l \\ & \sum_l p_l \leq P_T. \end{array}$$

- Very simple problem, but already contains sufficient elements such that one can try different decompositions.

- We will consider: i) a primal decomposition with respect to the power allocation and ii) a dual decomposition with respect to the flow constraints.

Applic. 1: Power-Constrained Rate Allocation (II)

- **Primal decomposition**: fix the power allocation $\mathbf{p}$, the link capacities become fixed numbers and the problem reduces to a basic NUM solved by dual decomposition.

- Master primal problem:

$$\begin{array}{ll} \underset{\mathbf{p}\geq\mathbf{0}}{\text{maximize}} & U^{\star}(\mathbf{p}) \\ \text{subject to} & \sum_{l} p_l \leq P_T, \end{array}$$

  where $U^{\star}(\mathbf{p})$ is the optimal objective value for a given $\mathbf{p}$.

- Subgradient of $U^{\star}(\mathbf{p})$ with respect to c_l is given by the Lagrange multiplier λ_l associated with the constraint $\sum_{s:l\in L(s)} x_s \leq c_l$.

Applic. 1: Power-Constrained Rate Allocation (III)

- Subgradient of $U^{\star}(\mathbf{p})$ with respect to p_l is given by $\lambda_l c_l'(p_l)$.

- Subgradient method for the master primal problem:

$$\mathbf{p}(t+1) = \left[\mathbf{p}(t) + \alpha \begin{bmatrix} \lambda_1^{\star}(\mathbf{p}(t))\, c_1'(p_1(t)) \\ \vdots \\ \lambda_L^{\star}(\mathbf{p}(t))\, c_L'(p_L(t)) \end{bmatrix} \right]_{\mathcal{P}}$$

where $[\cdot]_{\mathcal{P}}$ denotes the projection onto $\mathcal{P} \triangleq \{\mathbf{p} : \mathbf{p} \geq \mathbf{0}, \sum_l p_l \leq P_T\}$, which is a simplex.

- Due to the projection, this subgradient update cannot be performed independently by each link and requires some centralized approach.

Applic. 1: Power-Constrained Rate Allocation (IV)

- Projection: $\mathbf{p} = [\mathbf{p}_0]_{\mathcal{P}}$ is given by

$$p_l = \left(p_l^0 - \gamma\right)^+ \quad \forall l$$

where waterlevel γ is chosen as the minimum nonnegative value such that $\sum_l p_l \leq P_T$.

- Only the computation of γ requires a central node since the update of each power p_l can be done at each link.

Applic. 1: Power-Constrained Rate Allocation (V)

- **Dual decomposition**: relax the flow constraints $\sum_{s:l \in L(s)} x_s \leq c_l(p_l)$:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{p} \geq \mathbf{0}}{\text{maximize}} & \sum_s \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] + \sum_l c_l(p_l) \lambda_l \\ \text{subject to} & \sum_l p_l \leq P_T. \end{array}$$

- The master dual problem updates the λ_l's as in the basic NUM.

Applic. 1: Power-Constrained Rate Allocation (VI)

- The Lagrangian decomposes into one maximization for each source, as in the basic NUM, plus the following maximization to update the power allocation:

$$\begin{array}{ll} \underset{\mathbf{p}\geq\mathbf{0}}{\text{maximize}} & \sum_l \lambda_l c_l\left(p_l\right) \\ \text{subject to} & \sum_l p_l \leq P_T \end{array}$$

  which can be further decomposed via a second-level dual decomposition yielding the following subproblems

$$\underset{p_l\geq 0}{\text{maximize}} \quad \lambda_l c_l\left(p_l\right) - \gamma p_l$$

Applic. 1: Power-Constrained Rate Allocation (VII)

with solution given by

$$p_l = \left(c_l'\right)^{-1}\left(\gamma/\lambda_l\right)$$

and a secondary master dual problem that updates the dual variable γ as

$$\gamma\left(t+1\right) = \left[\gamma\left(t\right) - \alpha\left(P_T - \sum_l p_l^{\star}\left(\gamma\left(t\right)\right)\right)\right]^{+}.$$

Applic. 1: Power-Constrained Rate Allocation (VIII)

- We have obtained two different distributed algorithms for power-constrained rate allocation NUM:
  - **primal-dual decomposition**: master primal problem solved by a subgradient power update, which needs a small central coordination for the waterlevel, and for each set of powers the resulting NUM is solved via the standard dual-based decomposition.
    - ◇ Two levels of decompositions: on the highest level there is a master primal problem, on a second level there is a secondary master dual problem, and on the lowest level the subproblems.

Applic. 1: Power-Constrained Rate Allocation (IX)

- **dual-dual decomposition**: master dual problem solved with the standard price update independently by each link and then, for a given set of prices, each source solves its own subproblem and the power allocation subproblem is solved with some central node updating the price and each link obtaining the optimal power.
  - ◇ Two levels of decompositions: on the highest level there is a master dual problem, on a second level there are rate subproblems and a secondary master dual problem, and on the lowest level the power subproblems.

Illustration of Decomp. of Network Utility Maxim.: Cellular Downlink Power-Rate Control (I)

- Problem:

$$\begin{array}{ll} \underset{\{r_i,p_i\}}{\text{maximize}} & \sum_i U_i(r_i) \\ \text{subject to} & r_i \le \log(g_i p_i) \quad \forall i \\ & p_i \ge 0 \\ & \sum_i p_i \le P_T. \end{array}$$

- Decompositions: i) primal, ii) partial dual, iii) full dual.

- Many variants of full dual decomposition: the master problem is

$$\underset{\boldsymbol{\lambda}\ge\mathbf{0},\gamma\ge 0}{\text{minimize}} \quad g(\boldsymbol{\lambda},\gamma)$$

and can e solved as listed next.

Illustration of Decomp. of Network Utility Maxim.: Cellular Downlink Power-Rate Control (II)

1. Direct subgradient update of $\gamma(t)$ and $\boldsymbol{\lambda}(t)$:

$$\gamma(t+1) = \left[\gamma(t) - \alpha\left(P_T - \sum_i p_i(t)\right)\right]^+$$

$$\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) - \alpha(\log(g_i p_i(t))) - r_i(t)\right]^+.$$

2. Optimization of dual function $g(\boldsymbol{\lambda}, \gamma)$ with a Gauss-Seidel method optimizing $\boldsymbol{\lambda} \to \gamma \to \boldsymbol{\lambda} \to \gamma \to \cdots$ (each λ_i is computed locally at each subnode in parallel):

$$\lambda_i = U_i'(\log(g_i \lambda_i / \gamma)) \qquad \text{and} \qquad \gamma = \sum_i \lambda_i / P_T.$$

Illustration of Decomp. of Network Utility Maxim.: Cellular Downlink Power-Rate Control (III)

3. Similar to 2), but optimizing $\lambda_1 \rightarrow \gamma \rightarrow \lambda_2 \rightarrow \gamma \rightarrow \cdots$ (λ_i's are not updated in parallel but sequentially).

4. Use an additional primal decomposition to minimize $g(\boldsymbol{\lambda}, \gamma)$ (multilevel decomposition): minimize $g(\gamma) = \inf_{\boldsymbol{\lambda} \geq \mathbf{0}} g(\boldsymbol{\lambda}, \gamma)$ via a subgradient algorithm (again, the λ_i's are computed locally and in parallel).

5. Similar to 4), but changing the order of minimization: minimize $g(\boldsymbol{\lambda}) = \inf_{\gamma \geq 0} g(\boldsymbol{\lambda}, \gamma)$ via a subgradient algorithm.

Illustration of Decomp. of Network Utility Maxim.: Cellular Downlink Power-Rate Control (IV)

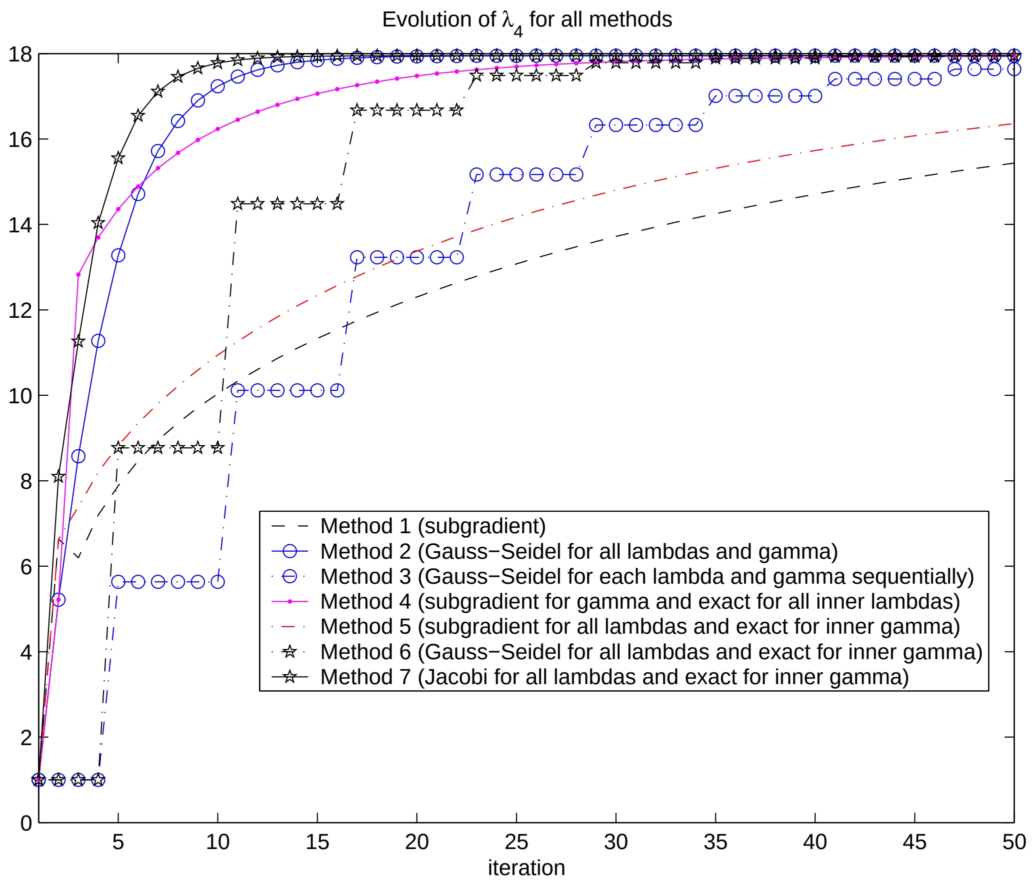
6. Similarly to 5), but with yet another level of decomposition on top of the primal decomposition of 5) (triple multilevel decomposition): minimize $g(\boldsymbol{\lambda})$ sequentially (Gauss-Seidel fashion) $\lambda_1 \to \lambda_2 \to \cdots$ (λ_i's are updated sequentially).

7. Similar to 5) and 6), but minimizing $g(\boldsymbol{\lambda})$ with in a Jacobi fashion $\boldsymbol{\lambda} \to \boldsymbol{\lambda} \to \cdots$ (λ_i's are updated in parallel). $\lambda_i^{(k+1)}$ is obtained by solving for λ_i in the following fixed-point equation:

$$\frac{g_i \lambda_i}{\exp\left(U_i'^{-1}(\lambda_i)\right)} = \gamma - \frac{\lambda_i^{(k)}}{P_T} + \frac{\lambda_i}{P_T}.$$

Numerical Results

- Downlink power/rate control problem with 6 nodes with utilities with utilities $U_i(r_i) = \beta_i \log r_i$. Evolution of λ_4 for all 7 methods:



Applic. 2: QoS Rate Allocation (I)

- Consider a NUM problem where users are differentiated in different QoS classes: this generates new coupling to the basic NUM problem.

- Denoting by $y_l^{(1)}$ and $y_l^{(2)}$ the aggregate rates of classes 1 and 2, respectively, along the lth link, the problem formulation is

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq\mathbf{0}}{\text{maximize}} & \sum_s U_s\left(x_s\right) \\ \text{subject to} & \sum_{s\in S_i:l\in L(s)} x_s \leq y_l^{(i)} \quad \forall l, i=1,2 \\ & \mathbf{y}^{(1)}+\mathbf{y}^{(2)} \leq \mathbf{c} \\ & \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)}. \end{array}$$

Applic. 2: QoS Rate Allocation (II)

- In the absence of the constraints $\mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)}$, it becomes the basic NUM.

- We will consider: i) a primal decomposition with respect to the aggregate rate of each class and ii) a dual decomposition with respect to the total aggregate rate constraints from both classes.

Applic. 2: QoS Rate Allocation (III)

- **Primal-Dual Decomposition**: fix the aggregate rates $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ and the problem becomes two independent subproblems, for $i = 1, 2$, identical to the basic NUM:

$$\begin{array}{ll} \underset{\mathbf{x} \geq \mathbf{0}}{\text{maximize}} & \sum_{s \in S_i} U_s\left(x_s\right) \\ \text{subject to} & \sum_{s \in S_i : l \in L(s)} x_s \leq y_l^{(i)} \qquad \forall l \end{array}$$

where the fixed aggregate rates $y_l^{(i)}$ play the role of the fixed link capacities in the basic NUM.

Applic. 2: QoS Rate Allocation (IV)

- The master primal problem is

$$\begin{array}{ll}\underset{\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq\mathbf{0}}{\text{maximize}} & U_1^{\star}\left(\mathbf{y}^{(1)}\right)+U_2^{\star}\left(\mathbf{y}^{(2)}\right)\\ \text{subject to} & \mathbf{y}^{(1)}+\mathbf{y}^{(2)}\leq\mathbf{c}\\ & \mathbf{c}_{\min}^{(i)}\leq\mathbf{y}^{(i)}\leq\mathbf{c}_{\max}^{(i)}\quad i=1,2\end{array}$$

  where $U_i^{\star}\left(\mathbf{y}^{(i)}\right)$ is the optimal objective value of the problem for the ith class for a given $\mathbf{y}^{(i)}$, with subgradient given by Lagrange multiplier $\boldsymbol{\lambda}^{(i)}$ associated to the constraints $\sum_{s\in S_i:l\in L(s)} x_s \leq y_l^{(i)}$.

- $\boldsymbol{\lambda}^{(i)}$ is the set of differential prices for the QoS class i.

Applic. 2: QoS Rate Allocation (V)

- Master primal problem can now be solved with a subgradient method by updating the aggregate rates as

$$\left[\begin{array}{c} \mathbf{y}^{(1)}(t+1) \\ \mathbf{y}^{(2)}(t+1) \end{array}\right] = \left[\left[\begin{array}{c} \mathbf{y}^{(1)}(t) \\ \mathbf{y}^{(2)}(t) \end{array}\right] + \alpha \left[\begin{array}{c} \boldsymbol{\lambda}^{\star(1)}\left(\mathbf{y}^{(1)}(t)\right) \\ \boldsymbol{\lambda}^{\star(2)}\left(\mathbf{y}^{(2)}(t)\right) \end{array}\right]\right]_{\mathcal{Y}}$$

where $[\cdot]_{\mathcal{Y}}$ denotes the projection onto the feasible convex set $\mathcal{Y} \triangleq \left\{ \left(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\right) : \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \leq \mathbf{c},\ \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)}\ i = 1, 2 \right\}$.

- This feasible set decomposes into a Cartesian product for each of the links: $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_L$. Subgradient update can be performed independently by each link simply with the knowledge of its corresponding Lagrange multipliers $\lambda_l^{(1)}$ and $\lambda_l^{(2)}$.

Applic. 2: QoS Rate Allocation (VI)

- **Partial Dual Decomposition**: dual decomposition by relaxing only the flow constraints $\sum_{s \in S_i : l \in L(s)} x_s \le y_l^{(i)}$:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y}^{(1)},\mathbf{y}^{(2)} \ge \mathbf{0}}{\text{maximize}} & \sum_{s \in S_1} \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] \\ & \quad + \sum_{s \in S_2} \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] \\ & \quad + \boldsymbol{\lambda}^{(1)T} \mathbf{y}^{(1)} + \boldsymbol{\lambda}^{(2)T} \mathbf{y}^{(2)} \\ \text{subject to} & \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \le \mathbf{c} \\ & \mathbf{c}_{\min}^{(i)} \le \mathbf{y}^{(i)} \le \mathbf{c}_{\max}^{(i)} \quad i = 1, 2. \end{array}$$

Applic. 2: QoS Rate Allocation (VII)

- This problem decomposes into one maximization for each source, as in the basic NUM, plus the following additional maximization to update the aggregate rates:

$$\begin{array}{ll} \underset{\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq\mathbf{0}}{\text{maximize}} & \boldsymbol{\lambda}^{(1)T}\mathbf{y}^{(1)}+\boldsymbol{\lambda}^{(2)T}\mathbf{y}^{(2)} \\ \text{subject to} & \mathbf{y}^{(1)}+\mathbf{y}^{(2)}\leq\mathbf{c} \\ & \mathbf{c}_{\min}^{(i)}\leq\mathbf{y}^{(i)}\leq\mathbf{c}_{\max}^{(i)} \quad i=1,2 \end{array}$$

which can be solved independently by each link with knowledge of its corresponding Lagrange multipliers $\lambda_l^{(1)}$ and $\lambda_l^{(2)}$.

Applic. 2: QoS Rate Allocation (VIII)

- Master dual problem is updated with the subgradient method

$$\lambda_l^{(i)}(t+1) = \left[\lambda_l^{(i)}(t) - \alpha\left(y_l^{(i)}(t) - \sum_{s \in S_i : l \in L(s)} x_s^\star(\lambda^{(i)s}(t))\right)\right]^+ \qquad \forall l, i = 1, 2.$$

Applic. 2: QoS Rate Allocation (IX)

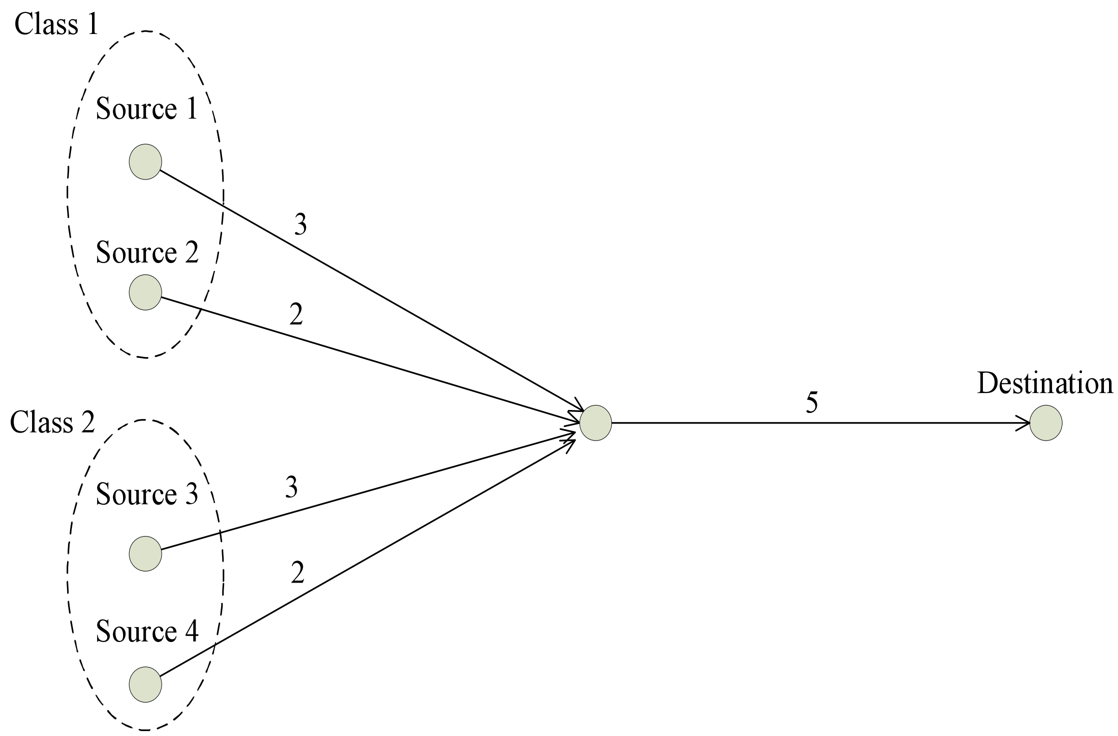
- We have obtained two different distributed algorithms for rate allocation among QoS classes:
  - **primal-dual decomposition**: master primal problem solved with the subgradient update for the aggregate rate carried out independently by each of the links and then, for a given set of aggregate rates, the two resulting basic NUMs are independently solved via the standard dual-based decomposition.
    - ◇ Two levels of decompositions: on the highest level there is a master primal problem, on a second level there is a secondary master dual problem, and on the lowest level the subproblems. There is no explicit signaling required.

Applic. 2: QoS Rate Allocation (X)

- **partial dual decomposition**: master dual problem is solved with the standard price update for each class which is carried out independently by each link and then, for a given set of prices, each source solves its own subproblem as in the canonical NUM and subproblem for the aggregate rate of each class solved independently by each link.
  - ◇ Only one level of decomposition and no explicit signaling is required.

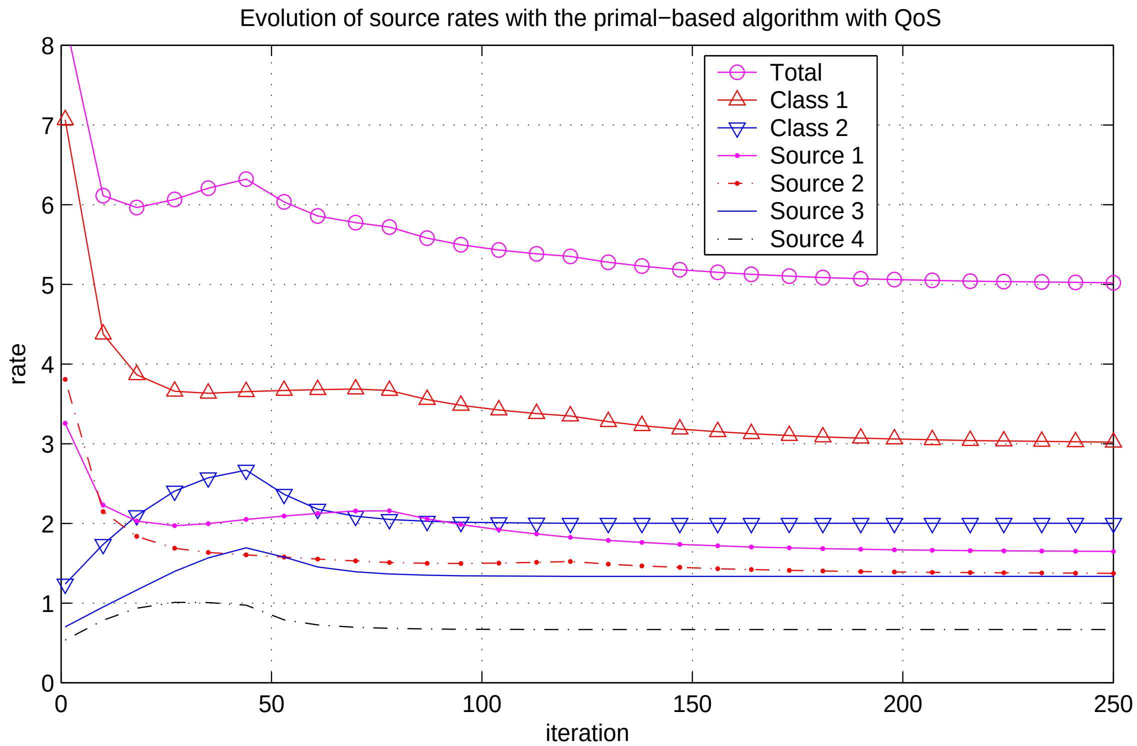
Numerical Results: QoS classes (I)

- Example with two classes (class 1 is aggressive $U_1(x) = 12\log(x)$ and $U_2(x) = 10\log(x)$ and class 2 not aggressive $U_3(x) = 2\log(x)$ and $U_4(x) = \log(x)$):



Numerical Results: QoS classes (II)

- With no QoS control, class 1 gets 4.5 out of the total available rate of 5, leaving class 2 only with a rate of 0.5. This is precisely the kind of unfair behavior that can be avoided with QoS control.
- We limit the rate of each class to 3.



Conclusions

- We have considered the design of networks based on general network utility maximization.
- We have developed a systematic approach to obtain different distributed algorithms for network utility maximization.
- Each distributed algorithm has different characteristics in terms of signalling, speed of convergence, complexity, robustness.
- For each particular application, we can compare the different possibilities and choose the most convenient.

End of Lecture

- For more information visit:

http://www.princeton.edu/~danielp

- Material of this lecture from

Daniel Palomar and Mung Chiang, "Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications," in *Proc. IEEE Infocom, Barcelona*, Spain, April 23 – 29, 2006.