Optimization Decomposition Methods with Applications to Networks

Project Report
Ahmed Almostafa Gashgash

Novermber 15, 2017

Abstract

Solving a problem by decomposition entails breaking it up into smaller problems and solving them in parallel or sequentially. In large complex problems were coupling constraints between the problems variables are present, sequential coordinated solution of the smaller subproblems is needed. The two most common methods for decomposition of such problems are primal decomposition and dual decomposition. A general structure and algorithm for these methods will be presented and analyzed in this paper. These algorithms are shown to be decentralized, therefore having the ability to scale well as the number of subproblems within the original problem increases. An understanding of these structures is crucial in solving the network utility maximization problem (NUM). In this paper, this problem is explained, and a general model is presented. The problem is then solved based on the dual decomposition method alongside sub gradient methods.

1 Introduction

In this paper, the general structure for the primal and dual decomposition methods will be explained, alongside the solution algorithm. These two methods are the pillars for decomposing large optimization problems into subproblems and a master coordinating problem. The network utility maximization problem will then be introduced and a general optimization problem will be formulated and solved based on the dual decomposition method.

2 Decomposition Methods

The underlining principle of decomposition is to to decompose a large-scale problem, whose solution is beyond the reach of standard techniques, into several solvable subproblems, controlled by a higher level coordinating problem. Another important motivation for using decomposition methods is to obtain decentralized solution methods that scale well with the increase of variables. For example, in networks where the number of nodes or sources are large, and an individual source is oblivious to the topology of the network as a whole, a decentralized approach is needed in order for each source to independently perform it's calculations to determine its fair rate within the network.

The majority of existing decomposition methods can be categorized as **primal decomposition**, which decomposes the primal optimization problem, or **dual decomposition methods**, which decomposes the Lagrangian dual problem. In this section, a general framework structure for the problem and the two methods will be introduced based on Boyd et al [1]. A general algorithm for each method will also be presented.

2.1 The General Optimization Problem

Suppose we have a system containing Q subsystems. Each subsystem has $x_i \in R^{n_i}$ private variables, $y_i \in R^{m_i}$ public variables, an objective function (e.g a cost function) $f_i(x_i, y_i) : R^{n_i} \times R^{m_i}$, and a local constraint set $\Lambda_i \subseteq dom(f_i)$. Additional constraints are needed to couple the subsystems.The constraint requires that various subsets of the components of the public variables y_i to be equal [**?**]. To explain this, suppose we have subnetworks within a network that only interact via common links or boundary connections. A link with a fixed capacity C would be shared by different subnetworks, which explains why a subset of the public variables (in this case the capacity C) are equal. To formulate this constraint, let $Y = (y_1, .., y_Q) \in R^m$ be the collection of public variables in the system, with a total of m variables. Note that Y_i is the ith component of Y (scalar), not the list of public vectors y_i (vector) of the subsystem i.

Suppose the system contains N nets, where the public variable on each net is equal. Let $z \in R^N$ be the vector holding the values of the shared public variables, and $E \in R^{m \times N}$ be the matrix where

$$E_{ij} = \begin{cases} 1 & \text{if } Y_i \text{ is not in net j} \\ 0 & \text{otherwise} \end{cases}$$

The rows of matrix E correspond to the list of all public variables, therefore we can be partition

it into blocks $E_i \in R^{m_i \times N}$, where each block only holds the rows that correspond to the public variables of the specific subsystem. The coupling constraint now becomes $y_i = E_i z$. This constraint lets us map the vector z of net variables into the public variables of each subsystem.
The optimization problem could now be formulated as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{Q} f_i(x_i, y_i) \\ \text{s.t} & (x_i, y_i) \in \Lambda_i \\ & y_i = E_i z \\ & i \in (1, ..., Q) \end{array} \tag{1}$$

The vector z is called the primal net variables [1].

2.2 Primal Decomposition

A primal decomposition algorithm decomposes the primal optimization problem. It is appropriate when there exists a coupling variable, that when fixed, the optimization problem could be decoupled into a list of subproblems [2]. From equation (1), if vector z is fixed, and $y_i = E_i z$ is also fixed, the optimization problem is now decentralized, and each subsystem can individually find its optimal solution for x_i.

The optimization problem (1) is now decomposed into subproblems

$$\sigma_i(y_i) = \begin{array}{rl} \text{minimize} & f_i(x_i, y_i) \\ \text{s.t} & (x_i, y_i) \in \Lambda_i \end{array} \tag{2}$$

Where $\sigma_i(y_i)$ is the optimal value of each subproblem evaluated at y_i. Minimizing σ for all values of y_i for a fixed z, is the master problem, and is equivalent to solving the original problem (1). It has the form

$$\text{minimize} \quad \sigma(z) = \sum_{i=1}^{Q} \sigma_i(E_i z) \tag{3}$$

In Boyd [1] and [2] the master problem is solved using a sub gradient method. First a gradient $g_i \in \partial\sigma_i(y_i)$ is found, and the sum of sub gradients adjacent to the net i results in the sub gradient for the master problem. In other words, the subgradient of σ is

$$g = \sum_{i=1}^{Q} E_i^T g_i \in \partial\sigma(z) \tag{4}$$

The general algorithm for the primal decomposition is as follows:

Step 1 Begin with distributing the net variables to all subsystems: $y_i = E_i z, i \in (1, ..., Q)$

Step 2 Solve equation (2) for optimal x_i and sub gradient g_i for all subsystems $i \in (1, ..., Q)$

Step 3 Find the sum of the sub gradients using equation (4)

Step 4 Update the net variable $z := z - \alpha g$, where α is the step size

Step 5 Repeat until termination criteria is met

This algorithm is completely decentralized. As in each subsystem is completely independent of other systems. The only external interaction for the subsystems is with the nets adjacent to it. Therefore the subsystems could still carry out their own calculations with very little knowledge of the rest of the system. The simplicity of this algorithm scales well as the system becomes larger.

2.3 Dual Decomposition

The dual decomposition algorithm decomposes the Lagrangian dual of the optimization problem. It is used when an optimization problem has a coupling constraint that if relaxed, the problem could be decoupled into a list of subproblems [2]. As far as we're concerned, in our problem from equation (1), this constraint is $y = Ez$. Therefore we can define its Lagrangian as follows

$$\begin{aligned} L(x,y,z,\lambda) &= \sum_{i=1}^{Q} f_i(x_i,y_i) + \lambda^T(y - Ez) \\ &= \sum_{i=1}^{Q}(f_i(x_i,y_i) + \lambda_i^T y_i) - \lambda^T Ez \end{aligned} \tag{5}$$

where $\lambda \in R^m$ is the Lagrange multiplier, and λ_i is the subvector of multipliers associated with each subsystem. If we minimize over z, the resulting condition is $E^T v = 0$. Which means that for each net the sum of the Lagrangian multiplier is zero. We now have a set of subproblems defined as follows

$$\begin{aligned} g_i(\lambda_i) = \quad \text{minimize} \quad & f_i(x_i,y_i) + \lambda_i^T y_i \\ \text{s.t} \quad & (x_i,y_i) \in \Lambda_i \end{aligned} \tag{6}$$

The dual of the original problem is as follows

$$\begin{aligned} \text{maximize} \quad & g(\lambda) = \sum_{i=1}^{Q} g_i(\lambda_i) \\ \text{s.t} \quad & E^T\lambda = 0 \end{aligned} \tag{7}$$

To solve this master problem, Boyd [1] suggests using a projected sub gradient method. In this approach, λ is projected onto the feasible set $E^T\lambda = 0$. This projection is done by multiplying by $I - E(E^TE)^{-1}E^T$, where for any $y \in R^m$, $(E^TE)^{-1}Ey$ is the average of the entries in y over each net. The projection is then found by subtracting the average of the other values in the net from each entry of y.

The general algorithm for the dual decomposition is as follows:

Parameters given : Initial price vector λ satisfying $E^T\lambda = 0$

Step 1 Solve equation (6) to obtain x_i, y_i

Step 2 Find the average value for the public variables z over each net: $\hat{z} = (E^TE)^{-1}Ey$

Step 3 Update the price vector: $\lambda := \lambda + \alpha(y - E\hat{z})$

Step 4 Repeat until termination criteria is met

Similar to the primal decomposition algorithm, this is also decentralized. Whereas the ascent gradient was used for the dual decomposition instead of the descent gradient.

3 Decomposition Methods in Network problems

In this section, the problem of network utility maximization (NUM) and the motivation for using decomposition methods will be explained based on chapter (2,3) of Srikant et el [3]. Also, the dual decomposition method explained in Section 2.4 will be used to solve the NUM problem for a decentralized network, as explained by Chiang et el [2].

3.1 Network Utility Maximization

In our discussion, the term utility will be used to mean value. For example, suppose a parent, with finite x dollars, wants to allocate x to his three children. Two of them are toddlers, whilst the third is an indebted senior college student. It's fair to say that the third child would have a greater value or utility for money than the two toddlers. More generally, the utility of an allocation would depend on a set of factors, therefore lets define $U(x)$ as the utility function for an allocation x. Lets assume that $U(x)$ is continuously differentiable. Also, it can be seen that as the allocation x is increased (more resource), there will be no reason for the utility to decrease, therefore we can assume that $U(x)$ is non-decreasing. Also, going back to the previous example, we can see that if allocation x was increased from 2$ to 200$, the student would feel the effect more than if x was increased from 25k$ to 26k$, even though the increase in the later was higher. Therefore we can assume that $U(x)$ is strictly concave.

3.1.1 Problem Formulation

Suppose a large communication network with a set of nodes N and a set of links L. Assume that every $linkl \in L$ has a fixed capacity C_l. A route $r \subset L$ is associated with each node in N, and has a transmission rate of x_r. The utility obtained by the source when using route r at rate x_r is $U_r(x_r)$ [3]. Since a route could be defined over multiple links, we must ensure that the the capacity constraint of each link $l \in L$ is maintained. Therefore, the utility maximization optimization problem can be defined as follows

$$\begin{aligned} \underset{x_r}{\text{maximize}} \quad & \sum_{r \in N} U_r(x_r) \\ \text{s.t} \quad & \sum_{r: l \in r} x_r \leq C_l, \forall l \end{aligned} \tag{8}$$

Note that all rates $x_r \geq 0, \forall r$. Also, the constraint set is closed and bounded. Since the utility function is strictly concave, it is therefore known to have a unique maximum solution over the constraint set.

3.1.2 Solution of the NUM problem using Dual decomposition

A standard distributed algorithm to solve the NUM optimization problem is presented by Chiang [2]. It is based on a dual decomposition. First we define the Lagrangian of problem (8) as follows

$$L(x,\lambda) = \sum_r U_r(x_r) + \sum_l \lambda_l \Big(C_l - \sum_{r:l\in r} x_r \Big) \tag{9}$$

Let λ_l be the link price associated with the flow constraint on l, $\lambda^r = \sum_{l\in r} \lambda_l$ be the aggregate route congestion price of the links used by route r, and $L_r(x_r, \lambda^r) = U_r(x_r) - \lambda^r x_r$ be the Lagrangian to be maximized by the rth route. The Lagrangian in equation (9) can now be written as

$$L(x,\lambda) = \sum_r L_r(x_r, \lambda^r) + \sum_l C_l \lambda_l \tag{10}$$

As a result of the dual decomposition, we can now solve for the unique optimal $x_r^*(\lambda^r) = argmax[L_r(x_r, \lambda^r)]$ for each route separately for a given λ^r. These represent the subproblems of NUM problem. Hence, the solution for the subproblems is unique due to the strict concavity of the utility function.
Define $g_r(v) = L_r(x_r^*(\lambda^r), \lambda^r)$, the master dual problem is

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & g(v) = \sum_r g_r(v) + v^T C \\ \text{s.t} \quad & v \geq 0 \end{aligned} \tag{11}$$

A gradient method is used to update the link cost λ_l, in Chiang [2], the following method is used:

$$\lambda_l(t+1) = \Big[\lambda_l(t) - \alpha \Big(C_l - \sum_{r:l\in r} x_r^*(\lambda^r(t)) \Big) \Big]^+ \tag{12}$$

The dual variable v will converge to the dual optimal v^* as the number of iterations $t \to \infty$.

The dual algorithm to solve the NUM problem presented in (8) is as follows:

Parameters : Each link needs a capacity C_l, and each route needs a utility U_r

Initialize : let t=0, and $\lambda_l(0) \geq 0 \forall l$

Step 2 Each node locally solves $x_r^*(\lambda^r) = argmax[L_r(x_r, \lambda^r)]$, and broadcasts its solution

Step 3 Using equation (12), each link updates its prices, and broadcasts the new price

Step 4 Increment t, and repeat until termination criteria is met

An observation to be made on the dual decomposition method in solving this problem is that it corresponds to allocating resources, in this case flow rate, based on pricing. The master problem in (11) sets the price for the resource for each subproblem, which then calculates the amount of resource to use according to the given price. The master problem of the primal decomposition method however, distributes the resources, giving each subproblem a specific amount of resource to use.

4 Discussion of Surveyed Papers

A good reference for understanding decomposition theory as applied to optimization problems is Boyd et el [1]. In this paper, primal and dual decomposition are presented in their simplest forms and gradually extended to the most general form. First, a decomposable unconstrained optimization problem that splits into two subproblems is explained, and a simple algorithm is presented. This problem is then extended by adding constraints on the variables of the subsystems. Its is shown how these problems are trivially decomposed and could be solved in parallel. In order to fully harness the power of decomposability in solving complex optimization problems, coupling constraints between the sub vectors of the problems variables are introduced. By doing so, these subproblems could not be solved in parallel or independent of the original problem, instead a sequence of smaller problems are solved iteratively. The subproblems were increased, and the coupling constraints were generalized in order to present a more general decomposition structure. In this paper, simple examples were presented to illustrate these ideas.

In Chiang et el [2], applications of the primal and dual decomposition algorithms are discussed. The dual algorithm is implemented to solve the NUM problem, which is the core of the paper. Other decomposition methods spanning of the the primal and dual methods are also discussed. For example, indirect decomposition is presented, where auxiliary variables are inserted in order to provide flexibility in terms of choosing a dual or primal decomposition. In extension to explaining the fundamentals of decomposition theory and its applications, this paper helps the reader to recognize hidden decomposability structures that might not be obvious. This is done by presenting different representations of the same optimization problem and attempting to reveal the decomposability structure of each.

In the third chapter of Srikant et el [3], the primal and dual algorithms were studied with a more rigorous approach. It was shown that both algorithms, when applied to the NUM problem, resulted in a fair and stable allocation. The concept of a Lyapunov function was introduced and used in their analysis to show that the algorithms converged to an optimal solution.

5 Conclusion

The motivation for decomposition methods are many. First it enables us to solve complex problem using standard techniques. It also provides decentralized algorithms that are appreciated in applications where the individual entities lack information about the larger network in which they're in. In this paper, the primal and dual decomposition algorithms were discussed, and the solution of the NUM problem based on the dual decomposition of its optimization problem was presented.

References

[1] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, “Notes on decomposition methods”

[2] D. P. Palomar and Mung Chiang, ”A tutorial on decomposition methods for network utility maximization,” in IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1439-1451, Aug. 2006.

[3] Srinivas Shakkottai and R. Srikant (2008), ”Network Optimization and Control”, Foundations and Trends in Networking: Vol. 2: No. 3, pp 271-379. http://dx.doi.org/10.1561/1300000007