Algorithm and Framework For Distributed Decoupling

Jayanth Krishna Mogali

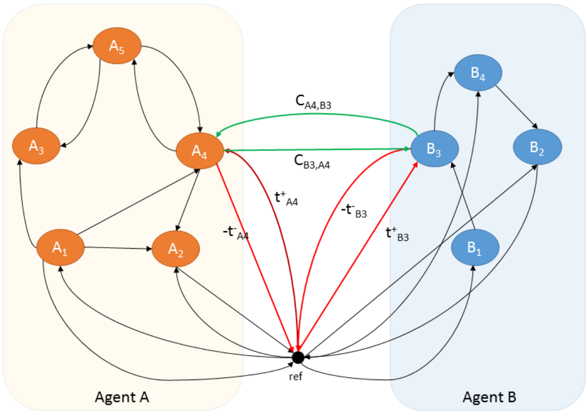
Carnegie Mellon University

jmogali@cs.cmu.edu

July 20, 2015

Overview

Problem Statement



Terminology

All constraints with ref point are indicated with *'t'* variable. Rest with *'C'* variable.

Decoupling Equation

$C_{B3,A4} < t_{A4} - t_{B3} < C_{A4,B3}$

$t^-_{A4} < t_{A4} < t^+_{A4}$

$t^-_{B3} < t_{B3} < t^+_{B3}$

$t^+_{A4} - t^-_{B3} < C_{A4,B3}$

$C_{B3,A4} < t^-_{A4} - t^+_{B3}$

$t^-_{A4} < t^+_{A4}$

$t^-_{B3} < t^+_{B3}$

Decoupling Problem

Max $\Sigma_{agent(i)} \Sigma_j \, t^+_{agent(i),j} - t^-_{agent(i),j}$

s.t.

$t^+_{agent(i),j} - t^-_{agent(i),k} < C_{agent(i),j,\,agent(i),k}$

$C_{agent(i),k,\,agent(i),j} < t^-_{agent(i),j} - t^+_{agent(i),k}$

Where all *C* are minimal intra agent STN constraints

Problem Formulation

The N agent decoupling problem is mathematically represented as follows-:

$$\min \sum_{i=1}^{N} f_i$$

subject to

$$\begin{bmatrix} B_1 & B_2 & B_3 & . & . & . & B_N \\ A_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & A_N \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ X_N \end{bmatrix} \leq \begin{bmatrix} b \\ c_1 \\ c_2 \\ . \\ c_N \end{bmatrix}$$

$$\begin{aligned} M_i &\rightarrow \text{Number of time point variables in agent i} \\ X_i &\in R^{\binom{M_i}{2}} \times 1 \\ \sum_{i=1}^{N} B_i X_i &\leq b \rightarrow \text{Inter-agent coupling constraints} \\ A_i X_i &\leq c_i, i = 1..N \rightarrow \text{Intra-agent triangular consistency constraints} \\ rows(A_i) &\rightarrow \binom{M_i}{3} \end{aligned}$$

Assumptions

- The objective function f_i is private to agent i.
- The number of colluding agents is known to all agents.
- Agent i is aware of matrices A_i and c_i.
- Agent i is only aware of the constraints that it participates in the equation $\sum_{i=1}^{N} B_i X_i \leq b$.
  1. Agent i is aware of matrix B_i.
  2. Suppose agents i and j share a coupling constraint between time point variables $t_{m,i}$ and $t_{n,j}$, then agent i has access to $t^+_{n,j}$ and $t^-_{n,j}$. Similarly agent j has access to $t^+_{m,i}$ and $t^-_{m,i}$.
- Two agents are allowed to communicate directly with each other only if there exists a coupling constraint between them.

Lagrangian Dual Basics

Consider the primal and its Lagrangian dual. If functions $f(x), g(x)$ and set Ω are convex, $h(x)$ is affine, then primal and dual optimum are equal.

Primal Problem

$$\min f(x)$$

sub to

$$g(x) \leq 0$$
$$h(x) = 0$$
$$x \in \Omega$$

Lagrangian Dual Problem

$$\max_{\lambda \geq 0, \nu} L(\lambda, \nu) = \inf_{x} f(x) + \lambda^T g(x) + \nu^T h(x)$$

sub to

$$x \in \Omega$$

Using the above duality relationship, our problem can be formulated as

$$\max_{\nu} -\nu^T b + \sum_{i=1}^{N} \underbrace{\inf_{X_i} f_i(X_i) + \nu^T B_i X_i}_{L_i(v)}$$

sub to

$$A_i X_i = c_i, i = 1, .., N$$

The above problem reduces to an unconstrained optimization problem w.r.t ν.

Lagrange Dual Basics Contd...

The dual objective $L(\nu)$ is a concave function. Algorithm 1 is a simple ascent algorithm maximizing $L(\nu)$ using steepest gradient.

Algorithm 1 Sub-Gradient Algorithm

Parameters: ν^0

```
procedure SUB-GRADIENT ASCENT
    for k ≥ 0 do
        X_i^* = argmin { f_i(X_i) + ν^k B_i X_i } sub to. A_i X_i = c_i, ∀i = 1..N
        Δν = ( Σ_{i=1}^N B_i X_i^* ) − b , Δν is a sub-gradient
        ν^{k+1} = ν^k + c_k Δν, Step-size c_k is computed via line search
        k = k + 1
    end for
end procedure
```

If f(x) is not strictly convex, then L(ν) is not differentiable. Convergence is sub linear.

Self Concordance

1. A function g is said to be self concordant if

$$|g'''(x)| \ \leq 2|g''(x)|^{3/2}$$

2. Strictly convex self concordant (SCSC) functions have positive Hessian, hence 2nd order methods such as Newtons method can be applied.
3. Minimizing a strictly convex self concordant function using Newton's method is polynomial.
4. IPM's for optimizing linear and quadratic objectives add barrier terms such that the augmented function is self concordant.
5. To exploit self concordance, we will modify the objective by augmenting a self concordant barrier i.e. $f(x) + t\phi(x)$.
6. Newton's decrement $\lambda(x) = -((\Delta x_{nt})^T \nabla^2 g^{-1}(x)(\Delta x_{nt}))$ is an provable estimate of $g(x) - p^* < (0.68)^2$ when $\lambda(x) \leq 0.68$.

Problem Reformulation

The decoupling problem is reformulated as follows-:

$$R(t) = \max_{\nu} -\nu^T b + \sum_{i=1}^{N} \underbrace{\inf_{X_i} f_i(X_i) + t\phi_i(X_i) + \nu^T B_i X_i}_{L_i(\nu,t)}, t > 0$$

sub to

$$A_i X_i = c_i, i = 1, .., N$$

- $f_i(X_i) + t\phi_i(X_i) + \nu^T B_i X_i$ is self concordant by construction. However we require $L_i(\nu, t)$ to be self concordant w.r.t ν.
- Turns out that $L_i(\nu, t)$ is related to Legendre transformation of $-f_i(X_i) - t\phi_i(X_i)$ via affine transformation.
- Since by construction $f_i(X_i) + t\phi_i(X_i)$ is self concordant, $L_i(\nu, t)$ has negative definite Hessian (w.r.t ν) and is bounded on X_i. Hence $L_i(\nu, t)$ is also self concordant.
- The optimal value to our problem is $\lim_{t \to 0} R(t)$.

How to Obtain self concordant $\phi(x)$?

- $\forall y_j \in X_i$, we bound $l_j \leq y_j \leq u_j$. Then $\phi_i(x) = -\sum_{y_j \in X_i} log(u_j - y_j) log(y_j - l_j)$.
- We exploit the property that edge weights (treated as variables) in a STN can only reduce by adding constraints.
- Initially each agent omits coupling constraints and computes a minimal STN.
- Let u_j be the value of y_j in the minimal STN of Agent i, say. Then $y_j \leq u_j$ with coupling constraints considered.
- To compute l_j, we consider the set (ω_j) of all closed paths of length 2 and 3 involving y_j in the minimal STN. All such path costs must be ≥ 0.
- Let the m^{th} element of ω_j be $[c_1, c_2, y_j]$. Then

$$d_m = c_1 + c_2 \tag{1}$$

$$y_j \geq -d_m \tag{2}$$

$$1 \ \& \ 2 \implies y_j \geq \max\{-d_m\} \forall m$$

Optimization Algorithm [Necoara et.al, 2009]

We use Newton's method for optimizing $L_i(\nu, t)$ w.r.t ν. The Newton's step is given by

$$
\begin{aligned}
\Delta\nu &= \left(\nabla^2_\nu L(\nu, t)\right)^{-1} \nabla_\nu L(\nu, t) \\
\nabla_\nu L(\nu, t) &= b + \sum_{i=1}^{N} \nabla_\nu L_i(\nu, t) \\
&= b - \sum_{i=1}^{N} B_i X_i(\nu, t) \text{ , where} \\
X_i(\nu, t) &= \arg\min f_i(X_i) + t\phi_i(X_i) + \nu^T B_i X_i \text{ s.t } A_i X_i = c_i \\
\left(\nabla^2_\nu L(\nu, t)\right) &= \sum_{i=1}^{N} \left(\nabla^2_\nu L_i(\nu, t)\right) \\
\text{Let } H_i(\nu, t) &= \nabla^2_\nu f_i(X_i(\nu, t)) + t\nabla^2_\nu \phi_i(X_i(\nu, t)), \text{ then}
\end{aligned}
$$

$$
\left(\nabla^2_\nu L_i(\nu, t)\right) = B_i \left[H_i(\nu, t)^{-1} - H_i(\nu, t)^{-1} {A_i}^T \left(A_i H_i(\nu, t)^{-1} {A_i}^T \right)^{-1} A_i H_i(\nu, t)^{-1} \right] {B_i}^T
$$

Optimization Algorithm Contd...

Initialization of the path following algorithm,

Algorithm 2 Initialization Of Path

Input: $\nu^0, t^0, \epsilon > 0, k = 0$

procedure PATH INITIALIZATION
  while (1) **do**
    Compute $X_i(\nu^k, t^0) \forall i.$ $\delta_k = \delta(\nu^k, t^0)$ ▷ X_i are computed separately by each agent subject to $A_i X_i = c_i$
    if $\delta_k \leq \epsilon$ **then**
      return ν^k
    end if
    $\nu^{k+1} = \nu^k + \sigma \Delta \nu(\nu^k, t_0)$ ▷ σ is a suitable step length
    k = k + 1
  end while
end procedure

Newton's Decrement- $\delta(\nu^k, t^0) = \frac{\alpha}{2} \sqrt{\nabla_\nu L(\nu^k, t)^T \left(\nabla_\nu^2 L(\nu^k, t)\right)^{-1} \nabla_\nu L(\nu^k, t)}$

Computation of $\delta(\nu^k, t^0)$ is through a distributed summation.

Optimization Algorithm Contd...

Path following algorithm initialized with previously found ν.

Algorithm 3 Path Following

Input: $\nu^0, t^0, \epsilon > 0$ satisfying $\delta(\nu^0, t^0) \leq \epsilon, 0 < \tau < 1$

procedure PATH FOLLOWING
- **while** (1) **do**
  - **if** $t^k N_\phi \leq \epsilon$ **then**
    - break
  - **end if**
  - $t^{k+1} = \tau t^k$
  - Initialize $\nu = \nu^k, t = t^{k+1}, \delta = \delta(t^{k+1}, \nu^k)$
  - **while** $\delta \geq \epsilon$ **do**
    - Compute $X_i = X_i(\nu, t) \forall i$. ▷ X_i computed separately by each agent subject to $A_i X_i = c_i$
    - $\nu^+ = \nu + \sigma \Delta(\nu, t)$ ▷ σ is a suitable step length
    - $\delta^+ = \delta(\nu^+, t)$, Update $\nu = \nu^+, \delta = \delta^+$ ▷ δ^+ computed via distributed summation
  - **end while**
  - $\nu^{k+1} = \nu$ and $X_i^{k+1} = X_i \, \forall i$ ▷ X_i^{k+1} is feasible to all agents respecting coupling constraints
- **end while**

end procedure

Network Communication

- Recall that communication over the N/W is required for computation of δ.
- Specifically $\nabla_\nu L(\nu, t)$ and $\nabla^2_\nu L(\nu, t)$ needs to be computed distributively. Computation of σ ??
- Depending on the project requirements, different network topologies can be devised.
- For minimizing N/W communication a minimum spanning tree can be constructed in a distributed fashion.
- This N/W topology allows only agents sharing a coupling constraint to communicate with each other.

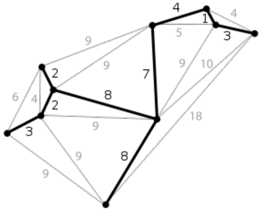


Figure : Spanning tree. Image taken from [Wikipedia]

Some Thoughts About The Method

Agent Coordination

Agents must coordinate in some fashion for creation of coupling matrix B. Specifically, each agent must assume responsibility for a specific row in B.

Feasible Solutions

After every outer iteration we get a feasible solution to our original problem.

Infeasibility Check

We check slack variables (belonging to B matrix) in the converged solution of path initialization. Since we are guaranteed a unique minimizer, checking for negativity of slack variables suffices.

Number of Outer Iterations

The number of outer iterations is given by $log_{\tau} \frac{\epsilon}{N_{\phi} t_0}$, $\epsilon \rightarrow$ level of accuracy.

$N_{\phi} = \sum_{i=1}^{N} N_i$, $N_i \rightarrow$ no. of time point variables in agent i.

Some Thoughts About The Method Contd...

Choosing σ for line search

Usually line search techniques are used such Armijo's criteria. This can reveal the private objective values of the agents. There may however be a scheme to select σ using just δ value. If agents can share the functional values, we may obtain faster convergence.

Distributed Computation

During distributed summation of $\nabla_\nu L(\nu, t)$ there is a chance that some values of time points may be revealed to a non associated agent. To avoid this we multiply each row in matrix B by a random number that is known only to the agents concerned.

Complexity

Polynomial!

References

Necoara, I and Suykens, JAK (2009)
Interior-point lagrangian decomposition method for separable convex optimization
Journal of Optimization Theory and Applications Springer pp 567–588

Minimum Spanning trees
https://en.wikipedia.org/wiki/Minimum_spanning_tree 12(3), 45 – 678.