Real-time GDA

ABSTRACT

ACM Reference format:
. 2016. Real-time GDA. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 5 pages.
DOI: 10.475/123_4

1 SINGLE DAG QUERY

1.1 Model

1.1.1 Wide Area Network. Let $\mathcal{S}$ be the set of sites that hold data and run tasks, and $\mathcal{L}$ be the set of directed edges that represent inter-site links. For each inter-site WAN link $(i, j) \in \mathcal{L}$, let B_{ij} be the bandwidth and C_{ij} be the cost to transfer one unit of data from site i to site j.

Assumption 1. *The bandwidths are stable within the time frame of real-time data analytics.*

For the analysis we make the following assumption:

Assumption 2. *Data transfers on a particular link are non-overlapping. (To do: Show that this assumption does not add suboptimality to our objective, i.e., any overlapping schedule can be transformed into a non-overlapping schedule without any loss to the optimal value.)*

1.1.2 DAG of tasks. We define a stage in the DAG as a group of tasks that have the same input data dependencies. Let $\mathcal{T}$ be the set of stages and $\mathcal{D}$ be the directed set of edges that represent stage dependencies for the DAG, respectively. Some of the stages do not have any incoming edges and so represent the locations of the raw input data. Let $\mathcal{R} \subset \mathcal{T}$ be the set of stages for the raw input data. The final stage in the DAG does not have any outgoing edges and so represent the final destination of the query's response denoted by $F \in \mathcal{T}$. Each stage dependency $(k, l) \in \mathcal{D}$ also has a corresponding amount of data D^{kl} that must be transfered from stage k to stage l.

Assumption 3. *A stage must have completely received all of its input data before it can process and start transferring data to another stage.*

The DAG also has a start time T_0 and a finish time T_f for which the schedule of data transfers must respect.

1.1.3 Stage assignment decisions. We model each stage as a group of tasks that must be scheduled to run together at the same site. The decision of stage k to be placed at site i is represented by the binary decision variable $x_i^k \in \{0, 1\}$. This means that for any directed edge $(k, l) \in \mathcal{D}$, then $D^{kl} x_i^k x_j^l$ of data will be transfered across link $(i, j) \in \mathcal{L}$. Note that the set of stages $\mathcal{R}$ and F which respectively represent the raw data and the DAG's final stage have decision variables which are preset according to its site-wise distribution and final stage location.

DOI: 10.475/123_4

1.2 Stage Assignment Problem

1.2.1 Problem Statement. Given the available links for each dependency we have the following problem to minimize the cost of the WAN by deciding where to place each stage:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(k,l)\in\mathcal{D}} \sum_{(i,j)\in\mathcal{L}} C_{ij} D^{kl} x_i^k x_j^l \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} && \text{(1a)} \\ & x_i^k \in \{0,1\} \quad \forall i \in \mathcal{S}, k \in \mathcal{T} && \text{(1b)} \end{aligned}$$

Note that for any stage $k \in \mathcal{R} \cup F$, the placement decision is preset.

1.2.2 Problem Reformulation. Since $x_i^k \in \{0, 1\}$ and $x_j^l \in \{0, 1\}$, then $x_i^k x_j^l = 1$ iff $\max\{0, x_i^k + x_j^l - 1\} = 1$, and $x_i^k x_j^l = 0$ iff $\max\{0, x_i^k + x_j^l - 1\} = 0$. This means that $x_i^k x_j^l$ and $\max\{0, x_i^k + x_j^l - 1\}$ are equivalent for the binary decisions and the later is a convex function where the former is not. Therefore, (1) can be reformulated with a convex objective function:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(k,l)\in\mathcal{D}} \sum_{(i,j)\in\mathcal{L}} C_{ij} D^{kl} \max\{0, x_i^k + x_j^l - 1\} \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} && \text{(2a)} \\ & x_i^k \in \{0,1\} \quad \forall i \in \mathcal{S}, k \in \mathcal{T} && \text{(2b)} \end{aligned}$$

The max function can be further simplified with the use of an auxiliary variable z_{ij}^{kl} and assume that $C_{ij} D^{kl} > 0$ to reformulate (2):

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \sum_{(k,l)\in\mathcal{D}} \sum_{(i,j)\in\mathcal{L}} C_{ij} D^{kl} z_{ij}^{kl} \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} && \text{(3a)} \\ & x_i^k \in \{0,1\} \quad \forall i \in \mathcal{S}, k \in \mathcal{T} && \text{(3b)} \\ & z_{ij}^{kl} \geq x_i^k + x_j^l - 1 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} && \text{(3c)} \\ & z_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} && \text{(3d)} \end{aligned}$$

If x_i^k can be relaxed from a binary variable to a nonnegative variable and still guarantee a binary optimal solution then (3) could be simply reformulated as a linear program.

Conjecture 1.1. *We can guarantee that relaxing the binary constraint* (3b) *to a nonnegativity constraint in* (3) *always has an optimal solution which is binary. One way to prove this by proving that the constraints* (3a) *and* (3c) *form a totally unimodular matrix. Another method is to prove from the optimality conditions. (To do: Prove or disprove this conjecture.)*

(To do: What are the optimality conditions?)

1.2.3 Optimality conditions of the linear program. We take (3) and relax the binary constraint to nonnegative:

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{(k,l)\in\mathcal{D}} \sum_{(i,j)\in\mathcal{L}} C_{ij} D^{kl} z_{ij}^{kl}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} \tag{4a}$$

$$x_i^k \geq 0 \quad \forall i \in \mathcal{S}, k \in \mathcal{T} \tag{4b}$$

$$z_{ij}^{kl} \geq x_i^k + x_j^l - 1 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{4c}$$

$$z_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{4d}$$

Start with the KKT conditions for optimality using the following dual variables $(\mathbf{u}, \boldsymbol{\phi}, \mathbf{v}, \boldsymbol{\psi})$ and assume that $C_{ij}D^{kl} > 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$:

$$C_{ij} D^{kl} - v_{ij}^{kl} - \psi_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5a}$$

$$-u^l - \phi_j^l + \sum_{i:(i,j)\in\mathcal{L}} \sum_{k:(k,l)\in\mathcal{D}} v_{ij}^{kl} + \sum_{n:(j,n)\in\mathcal{L}} \sum_{m:(l,m)\in\mathcal{D}} v_{jn}^{lm} = 0 \quad \forall j \in \mathcal{S}, \forall l \in \mathcal{T} \tag{5b}$$

$$\phi_j^l x_j^l = 0 \quad \forall j \in \mathcal{S}, \forall l \in \mathcal{T} \tag{5c}$$

$$(x_i^k + x_j^l - 1 - z_{ij}^{kl}) v_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5d}$$

$$z_{ij}^{kl} \psi_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5e}$$

$$\phi_j^l \geq 0 \quad \forall j \in \mathcal{S}, \forall l \in \mathcal{T} \tag{5f}$$

$$v_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5g}$$

$$\psi_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5h}$$

$$\sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} \tag{5i}$$

$$x_i^k \geq 0 \quad \forall i \in \mathcal{S}, k \in \mathcal{T} \tag{5j}$$

$$z_{ij}^{kl} \geq x_i^k + x_j^l - 1 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5k}$$

$$z_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{5l}$$

Using the first two equations to solve for ψ_{ij}^{kl} and ϕ_j^l and plug them into the rest, we have:

$$x_j^l \left(u^l - \sum_{i:(i,j)\in\mathcal{L}} \sum_{k:(k,l)\in\mathcal{D}} v_{ij}^{kl} - \sum_{n:(j,n)\in\mathcal{L}} \sum_{m:(l,m)\in\mathcal{D}} v_{jn}^{lm} \right) = 0 \quad \forall j \in \mathcal{S}, \forall l \in \mathcal{T} \tag{6a}$$

$$(x_i^k + x_j^l - 1 - z_{ij}^{kl}) v_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6b}$$

$$z_{ij}^{kl} \left(C_{ij} D^{kl} - v_{ij}^{kl} \right) = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6c}$$

$$\sum_{i:(i,j)\in\mathcal{L}} \sum_{k:(k,l)\in\mathcal{D}} v_{ij}^{kl} + \sum_{n:(j,n)\in\mathcal{L}} \sum_{m:(l,m)\in\mathcal{D}} v_{jn}^{lm} \geq u^l \quad \forall j \in \mathcal{S}, \forall l \in \mathcal{T} \tag{6d}$$

$$v_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6e}$$

$$C_{ij} D^{kl} \geq v_{ij}^{kl} \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6f}$$

$$\sum_{i\in\mathcal{S}} x_i^k = 1 \quad \forall k \in \mathcal{T} \tag{6g}$$

$$x_i^k \geq 0 \quad \forall i \in \mathcal{S}, k \in \mathcal{T} \tag{6h}$$

$$z_{ij}^{kl} \geq x_i^k + x_j^l - 1 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6i}$$

$$z_{ij}^{kl} \geq 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D} \tag{6j}$$

Conditions (6c) and (6j) mean that:
If $z_{ij}^{kl} > 0$ then $v_{ij}^{kl} = C_{ij} D^{kl} \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
Conditions (6b) and (6i) mean that:
If $z_{ij}^{kl} > x_i^k + x_j^l - 1$ then $v_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
The two previous results and conditions (6i), (6j) along with the assumption $C_{ij}D^{kl} > 0$ mean that:
$z_{ij}^{kl} = \max\{0, x_i^k + x_j^l - 1\} \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
Then with (6e) and (6f):
If $x_i^k + x_j^l < 1$, then $v_{ij}^{kl} = 0 \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
If $x_i^k + x_j^l = 1$, then $v_{ij}^{kl} \in [0, C_{ij}D^{kl}] \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
If $x_i^k + x_j^l > 1$, then $v_{ij}^{kl} = C_{ij}D^{kl} \quad \forall (i,j) \in \mathcal{L}, \forall (k,l) \in \mathcal{D}$.
Condition (6a) means that:
If $x_j^l > 0$, then $u^l = \sum_{i:(i,j)\in\mathcal{L}} \sum_{k:(k,l)\in\mathcal{D}} v_{ij}^{kl} + \sum_{n:(j,n)\in\mathcal{L}} \sum_{m:(l,m)\in\mathcal{D}} v_{jn}^{lm}$ $\forall j \in \mathcal{S}, \forall l \in \mathcal{T}$.
From the above result, (6g), and (6d):
$u^l = \min_j \left\{ \sum_{i:(i,j)\in\mathcal{L}} \sum_{k:(k,l)\in\mathcal{D}} v_{ij}^{kl} + \sum_{n:(j,n)\in\mathcal{L}} \sum_{m:(l,m)\in\mathcal{D}} v_{jn}^{lm} \right\}$ $\forall l \in \mathcal{T}$.

1.3 Data Transfer Scheduling Problem

The goal is to find a schedule π given stage assignment $\mathbf{x}$ that satisfies the stage dependencies $\mathcal{D}$ from the DAG, the query's start time T_0 and deadline T_f. Let t_0^{kl} and t_f^{kl} be the start and finish times to transfer data from stage k to stage l. Therefore, the schedule must satisfy:

$$\max_{k:(k,l)\in\mathcal{D}} t_f^{kl}(\pi(\mathbf{x})) \leq \min_{m:(l,m)\in\mathcal{D}} t_0^{lm}(\pi(\mathbf{x})) \quad \forall l \in \mathcal{T} \tag{7a}$$

$$\min_{(k,l)\in\mathcal{D}} t_0^{kl}(\pi(\mathbf{x})) \geq T_0 \tag{7b}$$

$$\max_{(k,l)\in\mathcal{D}} t_f^{kl}(\pi(\mathbf{x})) \leq T_f \tag{7c}$$

The bandwidth constraints on each link must also be satisfied. Let d_{ij}^{kl} be the total amount of time (not necessarily contiguous) that link (i,j) is devoted to transfer data from stage k to stage l. Therefore, the schedule must satisfy:

$$d_{ij}^{kl} \geq \frac{D^{kl}}{B_{ij}} \quad \forall (i,j,k,l) : (i,j) \in \mathcal{L}, (k,l) \in \mathcal{D}, x_i^k = 1, x_j^l = 1 \tag{8}$$

Conjecture 1.2. *[3] gives an optimal scheduling algorithm for a DAG on a single machine called Latest Deadline First. With some effort, this can be proved for our case when the stage assignments have already been decided (multiple single machines with a DAG relating them all).* *(To do: Prove or disprove this conjecture.)*

(To do: What are the necessary and sufficient conditions for a feasible schedule?)

1.4 Joint optimization and Algorithm Design

The Stage Assignment Problem chooses the stage assignment to minimize WAN usage cost while the Data Transfer Scheduling Problem decides whether a stage assignment can be feasibly scheduled. Therefore, the joint optimization problem becomes finding a feasible stage assignment that minimizes the WAN usage cost.

Conjecture 1.3. *The Data Transfer Scheduling Problem can decide whether a given stage assignment has a feasible schedule or not. If infeasible, it also finds the sequence of bottlenecking data transfers that make it not satisfy the query deadline constraint. This infeasibility can be relayed back the Stage Assignment Problem's objective function with a carefully chosen dual cost. (To do: Prove or disprove this conjecture.)*

(To do: How to incorporate the necessary and feasibility conditions for a feasible schedule into the optimality conditions?)
(To do: The algorithm should take advantage of sparsity.)

1.5 Numerical Simulations

1.5.1 Example setup. (To do: Make an example that incorporates scheduling line contentions.)

1.5.2 Optimal Solution. (To do: Sensitivity analysis of optimal solution.)

2 SINGLE MAPREDUCE QUERY

2.1 Model

Let $\mathcal{L}$ be the set of sites that holds data and runs tasks. For each inter-site WAN link, let B_{ij} be the bandwidth from site $i \in \mathcal{L}$ to site $j \in \mathcal{L}$. We assume that the bandwidths are stable within the time frame of doing real-time data analytics.

We perform the map tasks at the sites that contain the associated data and denote D_i as the output data from all of the map tasks at site i. The fraction of reduce tasks assigned to site j is denoted as r_j which is also the fraction of all other sites' data that must be transfered through the WAN to j. This means that the total WAN usage is for a given task distribution r is:

$$\sum_i \sum_{j\neq i} D_i r_j \tag{9}$$

which can be rearranged to:

$$\sum_i \sum_{j\neq i} D_i r_j = \sum_i D_i \sum_j r_j - \sum_j D_j r_j = \sum_i D_i - \sum_j D_j r_j \tag{10}$$

$$= \sum_j D_j(1-r_j) \tag{11}$$

If we are given a start time s after the map steps are all completed and a data shuffle deadline t for the reduce tasks, then the completion time is bounded as such:

$$s + \max_{(i,j):j\neq i} \left\{ \frac{D_i r_j}{B_{ij}} \right\} \leq t \tag{12}$$

which is caused by the heterogeneous WAN bandwidth and has a bottlenecking link(s). We assume that $t > s$.

2.2 Problem

Minimize WAN usage. When minimizing WAN usage for a MapReduce data shuffle we have the following optimization problem:

$$\min_r \quad \sum_j D_j(1-r_j) \tag{13a}$$

$$\text{s.t.} \quad \sum_j r_j = 1 \tag{13b}$$

$$r_j \geq 0 \quad \forall j \tag{13c}$$

Feasibility to meet deadline. We want to find a feasible r so that the data shuffle finishes at or before t:

$$\text{find } r \text{ s.t.} \quad \frac{D_i r_j}{B_{ij}} \leq t - s \quad \forall (i,j) : j \neq i \tag{14a}$$

$$\sum_j r_j = 1 \tag{14b}$$

$$r_j \geq 0 \quad \forall j \tag{14c}$$

Minimize WAN usage given a shuffle deadline. When minimizing WAN usage for a MapReduce data shuffle for a given deadline t we have the following optimization problem:

$$\min_r \quad \sum_j D_j(1-r_j) \tag{15a}$$

$$\text{s.t.} \quad \frac{D_i r_j}{B_{ij}} \leq t - s \quad \forall (i,j) : i \neq j \tag{15b}$$

$$\sum_j r_j = 1 \tag{15c}$$

$$r_j \geq 0 \quad \forall j \tag{15d}$$

2.3 Optima

Since Problem (15) is a convex optimization problem (linear program), we look at the Karush-Kuhn-Tucker (KKT) conditions for optimality using the following dual variables $(\mu_{ij}, \theta, \lambda_j) : \forall (i,j), i \neq j$:

$$-D_j + \sum_{i \neq j} \frac{D_i}{B_{ij}} \mu_{ij} + \theta - \lambda_j = 0 \quad \forall j \tag{16a}$$

$$\mu_{ij} \left(\frac{D_i}{B_{ij}} r_j - (t - s) \right) = 0 \quad \forall (i,j) : i \neq j \tag{16b}$$

$$r_j \lambda_j = 0 \quad \forall j \tag{16c}$$

$$\mu_{ij} \geq 0 \quad \forall (i,j) : i \neq j \tag{16d}$$

$$\lambda_j \geq 0 \quad \forall j \tag{16e}$$

$$\frac{D_i}{B_{ij}} r_j - (t - s) \leq 0 \quad \forall (i,j) : i \neq j \tag{16f}$$

$$\sum_j r_j - 1 = 0 \tag{16g}$$

$$r_j \geq 0 \quad \forall j \tag{16h}$$

where (16a) are the first stationary conditions, (16b) (16c) are the complimentary slackness conditions, (16d) (16e) are the dual feasibility conditions, and (16f) (16g) (16h) are the primal feasibility conditions.

Notice that for every site j, (16f) implies that there could be bottlenecking link if $\max_{i \neq j} \{D_i / B_{ij}\} > t - s$. There would not be a bottlenecking link only if making $r_j = 1$ meets the deadline. Let us denote

$$U_j := \max_{i \neq j} \{D_i / B_{ij}\} \tag{17}$$

as the maximum upload time for site j if $r_j = 1$, and denote

$$b_j := \arg\max_{i \neq j} \{D_i / B_{ij}\} \tag{18}$$

as the set of bottlenecking data sources. If for any $i \notin b_j$, then (16f) is satisfied if it satisfied for b_j and also $D_i / B_{ij} < t - s$. This fact and (16b) implies that $\mu_{ij} = 0$ if $i \notin b_j$. Now the KKT conditions (16) have redundant conditions that can be eliminated to:

$$-D_j + U_j \sum_{i \in b_j} \mu_{ij} + \theta - \lambda_j = 0 \quad \forall j \tag{19a}$$

$$\mu_{ij} \left(U_j r_j - (t - s) \right) = 0 \quad \forall (i,j) : i \in b_j \tag{19b}$$

$$r_j \lambda_j = 0 \quad \forall j \tag{19c}$$

$$\mu_{ij} \geq 0 \quad \forall (i,j) : i \in b_j \tag{19d}$$

$$\lambda_j \geq 0 \quad \forall j \tag{19e}$$

$$U_j r_j - (t - s) \leq 0 \quad \forall (i,j) : i \in b_j \tag{19f}$$

$$\sum_j r_j - 1 = 0 \tag{19g}$$

$$r_j \geq 0 \quad \forall j \tag{19h}$$

We have three cases for each r_j:

(1) $r_j = (t - s)/U_j$. Since $t - s > 0$ and $U_j > 0$, then $r_j > 0$. Then (19c) results in $\lambda_j = 0$. In this case (19a) turns into:

$$\theta = D_j - U_j \sum_{i \in b_j} \mu_{ij} \tag{20}$$

Since $\sum_{i \in b_j} \mu_{ij} \geq 0$ from (19d), then this means that $\theta \leq D_j$.

(2) $0 < r_j < (t - s)/U_j$. Then (19c) results in $\lambda_j = 0$ and (19b) results in $\mu_{ij} = 0 : \forall i \in b_j$. In this case (19a) turns into:

$$\theta = D_j \tag{21}$$

(3) $r_j = 0$. Then (19b) results in $\mu_{ij} = 0 : \forall i \in b_j$. In this case (19a) turns into:

$$\theta = D_j + \lambda_j \tag{22}$$

Since (19e), then $\theta \geq D_j$.

Since θ can only be a single value, the above cases give a natural ordering. For a particular θ: if $D_j > \theta$, then Case 1 applies; if $D_j < \theta$ then Case 3 applies; if $D_j = \theta$ then Cases 1,2, or 3 could apply. This also means that we can restrict θ to the set $\{D_j : \forall j\}$ without restricting the solution space of the task placements $r_j : \forall j$.

Also, if all sites are in Case 1 and we observe that from (19g) $\sum_j \frac{1}{U_j} < \frac{1}{t-s}$, then the deadline t is too soon and the problem is infeasible. Let us denote the lower bound on t as:

$$\underline{t} := s + \frac{1}{\sum_j \frac{1}{U_j}} \tag{23}$$

On the other hand, let $l := \arg\max_j \{D_j\}$ and if $U_l < t - s$, then $r_l = 1$ and $r_j = 0 : \forall j \neq l$. Let us denote the upper bound on t as:

$$\bar{t} := s + U_l \tag{24}$$

If $t \geq \bar{t}$, then $r_l = 1$, $r_j = 0 : \forall j \neq l$, and the WAN usage is $\sum_i D_i - D_l$.

If $t = \underline{t}$, then $r_j = (\underline{t} - s)/U_j : \forall j$ and the WAN usage is $\sum_i D_i - (\underline{t} - s) \sum_j (D_j / U_j)$.

Note that if either the maximum deadline range

$$\bar{t} - \underline{t} = U_l - \frac{1}{\sum_j \frac{1}{U_j}} \tag{25}$$

$$\tag{26}$$

or the maximum WAN savings

$$D_l - \frac{1}{\sum_j \frac{1}{U_j}} \sum_j \frac{D_j}{U_j} \tag{27}$$

has significant cost, then developing an optimization algorithm is worthwhile.

2.4 Algorithm

We have the following algorithm:

(1) **Initialize:**
- Order D_j in descending order and relabel the indexes for this ordering.
- Set $y := 1$, $k := 1$, and $r_j := 0 : \forall j$
- For each site j, set:
  $U_j := \max_{i \neq j} \{D_i / B_{ij}\}$
  $b_j := \arg\max_{i \neq j} \{D_i / B_{ij}\}$.

(2) **Test for feasibility:**
- If $t \leq s + \frac{1}{\sum_j \frac{1}{U_j}}$, then stop because the problem is infeasible.

(3) **Process:**
- Replace $r_k := \min\{y, (t - s)/U_k\}$.
- Update $y := y - r_k$ and $k := k + 1$.
- If $y \leq 0$ or $k > |\mathcal{L}|$, then stop and output $r_j : \forall j$. Otherwise repeat Step (3).

REFERENCES

[1] Dimitri P Bertsekas. 1999. *Nonlinear programming.* Athena scientific Belmont.
[2] CE Gounaris and CA Floudas. 2008. Convexity of products of univariate functions and convexification transformations for geometric programming. *Journal of Optimization Theory and Applications* 138, 3 (2008), 407–427.

[3] Eugene L Lawler. 1973. Optimal sequencing of a single machine subject to precedence constraints. *Management science* 19, 5 (1973), 544–546.
[4] Ray PöRn, Kaj-Mikael BjöRk, and Tapio Westerlund. 2008. Global solution of optimization problems with signomial parts. *Discrete optimization* 5, 1 (2008), 108–120.

APPENDIX