

Uncertain Networks Optimization Based on Random ADD Algorithm

Gu Daming

Yancheng Institute of Industry Technology, Yancheng, Jiangsu, 224005, China

Email: gdm9418@126.com

Abstract—In order to solve the traditional network optimization problem, the dual gradient descent algorithm is adopted. Although the algorithm can be realized by a distributed method, its convergence rate is slow. Acceleration dual decline algorithm mainly uses the distributed calculation of the approximative Newton step to improve its convergence rate. Due to the uncertainty characteristic of communication networks, its convergence is difficult to be guaranteed in the presence of restriction uncertainty. An ADD algorithm based on random modality is proposed to resolve the network optimization problem. It is proved theoretically that ADD algorithm can converge to an error neighborhood of optimum value with higher probability when the uncertainty mean square error is bounded; the ADD algorithm can also converge to the optimum value with higher probability when there is the more stringent restriction condition. Experiment results demonstrate the convergence rate of the proposed algorithm is more than 100 times faster than the random gradient descent algorithm.

Index Terms—Network Optimization; Acceleration Dual Decline Algorithm; Random ADD; Convergence Rate

I. INTRODUCTION

In order to optimize the distributed network traffic, a common approach is using the dual subgradient descent algorithm in the dual equation domain. Since these algorithms can be realized by a distributed manner, they are very attractive algorithm for looking for the optimal working point in wired or wireless communication networks. However, a notable feature of sub-gradient descent algorithm is very slow convergence rate. On the other hand, Newton method can get faster (square) convergence rate, but it cannot be realized with a distributed manner. In order to combines with the advantages of distributed method and square convergence, various modified Newton methods are proposed to improve the network optimization.

Due to the global cooperation, some early researches in this area are not distributed. In order to overcome the drawback, an approximation method of inverse Hessian matrix (with an approximate inverse of the diagonal elements) is proposed, i.e., the Newton step can be calculated by using Newton method of consistent iterative approximation or using the acceleration dual decrease (ADD) series algorithm based on the Taylor expansion. ADD series algorithm can be distinguished by labeling

the number N , which demonstrates the information of each node mainly collected by the N -hop in the center while establishing the approximation local Newton step. ADD- N algorithm has been proved that it can reach the square convergence rate, and can reduce the time of searching the optimal operating point, where the number of communication instance are adopted to measure the time of finding the optimal operating point and its time is less than 100 times slower than the sub-gradient descent algorithm.

Sub-gradient descent algorithm and distributed approximate Newton algorithm have a common defect that both of them need to know the external global rate in calculating the descent step, but these rates are impossible to beforehand know in a communication network. In addition, the data packet is received from the application layer and we also hope that the network adaptively acquires the average packet rate, while in the actual network we get real-time packet rate rather than the average packet rate.

When we only know the real-time rate rather than the average rate, the pressure algorithm is adopted to find the optimal work point. The algorithm can be interpreted as random sub-gradient descent algorithm which is used in dual function domain. The interpretation is based on two basic principles as follows: 1) the real-time rate can be seen as an unbiased estimation of the average rate; 2) the sub-gradient of dual function is a linear function of external global average rate. Therefore, we can replace the average rate with the real-time rate, which is to establish unbiased gradient estimation so as to obtain the minimum value of the dual function on average meaning. The formal interpretation can be used to ensure that the algorithm almost converges to the optimal operation point.

The paper mainly builds upon the following theoretical research, Newtonian approximation step of the ADD- N algorithm is also a linear function of the external global arriving rate vector. Therefore, ADD series algorithms are random versions that can be used to solve the single commodity network traffic optimization problem, which randomness is mainly reflected in the incomplete support vector knowledge. In Section two, the traffic optimization of the random single commodity network is firstly introduced, and the paper mainly focuses on the uncertain environment model. We consider the uncertainty characteristic of the second moment in the case of

constant upper bound. In addition, we also will be assessed a tame model that assumes that the uncertainty variance is proportional to the gradient norm of dual function. Then, the third section mainly reviews the ADD series algorithm, and describes their corresponding stochastic algorithm. The fourth section analyzes the convergence performance of random ADD algorithm. In addition, we prove the supermartingale convergence theorem and apply it to analyze the two uncertainty modes in paper. We also prove that the algorithm almost certainly converges to the optimal operation point when the uncertainty variance is proportional to the gradient norm of dual function. In Section five, we mainly describe the numerical experiment results of ADD-1 algorithm, which the algorithm uses one-hop neighbor information to approximate the Newton step. We claim that the convergence rate of the ADD-1 algorithm is more than 100 times faster than the random gradient descent algorithm, and reach the convergence rate of standard Newton algorithm.

II. OPTIMIZATION MODEL OF UNCERTAIN NETWORKS

Consider a network denoted by a directed graph $\mathcal{G}=(\mathcal{N},\mathcal{E})$, where nodes set is $\mathcal{N}=\{1,\dots,n\}$ and the edge set is $\mathcal{E}=\{1,\dots,E\}$. The network is planned to support single information stream, where the input rate in source node is $b^i > 0$ and the output rate in target node is $b^i < 0$. The rate is expressed by vector b which must satisfy $\sum_{i=1}^n b^i = 1$ in order to promise the resolution of the problem. Our target is to make sure of the flow vector $x = |x^e|_{e \in \mathcal{E}}$, where x^e is the amount of traffic in edge $e=(i,j)$. The flow conservation implies $Ax = b$, where A is the node-edge incidence matrix in $n \times E$ order and its definition is:

$$|A|_{ij} = \begin{cases} 1 & j < i \\ -1 & j > i \\ 0 & \text{others} \end{cases}$$

We define the incentive as the negative value of scalar cost function $\psi_e(x^e)$, which denotes the cost of x^e units of traffic through edge e . We suppose that the cost function ψ_e is strict convex and second-order consecutive derivable. In further, the network optimization problem of the maximum incentive can be defined as the following equation

$$\begin{aligned} & \text{maximize} && \sum_{e=1}^E -\psi_e(x^e) \\ & \text{subject to:} && Ax = b \\ & && c_l \leq x \leq c_u \end{aligned} \quad (1)$$

where c_l and c_u denote the constraints of link capacity. For convenience, all of the constraints are supposed to be the same.

Since the constraints of link capacity $c_l \leq x \leq c_u$ may cause ill condition of dual function, we introduce the log

bound $(1/t)\log(c_u - x^e)$ and $(1/t)\log(x^e - c_l)$, where t is the designated parameter so as to use the essential point algorithm. Consequently, the equation can be rewritten as:

$$\begin{aligned} & \text{maximize} && -f(x) = \sum_{e=1}^E -\phi_e(x^e) \\ & \text{subject to:} && Ax = b \\ & && c_l \leq x \leq c_u \end{aligned} \quad (2)$$

where,

$$\phi_e(x^e) = \psi_e(x^e) - (1/t) \left[\log(c_u - x^e) + \log(x^e - c_l) \right] \quad (3)$$

The relatively large t makes the equation (2) closer to the equation (1), while too small t will lead to the ill condition of dual function.

If the external global rate of vector b is given, then the equation (2) can be solved with ADD-N algorithm and it has super linear convergence rate. We will understand the dual gradient descent algorithm again and define the equation's Lagrange function as:

$$\mathcal{L}(x, \lambda) = -\sum_{e=1}^E \phi_e(x^e) + \lambda'(Ax - b) \quad (4)$$

And the dual function $q(\lambda)$ will be defined as:

$$\begin{aligned} q(\lambda) &= \sup_{x \in \mathbb{R}^E} \mathcal{L}(x, \lambda) = \sup_{x \in \mathbb{R}^E} (-f(x) + \lambda'Ax) - \lambda'b \\ &= \sum_{e=1}^E \sup_{x^e \in \mathbb{R}} (-\phi_e(x^e) + (\lambda'A)^e x^e) - \lambda'b \end{aligned} \quad (5)$$

In the last equation, $\lambda'Ax = \sum_{e=1}^E (\lambda'A)^e x^e$ and it exchanges the order of summation and the upper limit.

We can see from the equation, the estimation of dual function $q(\lambda)$ can be decomposed into E one-dimensional problem in summation. It is supposed that such a problem always should have an optimal solution and this solution should be the unique according to constrict convex of function ϕ_e . We define the only optimal solution as $x^e(\lambda)$ and apply the first-order optimum condition to these problems and get:

$$x^e(\lambda) = (\phi'_e)^{-1}(\lambda^i - \lambda^j) \quad (6)$$

where $i, j \in \mathcal{N}$ denote the source node and target node of edge $e=(i,j)$, respectively. It can be seen that the estimation of the optimal solution of each node e is based on the local information of edge cost function ϕ^e and the dual variants of incidence node i and j .

The minimization of the shown dual function $q(\lambda)$ can be solved with gradient descent algorithm. We consider an iteration step k and a random initial vector λ_0 and define that each iteration can get the result by the following equation

$$\lambda_{k+1} = \lambda_k - \alpha_k g_k \quad (7)$$

where $g_k = g(\lambda_k) = \nabla q(\lambda_k)$ denotes the gradient of dual function $q(\lambda)$ in the location $\lambda = \lambda_k$. Firstly, there is an important fact that the gradient can be denoted as $g_k = Ax(\lambda_k) - b$, where the element $x^e(\lambda_k)$ of vector $x(\lambda_k)$ can be fixed by the equation $\lambda = \lambda_k$. Then the other important fact is the i element g_k^i of gradient g_k can be evaluated by the equation on account of the sparse characteristic of node-edge incidence matrix.

$$g_k^i = \sum_{e=(i,j)} x^e(\lambda_k) - \sum_{e=(j,i)} x^e(\lambda_k) - b_i \quad (8)$$

The algorithm itself reflects its characteristic of distributed implementation. Each node i keeps the information of its dual iteration λ_k^i and the original iteration $x^e(\lambda_k)$ through edge $e=(i,j)$. According to the i equation, the i gradient component g_k^i can be fixed by using the local original iteration $x^e(\lambda_k)$ of edge $e=(i,j)$, the initial iteration $x^e(\lambda_k)$ of the adjacent edge $e=(j,i)$ and the external traffic rate value b_i of node i . Then the dual variants can be updated by the i equation. After updating dual iteration, it updates the initial variant according to the i equation, where the update need to adopt the local Lagrange multiplier λ_k^i and λ_k^j .

In many real demand cases, the external global traffic rate of vector b is unknown. What we can get is often that the unbiased estimation $\hat{b}_k$ of vector b can satisfy $\mathbb{E}[\hat{b}_k] = b$. Let us take an example that the flow rate of vector b in a communication network denotes the average packet arrival rate, which is usually unknown; nevertheless, the unbiased estimation $\hat{b}_k$ denotes the number of packet entering and leaving the network in a certain time point. According to the definition of unbiased estimation, $\mathbb{E}[\hat{b}_k] = b$ is equivalent. Using the unbiased estimation to replace the actual traffic b_i in the network, the random gradient component can be obtained.

$$\hat{g}_k^i = \sum_{e=(i,j)} x^e(\lambda_k) - \sum_{e=(j,i)} x^e(\lambda_k) - \hat{b}_k^i \quad (9)$$

These gradient components consist of random gradient vector $\hat{g}_k = Ax(\lambda_k) - \hat{b}_k$. Due to $\mathbb{E}[\hat{g}_k] = g_k$, and as we all know the gradient g_k shows the steepest descent direction, the random sub-gradient $\hat{g}_k$ points the average speediest descent direction. Using $\hat{g}_k$ to replace g_k in equation can get the random sub-gradient

descendant algorithm and its convergence characteristic is determined by the uncertain characteristic of b .

Demand fluctuation and truncation error may lead to two possible statistical property of $\hat{b}_k$. In the case of demand fluctuation, vector $\hat{b}_k$ changes incessantly over the time. We reasonably suppose that the square error of $\hat{b}_k$ is bounded, which is equivalent to indicating the second moment of random gradient $\hat{g}_k = Ax(\lambda_k) - \hat{b}_k$ has an upper bound, as shown in the following formula.

$$\mathbb{E}[\|Ax(\lambda_k) - \hat{b}_k\|^2] \leq \sigma^2 \quad (10)$$

In truncation error model, we suppose the uncertain characteristic derives from the estimation of constraint failure. According to the given true constraint failure g_k , the random gradient can be written as $\hat{g}_k = g_k + \delta_k$, where δ_k is a certain truncation error. Similar to the situation of floating-point operation, the norm of truncation error is in proportion to that of g_k , namely, $\|\delta_k\| \leq \gamma \|g_k\|$. Therefore, Let's suppose

$$\|\hat{g}_k\| \leq (1 + \gamma) \|g_k\| \quad (11)$$

It will constrain the uncertain characteristic of random gradient approximation $\hat{g}_k$. In the case of fluctuation demand, the analysis is described in 4.1, while the analysis of truncation error model is in 4.2. Before we describe these two situations, we will introduce random ADD series algorithm in Section 3.

III. RANDOM ADD ALGORITHM

Consider iteration step k and random initial vector λ_0 and define the recursive equation of each iteration step's result which is shown as follows:

$$\lambda_{k+1} = \lambda_k + \alpha_k d_k \quad (12)$$

where d_k denotes the descent direction and when it satisfies $g_k' d_k \leq 0$, α_k is the given step order. If the equation is defined as Newton step, it is Newton method. Newton step can be expressed as $d_k = -H_k^\dagger g_k$, where $H_k^\dagger$ denotes the pseudo-inverse of the dual function's Hessian matrix $H_k = \nabla^2 q(\lambda_k)$.

In order to get the expression of the dual function's Hessian matrix, we consider the situation of the given dual λ_k and the initial vector $x_k = x(\lambda_k)$, and then the second-order approximation which takes the present initial iteration value x_k as center.

$$\begin{aligned} \hat{f}(y) &= f(x_k) + \nabla f(x_k)'(y - x_k) \\ &\quad + \frac{1}{2}(y - x_k)' \nabla^2 f(x_k)(y - x_k) \end{aligned} \quad (13)$$

As shown in the function, the initial optimization problem now can be replaced by the maximization problem shown in the approximation equation $-\hat{f}(y)$ and it has the constraint $Ay = b$. The approximation problem is a second-planned problem and its duality is also the second time, which is shown in the following:

$$\min_{\lambda \in \mathbb{R}^n} q(\lambda_k) = \min_{\lambda \in \mathbb{R}^n} \frac{1}{2} \lambda_k' A \nabla^2 f(x_k)^{-1} A' \lambda_k + p' \lambda_k + r \quad (14)$$

Vector p and constant r can be expressed with the closed form about function $\nabla f(x_k)$ and $\nabla^2 f(x_k)$, but they are not the key points we will describe here. According to the equation we can conclude an important conclusion, namely, the conclusion of dual Hessian matrix, as shown in the following equation.

$$H_k = Q = A(-\nabla^2 f(x_k)^{-1})A' \quad (15)$$

According to the definition of $f(x)$ in the equation, the initial Hessian matrix $\nabla^2 f(x_k)$ is a diagonal matrix and also is a negative definite matrix about constrict convex function $f(x)$. After further observation, we can see, the dual Hessian matrix is calculated by transposing the product of incidence matrix, a positive diagonal matrix and incidence matrix and is also a weighted version of the network map's laplacian form. One of the reasons why H_k owns laplacian form is that the matrix is determined by its diagonal element. The other reason is that 1 is its eigenvalue (the other eigenvalue is 0). In addition, as long as the initial Hessian matrix $\nabla^2 f(x_k)$ is full rank, H_k is reversible in the subspace $\mathbf{1}^\perp$. Because gradient g_k is in subsidy space $\mathbf{1}^\perp$, we can find that Newton step is $d_k = -H_k^\dagger g_k$ that is just the expression we point out in the previous. However, it needs the global information to calculate the pseudo inverse $H_k^\dagger$. Therefore, we should further find the Newton step approximation method which only needs local information.

A. Acceleration Dual Descent Algorithm

ADD-N algorithm adopts matrix division to produce the approximation of Newton step and also need other nodes' information within N hops. We consider developing a proper Taylor in finite length to express Newton step. In iteration step k, Hessian matrix is divided into the diagonal element part D_k and the off-diagonal part B_k and it is expressed as $H_k = D_k - B_k$. In further, the Hessian matrix is rewritten as $H_k = D_k^{-\frac{1}{2}} \left(I - D_k^{-\frac{1}{2}} B_k D_k^{-\frac{1}{2}} \right) D_k^{-\frac{1}{2}}$, which denotes its pseudo inverse $H_k^\dagger = D_k^{-\frac{1}{2}} \left(I - D_k^{-\frac{1}{2}} B_k D_k^{-\frac{1}{2}} \right)^{-\dagger} D_k^{-\frac{1}{2}}$. Now

we will develop a Taylor in the middle part of the expression and adopt the equation $(I - X)^\dagger v = \left(\sum_{i=0}^{\infty} X^i \right) v$ and the equation can be effective for random vector v that is orthogonal with matrix X whose eigenvalue is 1. Because g_k and $\mathbf{1}$ are orthogonal, it can get:

$$d_k = -H_k^\dagger g_k = -\sum_{i=0}^{\infty} D_k^{-\frac{1}{2}} \left(I - D_k^{-\frac{1}{2}} B_k D_k^{-\frac{1}{2}} \right)^i D_k^{-\frac{1}{2}} g_k \quad (16)$$

Now Newton step is expressed as the sum of infinite series, therefore we can make truncation on the sum and define a series of approximation equations, which is shown as follows:

$$\begin{aligned} d_k^{(N)} &= -\sum_{i=0}^N D_k^{-\frac{1}{2}} \left(I - D_k^{-\frac{1}{2}} B_k D_k^{-\frac{1}{2}} \right)^i D_k^{-\frac{1}{2}} g_k \\ &:= -\bar{H}_k^{(N)} g_k \end{aligned} \quad (17)$$

where, the approximate pseudo inverse of Hessian matrix can be defined as the following equation:

$$\bar{H}_k^{(N)} := \sum_{r=0}^N D_k^{-\frac{1}{2}} \left(I - D_k^{-\frac{1}{2}} B_k D_k^{-\frac{1}{2}} \right)^r D_k^{-\frac{1}{2}} \quad (18)$$

In order to replace the Newton step in the equation with $d_k^{(N)} = -\bar{H}_k^{(N)} g_k$, we can get the acceleration dual descent algorithm. The N-order approximation of pseudo inverse $H^\dagger$ increase one item based on the N-1 order of $\bar{H}_k^{(N)}$, and the increased item's expression is $(D_k^{-1} B_k)^N D_k^{-1}$. Its sparse characteristic is reflected in the sparse characteristic of B_k , which is similar to that of N hop neighbor and at the same time, the element of local Newton step needs the information of N hop node. As a result, we explain the equation as a series of approximation marked with N, which is the Hessian approximation that needs the N hop neighbor's information in the network.

B. Random Acceleration Dual Descent Algorithm

After acquiring the unbiased estimation $\hat{b}$ of the external global flow rate vector b , we make analysis on its uncertain characteristic. If vector b is unknown, the approximate Newton step $d_k^{(N)}$ in the equation cannot be calculated. Therefore, we use the random approximate $\hat{d}_k^{(N)}$ of Newton step to replace $d_k^{(N)}$ and get the recursive equation shown in the following.

$$\lambda_{k+1} = \lambda_k - \alpha_k \hat{d}_k^{(N)}, \quad \forall k \geq 0 \quad (19)$$

Here is an important fact that Hessian matrix H_k doesn't depend on b and its approximation $H_k^{(N)}$ also doesn't depend on b , therefore we introduce random approximation Newton step as follows.

$$\hat{d}_k^{(N)} = H_k^{(N)} \hat{g}_k = H_k^{(N)} (Ax_k - \hat{b}_k) \quad (20)$$

We put it into the equation and then get random ADD-N algorithm which this paper will put forward and analyze.

Because of the expectation calculation's linear characteristic, we can get $\mathbb{E}[\hat{d}_k^{(N)}] = d_k^{(N)}$. As a result, the random Newton step approximation is the unbiased estimation of the random Newton step approximation $d_k^{(N)}$. From this we can get an important conclusion: $\hat{d}_k^{(N)}$ is the average speediest descent direction, for it satisfies the equation.

$$\mathbb{E}[g_k' \hat{d}_k^{(N)}] = \mathbb{E}[g_k' \bar{H}_k^{(N)} \hat{g}_k] = g_k' \bar{H}_k^{(N)} g_k < 0 \quad (21)$$

The reason why $\hat{d}_k^{(N)}$ satisfies the equation is that the approximate Hessian matrix $H_k^{(N)}$ is negative definite in the subspace $\mathbf{1}^\perp$. It makes us reasonably believe that Random ADD-N algorithm converges to the optimal Lagrange multiplier λ^* . A formal proof of this conclusion is in chapter 4. In further, we know ADD-N algorithm has the square convergence rate, therefore in theory, it has speedier convergence rate than random subsidiary gradient descendant algorithm. The calculation stimulation result in chapter 5 proves that the conclusion is right.

Before we make analysis on the convergence of random ADD algorithm, we offer some necessary assumptions and definitions. At first, we introduce Newton error, as is shown in equation.

$$\varepsilon_k = H_k \hat{d}_k^{(N)} + g_k \quad (22)$$

If $\hat{d}_k^{(N)} = d_k$, that is, it equals to real Newton step, then it can get $\varepsilon_k = 0$. In random ADD algorithm, Newton error ε_k is the variant which measures the difference between random approximate Newton step and the real Newton step. Notice that, due to the truncation of the equation's sum and the random characteristic of $\hat{b}_k$, we can get $\varepsilon_k \neq 0$.

Then we will further define constant $\bar{\rho}$, as is shown in the formula.

$$\rho(D_k^{-1} B_k) \leq \bar{\rho} \quad (23)$$

where $\rho(D_k^{-1} B_k)$ is spectral radius of matrix $D_k^{-1} B_k$ in subspace $\mathbf{1}^\perp$. This constant satisfies $\bar{\rho} \in (0,1)$ and is related to laplacian spectrum and it can indicate the spreading situation of information in the designated network.

The approximate Hessian inverse matrix still keeps well-conditioned in subspace $\mathbf{1}^\perp$, namely, for the mark N of the given ADD series algorithm and dual invariant λ ,

and all vectors $v \in \mathbf{1}^\perp$, there is equation which is effective.

$$\|\bar{H}^{(N)}(\lambda)v\| \leq M \|v\| \quad (23)$$

where M is the supermom of assumption $\|H(\lambda)\|$.

Assumption1 requires that Hessian matrix $H(\lambda)$ keeps well-conditioned in subspace $\mathbf{1}^\perp$, which is the normal requirement of descent algorithm. Assumption 2 requires the approximate Hessian inverse matrix $\bar{H}^{(N)}(\lambda)$ keeps well-conditioned in subspace $\mathbf{1}^\perp$, which is a reasonable assumption, because $\bar{H}^{(N)}(\lambda)$ is a truncation sum, and also the extreme value of this truncation sum is $H^*(\lambda)$ when N tends to infinity.

IV. THE ANALYSIS AND CONCLUSION OF CONVERGENCE CHARACTERISTIC

Due to the random characteristic of the problem, the analysis adopts the Martingale Theories, especially the super-martingale theory. In order to keep the integrity of the statement, we present a version of the theory related to the problem.

Consider a nonnegative random process $B(\mathbb{N})$ and it supports $\beta(k) \geq 0$ and a series of embedded σ algebra $\mathcal{B}(k)$ and includes $\beta(t)$ at least, $t \in \{1, \dots, k\}$. So there is the following inequality,

$$\mathbb{E}[\beta(k+1) | \mathcal{B}(k)] \leq \beta(k) \quad (24)$$

The super-martingale convergence theorem transforms our recursive expectation into a certain promise, namely, almost all of sequences are convergent or the random process $\beta(k)$ converges to an extreme random variant. In the following theorem, we illustrate that random variant sequence converges to 0.

A. The Situation of Demand Fluctuation

This chapter will describe the convergence performance of ADD-N algorithm when the uncertain characteristic of $\hat{b}$ performs in demand fluctuation. At this time, the second-order moment of the demand fluctuation has the bound shown in the equation. Firstly we will offer the result to prove the square distance sequence $\|\lambda_k - \lambda^*\|^2$ can satisfy the relationship similar to that of super martingale.

Lemma1. We will give the iteration λ_k defined by the equation, where step length is fixed, $\alpha_k = \alpha \in (0, 1/(2-2\bar{\rho}^{N+1}))$ and the dual optimal Lagrange multiplier λ^* and the formula is as follows:

$$\begin{aligned} & \mathbb{E}[\|\lambda_{k+1} - \lambda^*\|^2 | \mathcal{L}_k] \\ & \leq (1 - 2\alpha(1 - \bar{\rho}^{N+1})) \|\lambda_k - \lambda^*\|^2 + \alpha^2 M^2 \sigma^2 \end{aligned} \quad (25)$$

where σ algebra $\mathcal{L}_k$ includes $\|\lambda_k - \lambda^*\|^2$, $t \leq k$.

When $\|\lambda_k - \lambda^*\|^2 \leq \alpha^2 M^2 \sigma^2 / [2(1 - \bar{\rho}^{N+1})]$, according to the lemma, we know the sequence $\|\lambda_k - \lambda^*\|^2$ can satisfy the super martingale feature. To build a proper super martingale, we define the mark:

$$k^* = \min_k \left\{ k : \|\lambda_k - \lambda^*\|^2 \leq \alpha^2 M^2 \sigma^2 / [2(1 - \bar{\rho}^{N+1})] \right\} \quad (26)$$

Record the iteration step of $\|\lambda_k - \lambda^*\|^2$ which is lower than $\alpha^2 M^2 \sigma^2 / [2(1 - \bar{\rho}^{N+1})]$ at the first time, and then define the following sequence:

$$G_k = \|\lambda_k - \lambda^*\|^2 - \frac{\alpha^2 M^2 \sigma^2}{2(1 - \bar{\rho}^{N+1})}, k < k^* \quad (27)$$

When it satisfies $k^* \leq k$, we have $G_k = 0$.

Lemma 2. We will give the iteration λ_k defined by the equation, where step length is fixed, $\alpha_k = \alpha \in (0, 1/(2 - 2\bar{\rho}^{N+1}))$ and the dual optimal Lagrange multiplier λ^* , and the sequence G_k defined by the equation almost makes sure of converging to 0.

Lemma 2 shows the sequence G_k is a super martingale. The fact is used to make conclusion on λ_k in the following theorem.

Lemma 3. We will give the iteration λ_k defined by the equation, where step length is fixed, $\alpha_k = \alpha \in (0, 1/(2 - 2\bar{\rho}^{N+1}))$ and the dual optimal Lagrange multiplier λ^* . When $k \rightarrow \infty$, it has the equation:

$$\Pr \left(\liminf \|\lambda_k - \lambda^*\|^2 \leq \frac{\alpha^2 M^2 \sigma^2}{2(1 - \bar{\rho}^{N+1})} \right) = 1 \quad (28)$$

where the coefficient M of constrict convex is defined by assumption 1, σ denotes the uncertain characteristic defined by the equation, $\bar{\rho}$ is the connecting coefficient, N is the parameter in ADD-N algorithm.

Lemma 3 promises that random ADD-N algorithm almost makes sure that λ_k can often enter the error neighbor domain of the optimal multiplier λ^* . Before entering the iteration stepson of this neighbor domain, it may bring about the iteration step far away from the optimal multiplier; however, the magnitude of the iteration value has high possibility to get the minimum value. The error radius increases with σ and $\sqrt{\alpha}$. It will select a small step length in order to decrease the error radius but this will decrease the convergence rate.

B. The Situation of Truncation Error

This chapter characterizes the convergence performance of ADD-N algorithm when the uncertain

characteristic of the gradient approximate satisfies the equation. If the error derives from the value truncation due to constraint failure, such situation may happen. The following lemma shows that the Newton error will vanish when it comestible to the optimal point.

We will give the iteration λ_k defined by equation, and the fixed step length $\alpha_k = \alpha \in (0, 1]$. On the premise of the assumption as is described in the equation, there is:

$$\mathbb{E} \|\varepsilon_k\|^2 \leq \left(1 + \alpha^2 (1 + \gamma)^2 (1 + \bar{\rho}^{N+1})^2 - 2\alpha (1 - \bar{\rho}^{N+1}) \right) \|g_k\|^2 \quad (29)$$

where g_k is the Newton error defined in the equation, γ denotes the ration uncertain characteristic defined by equation, $\bar{\rho}$ is the connecting coefficient defined by equation, and N is the parameter in ADD-N algorithm.

Lemma 3 shows the approximate Newton direction proposed in the paper tends to the real Newton direction which has come to the optimal value. Because the performance of ADD-N algorithm near the optimal point is similar to that of the real Newton method, we can think this algorithm can converge to the optimal point and be also the square convergence.

We will give the iteration λ_k defined by equation, and the fixed step length satisfies the fixed equation:

$$\alpha_k = \alpha \in \left(0, \frac{2(1 - \bar{\rho}^{N+1})}{(1 + \bar{\rho}^{N+1})^2 (1 + \gamma)^2} \right) \quad (30)$$

where, according to the ratio error bound defined by equation, $\gamma > 0$. Then when $k \rightarrow \infty$, there is:

$$\Pr(\lambda_k = \lambda^*) \rightarrow 1 \quad (31)$$

where λ^* is dual optimal multiplier, $\bar{\rho}$ is the connecting coefficient defined by equation, N is the parameter in ADD-N algorithm.

We can know from theorem 4 that ADD-N algorithm can make sure that λ_k almost converges to λ^* . The larger is N , the higher is the network connecting possibility ($\bar{\rho}$ is smaller); the smaller is γ , the more accurate is the approximate Newton direction, which makes the relatively large α possible. The relatively large α and the increased accuracy both improve the expected convergence rate.

V. THE NUMERICAL EXPERIMENT RESULT

The numerical experiment compares the performances among random ADD-N algorithm, random gradient descent algorithm and random Newton algorithm. The experiment mainly focuses on ADD-1 algorithm, while other N value experiment results can be obtained. Considering the network traffic problem of single commodity, the network graph is composed of 50 nodes and 100 edges and these nodes and edges will be uniform

random. The function $f_e(x_e) = \exp\{x_e\} - \exp\{-x_e\}$, $e \in \mathcal{E}$ is selected as the cost function in the simulation due to its concave feature. Because the cost function increases fast with the increase of flow x_e , the network traffic value x_e in the simulation is relatively small. The simulation results of the situation in variant demand fluctuation and truncation error are shown in figure 1 and 2.

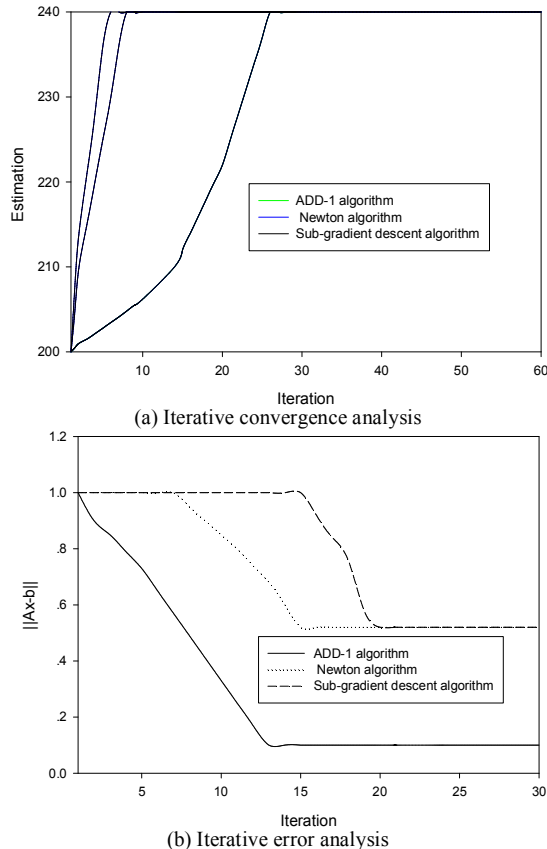


Figure 1. The performance of random ADD-1 algorithm in the case of variant demand fluctuation

We can see from Figure 2 that ADD-1 algorithm can obtain smaller error neighborhood than sub-gradient descent algorithm and its iteration step reduces an order of magnitude in the neighborhood, and at the same time, ADD-1 algorithm can achieve the error neighborhood similar to that of Newton algorithm. As to the convergence accuracy, all of three algorithms can achieve 10^{-8} precision at least. Therefore, we can verify the convergence of ADD-1 algorithm and the convergence of the relative sub-gradient descendant algorithm.

We can see from figure 2 that the error neighborhood of ADD-1 algorithm is almost similar to that of Newton algorithm in the case of truncation error, and meanwhile the convergence rate is improved. After the first 120 iteration, ADD-1 algorithm can achieve the machine precision, which is nearly equal to that of Newton algorithm after 75 iteration; however, in order to achieve the same precision, the sub-gradient descent algorithm need far more than 2000 iterations.

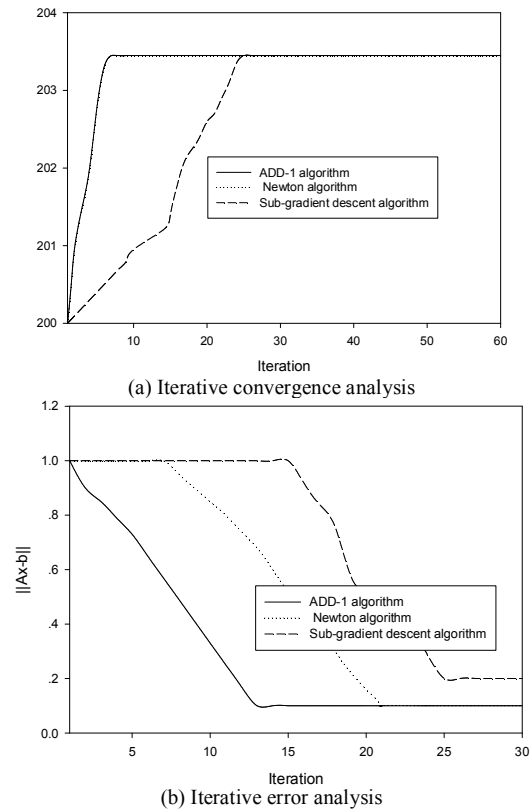


Figure 2. The performance of random ADD-1 algorithm in the case of truncation error

VI. CONCLUSION

This paper shows that as long as we enforce some constraints on the uncertain characteristic, the acceleration dual descendant algorithm can be developed into random form. When the second-moment method of the random error has upper bound, it can guarantee the algorithm converging to the error neighbourhood of the optimal value with probability 1. When the random error and constraint failure are in proportion, the algorithm can converge to the optimal solution with the possibility 1. The super-martingale convergence theorem implies the linear average convergence rate, which can be used to prove the related theorem in the paper. However, the numerical experiment result shows that, in real network environment, the convergence rate of random ADD-1 algorithm is superior to that of random gradient descent algorithm. The next research of this paper is that we will apply the relevant conclusion to the business cellular network, for example, CDMA network and 3G cellular network, so as to get better cellular network performance.

REFERENCES

- [1] Zhong S, Xia K, Yin X, et al. The representation and simulation for reasoning about action based on Colored Petri Net//*Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on. IEEE*, 2010 pp. 480-483.
- [2] Le, D., Jin, Y., Xia, K., & Bai, G. (2010, March). Adaptive error control mechanism based on link layer frame importance valuation for wireless multimedia sensor networks. In *Advanced Computer Control (ICACC), 2010*

- 2nd International Conference on (Vol. 1, pp. 465-470). IEEE.
- [3] Xia K, Cai J, Wu Y. Research on Improved Network Data Fault-Tolerant Transmission Optimization Algorithm. *Journal of Convergence Information Technology*, 2012, 7(19).
 - [4] XIA, K., WU, Y., REN, X., & JIN, Y. (2013). Research in Clustering Algorithm for Diseases Analysis. *Journal of Networks*, 8(7), pp. 1632-1639.
 - [5] Yao, Yufeng, Jinyi Chang, and Kaijian Xia. "A case of parallel eeg data processing upon a beowulf cluster." *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on. IEEE*, 2009.
 - [6] Kai-jian, Xia, et al. "An edge detection improved algorithm based on morphology and wavelet transform." *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on. Vol. 1. IEEE*, 2010.
 - [7] Zou, Junni, et al. "Lifetime and distortion optimization with joint source/channel rate adaptation and network coding-based error control in wireless video sensor networks." *Vehicular Technology, IEEE Transactions on* 60.3 (2011) pp. 1182-1194.
 - [8] Suo, Hui, et al. "Issues and challenges of wireless sensor networks localization in emerging applications." *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on. Vol. 3. IEEE*, 2012.
 - [9] Gou, Haosong, Younghwan Yoo, and Hongqing Zeng. "A partition-based LEACH algorithm for wireless sensor networks." *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on. Vol. 2. IEEE*, 2009.
 - [10] Gou, Haosong, Younghwan Yoo, and Hongqing Zeng. "A partition-based LEACH algorithm for wireless sensor networks." *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on. Vol. 2. IEEE*, 2009.
 - [11] Xia K, Cai J, Wu Y. Research on Improved Network Data Fault-Tolerant Transmission Optimization Algorithm. *Journal of Convergence Information Technology*, 2012, 7(19).
 - [12] Abdulsalam, Hanady M., and Layla K. Kamel. "W-LEACH: Weighted Low Energy Adaptive Clustering Hierarchy aggregation algorithm for data streams in wireless sensor networks." *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE*, 2010.
 - [13] Katiyar, Vivek, et al. "Improvement in LEACH protocol for large-scale wireless sensor networks." *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on. IEEE*, 2011.
 - [14] Kai-jian X, Yu-feng Y, Jin-yi C, et al. An edge detection improved algorithm based on morphology and wavelet transform//*Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on. IEEE*, 2010, 1 pp. 404-407.