2018 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2018, Tarragona, Spain, September 2-6, 2018

A Distributed Algorithm for Solving Sandberg-Willson Equations

Masaaki Takeuchi and Norikazu Takahashi

Graduate School of Natural Science and Technology, Okayama University
3–1–1 Tsushima-naka, Kita-ku, Okayama 700–8530, Japan
Email: takeuchi@momo.cs.okayama-u.ac.jp, takahashi@cs.okayama-u.ac.jp

Abstract—A distributed algorithm for solving Sandberg-Willson equations, which are well-known nonlinear equations in the field of circuit theory, is proposed in this paper. It is shown numerically that, by using this algorithm, all agents in a network can find the unique solution. It is also shown theoretically that the sequence of solutions of every agent converges to the unique solution under some assumptions on the nonlinear function and the coefficient matrix in the equation.

1. Introduction

Distributed computations by multi-agent networks have recently attracted a great deal of attention [1–5]. One of the most well-known distributed computations is the average consensus [1]; the state value of each agent is updated based on the state values of agents in its neighborhood, and converges to the average of the initial values of all agents. Other examples are the computation of the algebraic connectivity of the network [4, 5], the constrained consensus [2, 3], and the constrained optimization [2].

In this paper, we consider the distributed computation of a class of nonlinear algebraic equations studied by Sandberg and Willson Jr. [6]. An important property of these equations is that the existence and the uniqueness of the solution is guaranteed. However, the problems are neither linear nor convex, conventional approach [2, 3] based on projection cannot be directly applied. We propose a new distributed algorithm for solving the above-mentioned equations. This algorithm is based on a consensus algorithm and numerical techniques such as the Newton method [7] for solving nonlinear equations. It is shown numerically that, by using this algorithm, all agents in a network can find the unique solution. It is also shown theoretically that the sequence of solutions of every agent converges to the unique solution under some assumptions on the nonlinear function and the coefficient matrix in the equation.

2. Problem Statement

We consider nonlinear algebraic equations for $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \in \mathbb{R}^n$ of the following form:

$$\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ is a given constant matrix, $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a given constant vector, and $\boldsymbol{f}(\boldsymbol{x}) = (f_1(x_1), f_2(x_2), \ldots, f_n(x_n))^{\mathrm{T}}$ is a given nonlinear function from $\mathbb{R}^n$ to $\mathbb{R}^n$. Throughout this paper, we assume that (1) satisfies the following two assumptions.

Assumption 1 $\boldsymbol{A}$ is a P_0 matrix.

Assumption 2 For all $i \in \{1, 2, \ldots, n\}$, the function $f_i : \mathbb{R} \to \mathbb{R}$ is continuous, strictly monotone increasing, and surjective.

Sandberg and Willson, Jr. [6] proved that (1) has a unique solution if Assumptions 1 and 2 are valid. Hence we hereafter call equations of the form (1) that satisfy Assumptions 1 and 2 Sandberg-Willson equations.

We want to let a network of n agents solve Sandberg-Willson equations under the following situation: i) agent i only knows the i-th equation, that is,

$$f_i(x_i) + \boldsymbol{a}_i\boldsymbol{x} = b_i \tag{2}$$

where $\boldsymbol{a}_i$ is the i-th row of the matrix $\boldsymbol{A}$, ii) each agent can find a solution of its own equation, iii) each agent can send its solution to some other agents and receive their solutions through communication channels (see Fig. 1). The objective of this paper is to design a distributed algorithm for the multi-agent network so that every agent can find the unique solution of (1) under the above-mentioned situation.

The communication among agents can be represented by a simple and undirected graph $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ is the vertex set and E is the edge set. Vertex i corresponds to agent i. Each edge is an unordered pair of distinct vertices. Edge $\{i, j\}$ is contained in E if and only if agents i and j can directly communicate with each other. The neighborhood $\mathcal{N}_i$ of vertex i is defined as $\mathcal{N}_i = \{j \mid \{i, j\} \in E\} \cup \{i\}$.

3. Proposed Algorithm

Let us define n functions $g_1, g_2, \ldots, g_n$ by

$$g_i(x_i) \triangleq f_i(x_i) + a_{ii}x_i, \quad i = 1, 2, \ldots, n. \tag{3}$$

Note that a_{ii} is nonnegative for all i because $\boldsymbol{A}$ is P_0 matrix. So $g_i(x_i)$ is continuously differentiable, strictly monotone increasing, surjective, and therefore has the inverse function $g_i^{-1} : \mathbb{R} \to \mathbb{R}$ which is also continuously differentiable, strictly monotone increasing and surjective. Let

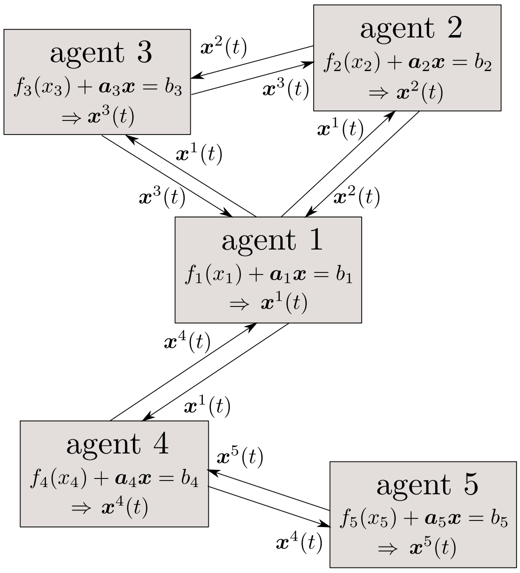

Figure 1: A multi-agent network for solving (1).

$\boldsymbol{x}^i(t) = (x_1^i(t), x_2^i(t), \ldots, x_n^i(t))^{\mathrm{T}}$ be the solution estimated by agent i at discrete time $t \in \mathbb{Z}_+$ where $\mathbb{Z}_+$ is the set of nonnegative integers. Then the distributed algorithm we propose in this paper is described as follows.

In the first step, each agent i randomly chooses $\boldsymbol{x}^i(0)$, and then obtain $\boldsymbol{x}^i(1)$ by

$$x_j^i(1) = \begin{cases} g_i^{-1}\left(b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij} x_j^i(0)\right), & \text{if } j = i, \\ x_j^i(0), & \text{if } j \neq i. \end{cases} \tag{4}$$

Here we have assumed that each agent i can compute the value of $g_i^{-1}(y)$ for any $y \in \mathbb{R}$. Taking (3) into account, we see that $\boldsymbol{x}^i(1)$ is a solution of (2). In the t-th step ($t \geq 2$), each agent i computes the average of the solutions of agents $k \in \mathcal{N}_i$ by

$$\boldsymbol{w}^i(t-1) = \frac{1}{|\mathcal{N}_i|} \sum_{k \in \mathcal{N}_i} \boldsymbol{x}^k(t-1) \tag{5}$$

and then obtain $\boldsymbol{x}^i(t)$ by

$$x_j^i(t) = \begin{cases} g_i^{-1}\left(b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij} w_j^i(t-1)\right), & \text{if } j = i, \\ w_j^i(t-1), & \text{if } j \neq i. \end{cases} \tag{6}$$

We see that $\boldsymbol{x}^i(t)$ is also a solution of (2).

It is impossible in general to express g_i^{-1} explicitly. So each agent i has to solve the equation $g_i(x_i) = b_i - \sum_{j=1, j\neq i}^{n} a_{ij} w_j^i(t-1)$ for x_i numerically. There are many good algorithms for solving such equations. For example, the inexact Newton method for solving systems of monotone equations proposed by Solodov and Svaiter [7] has an important property that the sequence generated by the method always globally convergent to the unique solution. We therefore use this method in the numerical experiments.

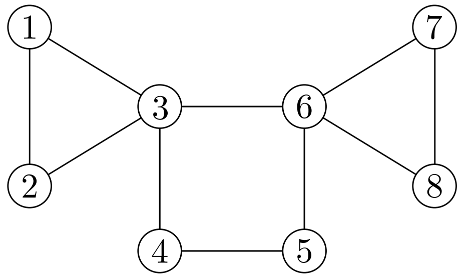

Figure 2: Communication graph G_1.

4. Numerical Experiments

In order to check the validity of the proposed algorithm, we conducted some numerical experiments. In the first experiment, we set

$$\boldsymbol{A} = \begin{pmatrix} 0.810 & 0.123 & -0.139 & -0.599 & -0.005 & 0.426 & -0.073 & 0.463 \\ 0.418 & 2.185 & -0.304 & -0.066 & 0.202 & 0.300 & 0.363 & -0.285 \\ -0.081 & -0.392 & 0.902 & -0.414 & 0.261 & 0.047 & 0.121 & 0.164 \\ -0.133 & 0.147 & -0.057 & 1.038 & -0.182 & 0.193 & -0.028 & -0.328 \\ -0.064 & -0.236 & 0.031 & -0.154 & 0.450 & -0.173 & 0.133 & 0.105 \\ 0.392 & 0.186 & 0.086 & -0.587 & 0.138 & 1.131 & -0.100 & 0.573 \\ -0.075 & 0.176 & -0.073 & 0.204 & 0.037 & -0.014 & 0.802 & -0.174 \\ 0.158 & 0.167 & 0.115 & -0.337 & 0.261 & 0.096 & 0.068 & 0.976 \end{pmatrix}$$

$$\boldsymbol{b} = (1.902, 30.594, 31.889, 42.617, 22.120, 46.400, 609.255, 5.754)^{\mathrm{T}},$$

and

$$f_i(x_i) = i x_i + \mathrm{e}^{0.1 i x_i}, \quad i = 1, 2, \ldots, 8.$$

We also set the communication graph G of the multi-agent network to G_1 shown in Fig. 2.

We obtained $\boldsymbol{A}$ above in the following way. We first generated a matrix $\boldsymbol{B} = (b_{ij}) \in \mathbb{R}^{n\times n}$ such that $b_{jj} > \sum_{i=1, i\neq j}^{n} |b_{ij}|$ for $j = 1, 2, \ldots, n$. This is easily done by choosing the values of off-diagonal entries randomly and then setting the value of diagonal entries so that the above inequalities hold. We next generated a matrix $\boldsymbol{C} = (c_{ij}) \in \mathbb{R}^{n\times n}$ such that $c_{jj} \geq \sum_{i=1, i\neq j}^{n} |c_{ij}|$ for $j = 1, 2, \ldots, n$ in a similar way as $\boldsymbol{B}$. We finally set $\boldsymbol{A} = \boldsymbol{B}^{-1}\boldsymbol{C}$ because it is guaranteed that any $\boldsymbol{A}$ constructed in this way is a P_0 matrix. As for the vector $\boldsymbol{b}$, we chose the values of its entries so that (1) has the unique solution $\boldsymbol{x}^* = (2, 8, 5, 6, 3, 4, 9, 1)^{\mathrm{T}}$.

Each agent i sets the initial solution $\boldsymbol{x}^i(0)$ to $\boldsymbol{0}$, and then updates the solution by using (4) and (6). Computation

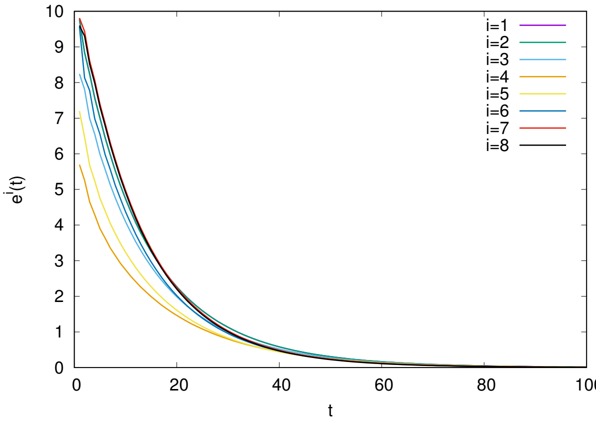

Figure 3: Behavior of the error $e^i(t) = \|\boldsymbol{x}^i(t) - \boldsymbol{x}^*\|$ when $G = G_1$.

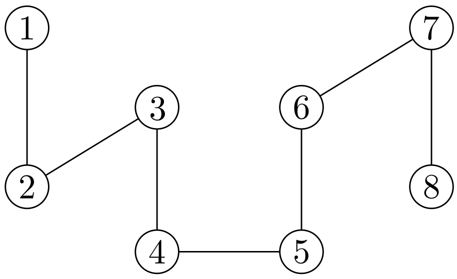

Figure 4: Communication graph G_2.

of g_i^{-1} is done by the inexact Newton method proposed by Solodov and Svaiter [7] under the following setting: $\beta = 0.5$, $\lambda = 0.01$, G_k is a number randomly chosen from $[0, 1]$, $\mu_k = 0.01$, $\rho_k = 0.01$, the initial solution is set to $x_i^i(t-1)$, and the stopping condition is set to $d^k \leq 10^{-4}$. For more details of the inexact Newton method, see [7].

Figure 3 shows how the error $e^i(t) = \|\boldsymbol{x}^i(t) - \boldsymbol{x}^*\|$ between the solution $\boldsymbol{x}^i(t)$ of agent i at time t and the true solution $\boldsymbol{x}^*$ varies with t for $i = 1, 2, \ldots, 8$. It is clear from the figure that $e^i(t)$ converges to zero for all i, which means that all agents successfully find the true solution $\boldsymbol{x}^*$.

In the second experiment, we used the same setting as in the first experiment except the communication graph. We set the communication graph G to G_2 in Fig. 4, which is a path graph obtained by removing three edges from G_1. Figure 5 shows how the error $e^i(t) = \|\boldsymbol{x}^i(t) - \boldsymbol{x}^*\|$ between the solution $\boldsymbol{x}^i(t)$ of agent i at time t and the true solution $\boldsymbol{x}^*$ varies with t for $i = 1, 2, \ldots, 8$. It is clear from the figure that all agents successfully find the true solution $\boldsymbol{x}^*$.

5. Convergence Analysis

In this section, we show theoretically that if the nonlinear function $\boldsymbol{f}(\boldsymbol{x})$ and the matrix $\boldsymbol{A}$ satisfies a few additional assumptions then, by using the proposed algorithm, all agents can find the unique solution of (1).

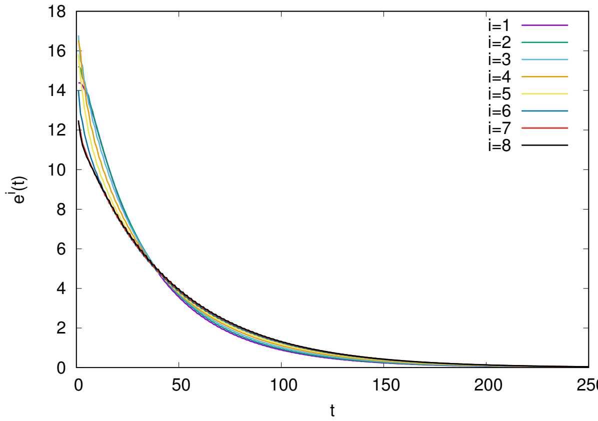

Figure 5: Behavior of the error $e^i(t) = \|\boldsymbol{x}^i(t) - \boldsymbol{x}^*\|$ when $G = G_2$.

Theorem 1 Suppose that $f_i(x_i)$ is continuously differentiable for $i = 1, 2, \ldots, n$ and there exists a positive constant δ such that

$$f_i'(x_i) \geq \delta > 0, \quad i = 1, 2, \ldots, n. \tag{7}$$

If the communication graph G of the multi-agent network is connected and the matrix $\boldsymbol{A}$ satisfies

$$\delta + a_{ii} > \sum_{j=1,\, j \neq i}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n, \tag{8}$$

then we have

$$\lim_{t \to \infty} \boldsymbol{x}^i(t) = \boldsymbol{x}^*, \quad i = 1, 2, \ldots, n \tag{9}$$

where $\boldsymbol{x}^*$ is the unique solution of (1).

Proof: Let $\tilde{\boldsymbol{a}}_i$ be the row vector obtained from $\boldsymbol{a}_i$ by setting the i-th entry a_{ii} to zero. It follows from (7) and (8) that

$$\begin{aligned}
\left|x_i^i(t+1) - x_i^*\right| &= \left|g_i^{-1}(b_i - \tilde{\boldsymbol{a}}_i \boldsymbol{w}^i(t)) - g_i^{-1}(b_i - \tilde{\boldsymbol{a}}_i \boldsymbol{x}^*)\right| \\
&= \left|g_i^{-1}\left(b_i - \frac{1}{|\mathcal{N}_i|}\tilde{\boldsymbol{a}}_i \sum_{k \in \mathcal{N}_i} \boldsymbol{x}^k(t)\right) - g_i^{-1}\left(b_i - \tilde{\boldsymbol{a}}_i \boldsymbol{x}^*\right)\right| \\
&\leq \frac{1}{\delta + a_{ii}} \left|\frac{1}{|\mathcal{N}_i|}\tilde{\boldsymbol{a}}_i \sum_{k \in \mathcal{N}_i} \boldsymbol{x}^k(t) - \tilde{\boldsymbol{a}}_i \boldsymbol{x}^*\right| \\
&= \frac{1}{(\delta + a_{ii})|\mathcal{N}_i|} \left|\tilde{\boldsymbol{a}}_i \sum_{k \in \mathcal{N}_i} (\boldsymbol{x}^k(t) - \boldsymbol{x}^*)\right| \\
&\leq \frac{1}{(\delta + a_{ii})|\mathcal{N}_i|} \sum_{k \in \mathcal{N}_i} \sum_{j=1,\, j \neq i}^{n} |a_{ij}| \left|x_j^k(t) - x_j^*\right| \\
&\leq \frac{\sum_{j=1,\, j \neq i}^{n} |a_{ij}|}{\delta + a_{ii}} \max_{k \in \mathcal{N}_i,\, j \in \{1,2,\ldots,n\} \setminus \{i\}} \left|x_j^k(t) - x_j^*\right| \\
&\leq \max_{k \in \mathcal{N}_i,\, j \in \{1,2,\ldots,n\} \setminus \{i\}} \left|x_j^k(t) - x_j^*\right|
\end{aligned} \tag{10}$$

holds for $i = 1, 2, \ldots, n$. Also, for any pair of i and j such that $i \neq j$, we have

$$\begin{aligned} \left|x_j^i(t+1) - x_j^*\right| &= \left|w_j^i(t) - x_j^*\right| \\ &= \left|\frac{1}{|\mathcal{N}_i|} \sum_{k \in \mathcal{N}_i} x_j^k(t) - x_j^*\right| \\ &\leq \frac{1}{|\mathcal{N}_i|} \sum_{k \in \mathcal{N}_i} \left|x_j^k(t) - x_j^*\right| \\ &\leq \max_{k \in \mathcal{N}_i} \left|x_j^k(t) - x_j^*\right|. \end{aligned} \tag{11}$$

Let us now define $Z(t)$ as $Z(t) = \max_{i,j \in \{1,2,\ldots,n\}} \left|x_j^i(t) - x_j^*\right|$. Then we have

$$Z(t+1) \leq Z(t), \quad t = 1, 2, \ldots \tag{12}$$

from (10) and (11).

Next we show that if $Z(t) > 0$ then $Z(t+n) < Z(t)$. We first see from (10) and (12) that

$$\begin{aligned} \left|x_i^i(t+n) - x_i^*\right| &\leq \frac{\sum_{j=1, j\neq i}^n |a_{ij}|}{\delta + a_{ii}} Z(t+n-1) \\ &< Z(t+n-1) \\ &\leq Z(t) \end{aligned}$$

holds for $i = 1, 2, \ldots, n$. Suppose that $i \neq j$. If there exists a $k \in \mathcal{N}_i$ such that $\left|x_j^k(t) - x_j^*\right| < Z(t)$ then it follows from (11) that $\left|x_j^i(t+1) - x_j^*\right| < Z(t)$. For the same reason, we have $\left|x_j^i(t+2) - x_j^*\right| < Z(t)$. Repeating this argument, we finally have $\left|x_j^i(t+n) - x_j^*\right| < Z(t)$. If $\left|x_j^k(t) - x_j^*\right| = Z(t)$ holds for all $k \in \mathcal{N}_i$ then it follows from (11) that $\left|x_j^i(t+1) - x_j^*\right| = Z(t)$. However, because $\left|x_j^j(t+1) - x_j^*\right| < Z(t)$, for all vertices j_1 adjacent to vertex j, $\left|x_j^{j_1}(t+2) - x_j^*\right| < Z(t)$ holds. This implies that for all vertices j_2 such that the shortest path length from vertex j is two, $\left|x_j^{j_2}(t+3) - x_j^*\right| < Z(t)$ holds. Repeating this argument and making use of the assumption that G is connected, we finally have $\left|x_j^i(t+n) - x_j^*\right| < Z(t)$.

Let $\hat{Z}(t) \triangleq Z(nt)$ for all $t \in \mathbb{Z}_+$. Then, we see from the above discussion that the sequence $\{\hat{Z}(t)\}_{t=1}^{\infty}$ is monotone decreasing and $\hat{Z}(t+1) = \hat{Z}(t)$ if and only if $\hat{Z}(t) = 0$, that is, $\boldsymbol{x}^i(nt) = \boldsymbol{x}^*$ for $i = 1, 2, \ldots, n$. Also, for all $i, j \in \{1, 2, \ldots, n\}$ and all $t \in \mathbb{Z}_+$, we have $|x_j^i(nt) - x_j^*| \leq Z(1)$, which means that $(\boldsymbol{x}^1(nt), \boldsymbol{x}^2(nt), \ldots, \boldsymbol{x}^n(nt))$ is included in a bounded and closed subset in $(\mathbb{R}^n)^n$. Furthermore, $(\boldsymbol{x}^1((n+1)t), \boldsymbol{x}^2((n+1)t), \ldots, \boldsymbol{x}^n((n+1)t))$ depends continuously on $(\boldsymbol{x}^1(nt), \boldsymbol{x}^2(nt), \ldots, \boldsymbol{x}^n(nt))$. Therefore, by Zangwill's global convergence theorem [8], we have

$$\lim_{t\to\infty} \boldsymbol{x}^i(nt) = \boldsymbol{x}^*, \quad i = 1, 2, \ldots, n$$

which means that $\lim_{t\to\infty} \hat{Z}(t) = 0$. From this result and the monotonicity of the sequence $\{Z(t)\}_{t=1}^{\infty}$, we can conclude that $\lim_{t\to\infty} Z(t) = 0$ which is equivalent to (9). □

6. Conclusion

We have proposed a distributed algorithm for solving Sandberg-Willson equations. The authors confirmed through a large number of numerical experiments that the unique solution is always found by the proposed algorithm. However, we have only proved the validity of the algorithm in a special case. Further theoretical analysis is needed to understand the behavior of the proposed algorithm.

References

[1] R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[2] A. Nedić, A. Ozdaglar and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[3] S. Mou and A. S. Morse, "A fixed-neighbor, distributed algorithm for solving a linear algebraic equation," *Proceedings of 2013 European Control Conference*, pp. 2269–2273, 2013.

[4] K. Yamane and N. Takahashi, "Proposal of a truly decentralized algorithm for estimating algebraic connectivity of multi-agent networks," *Proceedings of 2015 International Symposium on Nonlinear Theory and its Applications*, pp. 277–280, 2015.

[5] K. Endo and N. Takahashi, "A new decentralized discrete-time algorithm for estimating algebraic connectivity of multiagent networks," *Proceedings of 2016 Asia Pacific Conference on Circuits and Systems*, pp. 232–235, 2016.

[6] I. W. Sandberg and A. N. Willson, Jr., "Some theorems on properties of dc equations of nonlinear networks," *Bell System Technical Journal*, vol. 48, pp. 1–34, January 1969.

[7] M. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations," *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp. 355–369, 1998.

[8] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, 1969.