

# Distributed Alternating Direction Method of Multipliers for Multi-Objective Optimization

Hui Deng\*, Yangdong Xu

Key Laboratory of Intelligent Analysis and Decision on Complex Systems, Chongqing University of Posts and Telecommunications, Chongqing, China  
Email: \*dhS190602017@163.com

**How to cite this paper:** Deng, H. and Xu, Y.D. (2022) Distributed Alternating Direction Method of Multipliers for Multi-Objective Optimization. *Advances in Pure Mathematics*, 12, 249-259.

<https://doi.org/10.4236/apm.2022.124019>

**Received:** February 20, 2022

**Accepted:** April 3, 2022

**Published:** April 6, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

In this paper, a distributed algorithm is proposed to solve a kind of multi-objective optimization problem based on the alternating direction method of multipliers. Compared with the centralized algorithms, this algorithm does not need a central node. Therefore, it has the characteristics of low communication burden and high privacy. In addition, numerical experiments are provided to validate the effectiveness of the proposed algorithm.

## Keywords

Alternating Direction Method of Multipliers, Distributed Algorithm, Multi-Objective Optimization, Multi-Agent System

---

## 1. Introduction

In the network age, there exist a variety of network systems, such as communication network system, transportation network system, power network system and sensor network system. These network systems are usually called multi-agent systems. A multi-agent system is a large-scale network system associated with multiple agents. In the multi-agent system, each individual with communication, computing, perception, decision-making, learning and execution abilities is called an agent, also known as a node in the network topology. In a broad sense, agents can be a robot, an aircraft, a computer and other entities, which can interact with each other through network connections and cooperate to complete some complex tasks.

Many optimization problems encountered in multi-agent systems, such as the cyber-physical social system [1], classification [2], task scheduling [3] and energy system [4], often involve multiple conflicting objectives. This kind of optimization problem is called Multi-Objective Optimization Problem (MOP).

There are many ways to solve multi-objective optimization problems, which can be divided into three types: scalarization methods, classical optimization algorithms and intelligent algorithms. Scalarization methods transform complex multi-objective optimization problems into single-objective optimization problems, containing the main objective method, linear weighting method, minimax method, ideal point method and so on [5]. Classical optimization algorithms extend the algorithms for solving single-objective optimization problems to multi-objective optimization problems, including steepest descent method [6], Newton method [7] [8], proximal point method [9] [10], projected gradient method [11] [12] and penalty-type function method [13]. Intelligent algorithms include the genetic algorithm [14], evolutionary algorithm [15] [16] and so on.

Among these methods, scalarization methods are widely used because their models are easy to understand and relatively easy to solve. Among them, the linear weighting method is one of the commonly used approaches to solve multi-objective optimization problems. It can also coordinate the preference settings of multiple targets, which increases the subjectivity factor to some extent.

Most of the above optimization algorithms are centralized, and such methods need a central node to collect data and calculate the optimal decision. For example, in the electric-gas integrated energy system, a joint control center should be introduced as the decision-making mechanism for unified control of the electric-gas coupling system. Therefore, the communication burden and computation cost of the central node are high, and its scalability, robustness and adaptability are poor [17]. In contrast, distributed optimization algorithms are a kind of decentralized algorithms, which only require the network to be a general-connected graph, but do not need the central node. They decompose a large optimization problem into several sub-problems and compute each sub-problem, so as to optimize the original problem. Therefore, distributed optimization algorithms have high computational efficiency, small communication volume and strong reliability [18].

There are many distributed algorithms, including discrete algorithms and continuous algorithms [19] [20] [21] [22]. Among them, the discrete distributed algorithms can be divided into two kinds. The first one is the original algorithm [23] [24] [25], which weighted averages local estimates with neighboring states to make local states consistent. One of the great advantages of these methods is that they require less computation, but their convergence speed is slow and their accuracy is relatively low. The other is the dual algorithm that includes the augmented Lagrange method [26] and Alternating Direction Method of Multipliers (ADMM) [27]-[32]. Because each node needs to solve a sub-problem, the dual algorithm usually requires a relatively large amount of computation, but they can quickly converge to the exact optimal solution.

As an important method of distributed optimization algorithm, the alternating direction multiplier method has the characteristics of protecting data privacy and fast convergence. It has been shown that even for non-smooth and non-convex cost functions, the alternating direction multiplier method can achieve satisfac-

tory convergence speed [33]. In addition, it is robust to noise and error [34].

Based on the above reasons, this paper proposes a distributed alternating direction method of multipliers to solve a class of multi-objective optimization problems in multi-agent systems. This algorithm does not need the central node, and thus it can effectively avoid the shortcomings of the centralized algorithms. As a result, the algorithm proposed in this paper has the characteristics of low communication cost, strong privacy, fast convergence and so on.

The rest of this paper is organized as follows. In Section 2, a multi-objective optimization problem is introduced and it is transformed into a single-objective optimization problem. Based on this single-optimization objective problem, a distributed ADMM algorithm is proposed in Section 3. In Section 4, numerical experiments are provided to verify the effectiveness of the proposed algorithm. Finally, the conclusions of this paper are presented in Section 5.

## 2. Problem Model

This paper studies the following multi-objective optimization problem:

$$\min_y \sum_{i=1}^n F_i(y) \tag{1}$$

where  $y \in \mathbb{R}^p$  is the global optimization variable,  $n$  is the number of agents in the multi-agent system and  $F_i : \mathbb{R}^p \rightarrow \mathbb{R}^q$  is a convex local cost function. Each  $F_i$  is known only by agent  $i$  and the agents cooperatively solve the multi-objective optimization problem.

The network topology of the multi-agent system is assumed to be a general undirected connected graph, which is described as  $G = \{V, E\}$ , where  $V$  denotes the set of agents,  $E$  denotes the set of the edges and  $|V| = n$ ,  $|E| = m$ . These agents are arranged from 1 to  $n$ . Edges between  $i$  and  $j$  with  $i < j$  are represented by  $(i, j)$  or  $e_{ij}$  and  $(i, j) \in E$  means that agents  $i$  and  $j$  can exchange data with each other. The neighbors of agent  $i$  are denoted by  $N(i) := \{j \in V \mid (i, j) \in E \text{ or } (j, i) \in E\}$  and  $d_i = |N(i)|$ .

The edge-node incidence matrix of the network  $G$  is denoted by  $\tilde{A} \in \mathbb{R}^{m \times n}$ . The row in  $A$  that corresponds to the edge  $e_{ij}$  is denote by  $[\tilde{A}]^{e_{ij}}$ , which is defined by:

$$[\tilde{A}]_k^{e_{ij}} = \begin{cases} 1, & \text{if } k = i, \\ -1, & \text{if } k = j, \\ 0, & \text{otherwise.} \end{cases}$$

Here, the edges of the network are sorted by the order of their corresponding agents. For instance, the edge-node incidence matrix of the network  $G$  in **Figure 1** is given by:

$$\tilde{A} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

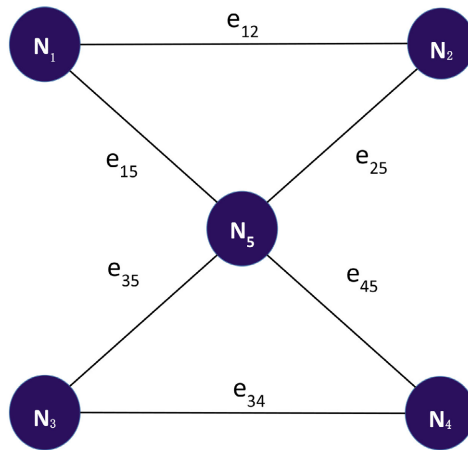


Figure 1. An example of the network  $G$ .

According to the edge-node incidence matrix, the extended edge-node incidence matrix  $A$  of the network  $G$  is given by:

$$A := \tilde{A} \otimes I_p = \begin{pmatrix} \tilde{a}_{11} I_p & \cdots & \tilde{a}_{1n} I_p \\ \vdots & \ddots & \vdots \\ \tilde{a}_{m1} I_p & \cdots & \tilde{a}_{mn} I_p \end{pmatrix} \in \mathbb{R}^{mp \times np},$$

where  $\otimes$  denotes the Kronecker product. Obviously,  $A$  is a block matrix with  $m \times n$  blocks of  $p \times p$  matrix.

By introducing separating decision variable  $x_i$  for each agent  $i = 1, 2, \dots, n$ , problem (1) has the following form:

$$\begin{aligned} \min_x F(x) &:= \sum_{i=1}^n F_i(x_i) \\ \text{s.t. } x_i &= x_j, \quad \forall (i, j) \in E, \end{aligned} \tag{2}$$

where  $x = [x_1^T, x_2^T, \dots, x_n^T]^T \in \mathbb{R}^{np \times 1}$ . Clearly, the problem (2) is equivalent to problem (1) if  $G$  is connected.

With the help of the extended edge-node incidence matrix  $A$ , the problem (2) can be rewritten in the following compact form:

$$\begin{aligned} \min_x F(x) \\ \text{s.t. } Ax &= 0. \end{aligned} \tag{3}$$

Denote the feasible region of problem (3) by  $R = \{x \in \mathbb{R}^{np \times 1} \mid Ax = 0\}$ , and thus the definition of effective solution (or weak effective solution) to problem (3) is given below.

**Definition 2.1.** [5] If there does not exist  $x \in R$ , such as  $F(x) \preceq F(x^*)$  (or  $F(x) \prec F(x^*)$ ), where  $x^* \in R$ , then  $x^*$  is the effective solution (or weak effective solution) of problem (3). Here,  $\preceq$  and  $\prec$  are taken element-wise.

In order to solve the multi-objective optimization problem by alternating direction multiplier method, it needs to be transformed into a single-objective optimization problem. By the linear weighting method, the single-objective optimization problem of problem (3) is constructed as follows:

$$\begin{aligned} & \min_x \theta^T F(x) \\ & \text{s.t. } Ax = 0. \end{aligned} \tag{4}$$

Here,  $\theta = (\theta_1, \theta_2, \dots, \theta_q)^T \in \Lambda^+$  or  $\Lambda^{++}$ , where:

$$\begin{aligned} \Lambda^+ &= \left\{ (\theta_1, \theta_2, \dots, \theta_q)^T \mid \theta_i \geq 0, \sum_{j=1}^q \theta_j = 1 \right\}, \\ \Lambda^{++} &= \left\{ (\theta_1, \theta_2, \dots, \theta_q)^T \mid \theta_i > 0, \sum_{j=1}^q \theta_j = 1 \right\}. \end{aligned}$$

**Remark 2.1.** [5] For each given  $\theta \in \Lambda^{++}$  (or  $\Lambda^+$ ), the optimal solution of problem (4) must be a efficient (or weakly efficient) solution to problem (3).

Denote  $f^\theta(x) := \theta^T F(x)$ ,  $f_i^\theta(x_i) := \theta^T F_i(x_i)$ , then problem (4) can be rewritten as:

$$\begin{aligned} & \min_x f^\theta(x) := \sum_{i=1}^n f_i^\theta(x_i) \\ & \text{s.t. } Ax = 0. \end{aligned} \tag{5}$$

**Remark 2.2.** According to the convexity of  $F_i$ , each component  $F_i^j$  of  $F_i$  is a convex function, and thus  $f_i^\theta$  is also convex.

### 3. Algorithm Design

The augmented Lagrangian function of problem (5) is defined by:

$$\mathcal{L}_{\theta, \rho}(x, \lambda) = f^\theta(x) - \lambda^T Ax + \frac{\rho}{2} \|Ax\|^2, \tag{6}$$

where  $\rho$  is the penalty parameter of the constraint  $Ax = 0$  and  $\rho > 0$ . The Jacobi-Proximal ADMM algorithm for problem (5) is designed as follows:

$$x^{k+1} := \arg \min_x \mathcal{L}_{\theta, \rho}(x, \lambda^k) + \frac{1}{2} \|x - x^k\|_P^2, \tag{7}$$

$$\lambda^{k+1} := \lambda^k - \rho Ax^{k+1}, \tag{8}$$

where the proximal matrix  $P$  is a block diagonal matrix and  $\|x\|_P^2 = x^T Px$ . Combining the linear and quadratic terms of the augmented Lagrangian function, (7) can be rewritten as:

$$x^{k+1} := \arg \min_x f^\theta(x) + \frac{\rho}{2} \left\| Ax - \frac{1}{\rho} \lambda^k \right\|^2 + \frac{1}{2} \|x - x^k\|_P^2. \tag{9}$$

Denoting the block corresponding to the agent  $i$  by  $P_i \in \mathbb{R}^{p \times p}$ , one can rewrite (9) in the following form:

$$x^{k+1} := \arg \min_x f^\theta(x) + \frac{\rho}{2} \left\| Ax - \frac{1}{\rho} \lambda^k \right\|^2 + \frac{1}{2} \sum_{i=1}^n \|x_i - x_i^k\|_{P_i}^2. \tag{10}$$

It is worth noting that each row of the constraint  $Ax = 0$  corresponds to one edge  $e_{pq}$  in the set  $E$ . Hence, one has  $x_p - x_q = 0$  for any  $(p, q) \in E$ , where  $p < q$ . Denote the multiplier corresponding to the edge  $e_{pq}$  by  $\lambda_{e_{pq}}$ . Then, the iteration of  $x$  can be refined as:

$$x^{k+1} := \arg \min_x \sum_{i=1}^n f_i^\theta(x_i) + \frac{\rho}{2} \sum_{(p,q) \in E} \left\| x_p - x_q - \frac{1}{\rho} \lambda_{e_{pq}}^k \right\|^2 + \sum_{i=1}^n \frac{1}{2} \|x_i - x_i^k\|_{P_i}^2. \quad (11)$$

Thus, the iteration of  $x$  have the following distributed version:

$$x_i^{k+1} := \arg \min_{x_i} f_i^\theta(x_i) + \frac{\rho}{2} \sum_{j \in P(i)} \left\| x_j^k - x_i - \frac{1}{\rho} \lambda_{e_{ji}}^k \right\|^2 + \frac{\rho}{2} \sum_{j \in S(i)} \left\| x_i - x_j^k - \frac{1}{\rho} \lambda_{e_{ij}}^k \right\|^2 + \frac{1}{2} \|x_i - x_i^k\|_{P_i}^2. \quad (12)$$

Similarly, by the characteristic of matrix  $A$ , (8) can be rewritten as:

$$\lambda_{e_{pq}}^{k+1} := \lambda_{e_{pq}}^k - \rho(x_p^{k+1} - x_q^{k+1}), e_{pq} \in E. \quad (13)$$

Therefore, the iterations of the multipliers are also distributed to the agents.

The distributed Jacobi-Proximal ADMM algorithm for problem (5) is described as **Algorithm 1**.

---

**Algorithm 1:** Distributed Jacobi-Proximal ADMM Algorithm (DJP-ADMM)

---

**Initialization::** Choose appropriate  $P_i, \rho, \{x_i^0\}, \{\lambda_{e_{ij}}^0\}$  and  $\{\lambda_{e_{ij}}^0\}, i = 1, 2, \dots, n$ .

- 1  $0 \leftarrow k;$
  - 2 **while** some stop criteria are not met **do**
  - 3     **for**  $i \leftarrow 1$  **to**  $n$  **do**
  - 4         Update  $x_i^{k+1}$  by
 
$$x_i^{k+1} := \arg \min_{x_i} f_i^\theta(x_i) + \frac{\rho}{2} \sum_{j \in P(i)} \|x_j^k - x_i - \frac{1}{\rho} \lambda_{e_{ji}}^k\|^2 + \frac{\rho}{2} \sum_{j \in S(i)} \|x_i - x_j^k - \frac{1}{\rho} \lambda_{e_{ij}}^k\|^2 + \frac{1}{2} \|x_i - x_i^k\|_{P_i}^2;$$
  - 5         **for**  $j \in P(i)$  **do**
  - 6             Update  $\lambda_{e_{ji}}^{k+1}$  by
 
$$\lambda_{e_{ji}}^{k+1} := \lambda_{e_{ji}}^k - \rho(x_j^{k+1} - x_i^{k+1});$$
  - 6      $k \leftarrow k + 1;$
- 

### 4. Numerical Experiment

Consider the following multi-objective optimization problem:

$$\min_y \sum_{i=1}^n F_i(y) = \left[ \sum_{i=1}^n F_i^1(y), \sum_{i=1}^n F_i^2(y) \right]^T \quad (14)$$

where  $F_i^1(y) = \frac{1}{2}(y - \alpha_i)^2, F_i^2(y) = \frac{1}{2}(y - \beta_i)^2, i = 1, 2, \dots, n$  and  $y \in \mathbb{R}$ . Ob-

viously, the solutions to these two objective functions are  $y^* = \bar{\alpha} = \frac{1}{n} \sum_{i=1}^n \alpha_i$

and  $y^* = \bar{\beta} = \frac{1}{n} \sum_{i=1}^n \beta_i$ , respectively. By the convexity of these functions, the weak

efficient solution of problem (14) is  $[\bar{\alpha}, \bar{\beta}]$  and its efficient solution is  $(\bar{\alpha}, \bar{\beta})$  if  $\bar{\alpha} < \bar{\beta}$ .

Assume the connected network  $G = \{V, E\}$  with  $n$  agents and  $m$  edges. Each edge is generated randomly. The connectivity ratio of the network  $G$  is denoted

by  $d = \frac{2m}{n(n-1)}$ . The multi-objective optimization problem (14) can be reformulated into a distributed version:

$$\begin{aligned} \min_x F(x) &:= \left[ \frac{1}{2} \sum_{i=1}^n (x_i - \alpha_i)^2, \frac{1}{2} \sum_{i=1}^n (x_i - \beta_i)^2 \right]^T \\ \text{s.t. } x_i &= x_j, \quad \forall (i, j) \in E, \end{aligned} \tag{15}$$

where  $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ .

By linearly weighting the objective functions, problem (15) can be transformed into the following single-objective optimization problem:

$$\begin{aligned} \min_x f^\theta(x) &:= \frac{1}{2} \sum_{i=1}^n \left[ \theta (x_i - \alpha_i)^2 + (1-\theta) (x_i - \beta_i)^2 \right] \\ \text{s.t. } Ax &= 0. \end{aligned} \tag{16}$$

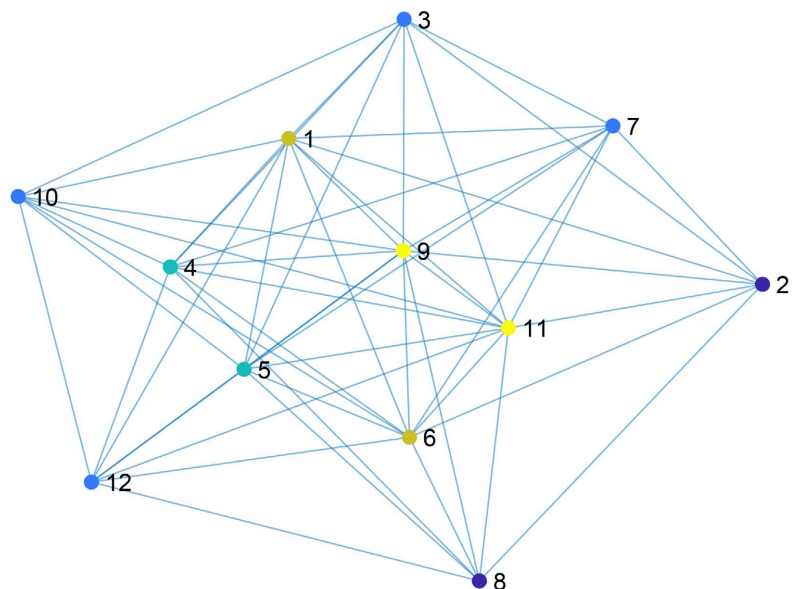
where  $A$  is the extended edge-node incidence matrix of the network  $G$ ,  $\theta \in \mathbb{R}$  and  $f^\theta(x)$  is convex. Therefore, **Algorithm 1** can be used to solve problem (16).

The proximal matrix of **Algorithm 1** is set by  $P_i = \rho d_i I$ . In this case, the iteration of  $x$  has a closed-form solution, which is shown as follows:

$$x_i^{k+1} = \frac{\rho d_i x_i^k + \rho \sum_{j \in N(i)} x_j^k + \sum_{j \in S(i)} \lambda_{ij}^k - \sum_{j \in P(i)} \lambda_{ji}^k + \theta \alpha_i + (1-\theta) \beta_i}{1 + 2\rho d_i},$$

where  $d_i$  is the number of neighbors of the agent  $i$ .

To verify the effectiveness of **Algorithm 1**, we generate a network with  $n = 12$  nodes and  $d = 0.8$ . The network generated is shown as **Figure 2**. The value of the parameters are set as:  $\alpha_i = 1, 3, \dots, 23$  and  $\beta_i = 2, 4, \dots, 24$ . The graph of the two objective functions is shown as **Figure 3**. Obviously, the weak efficient solution is  $[12, 13]$  and the efficient solution is  $(12, 13)$ .



**Figure 2.** The network of problem (14).

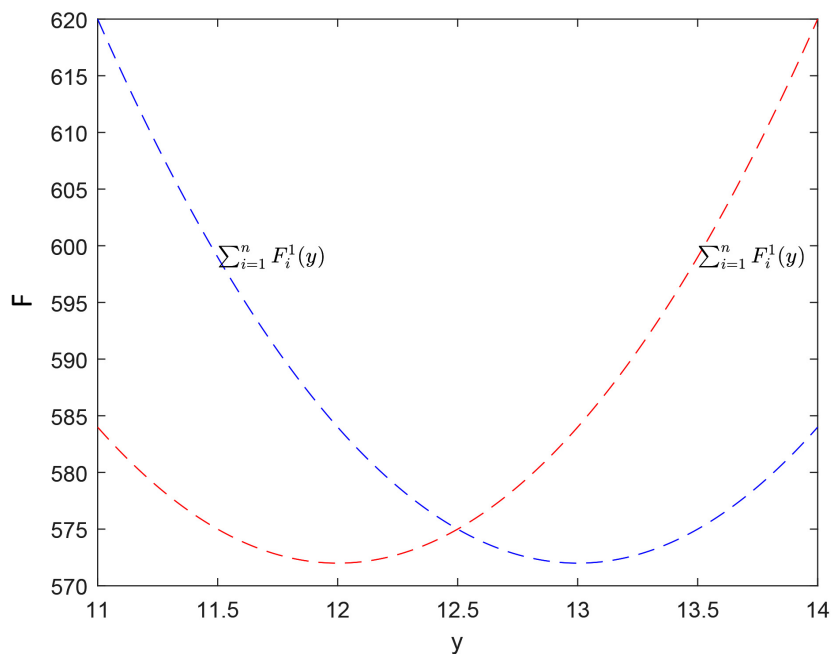


Figure 3. The graph of the two objective functions.

Table 1. Different weights and solutions to the corresponding problems.

$\theta$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Solution	13	12.9	12.8	12.7	12.6	12.5	12.4	12.3	12.2	12.1	12

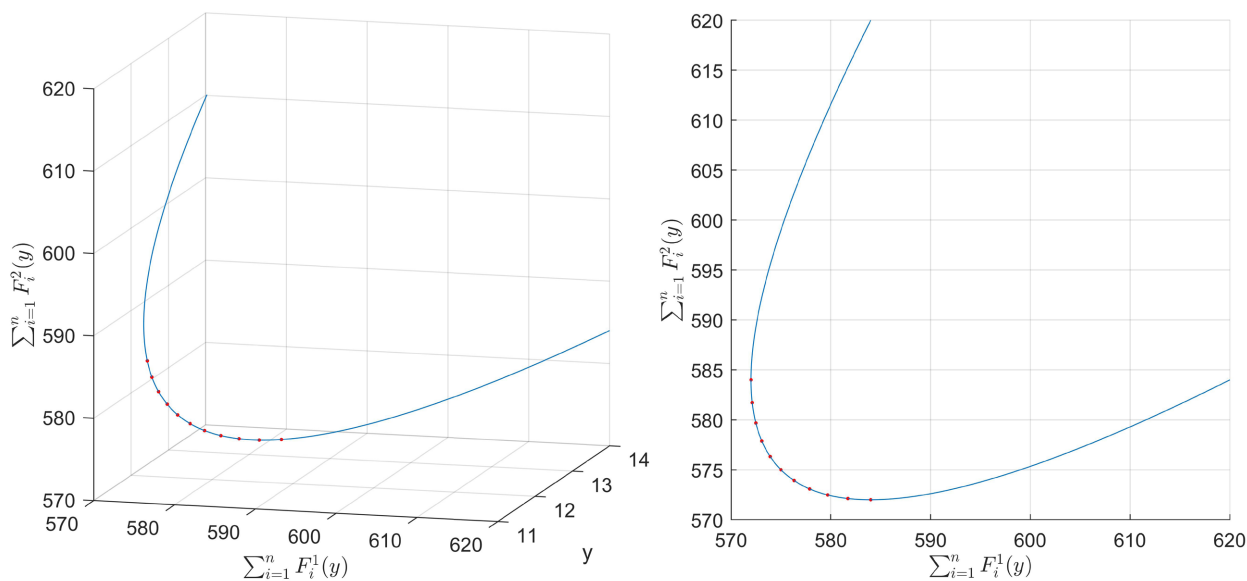


Figure 4. The graph of the objective function and the solutions of problem (16).

Different weights are selected, and then **Algorithm 1** is used to solve the corresponding single-objective problem (16). The results obtained are shown in **Table 1**. Besides, the graph of the objective function and the solutions of problem (16) is shown as **Figure 4**.



## 5. Conclusions

In this paper, a distributed ADMM algorithm is put forward to solve a kind of multi-objective optimization problem. This algorithm does not need the central node, and thus its communication cost is low and privacy is high. In addition, the effectiveness of the algorithm is verified by some numerical experiments.

There are a few distributed algorithms for multi-objective optimization problems. The method adopted in this paper is to transform the multi-objective optimization problem into a single-objective optimization problem through the linear weighting method, and then solve this single objective optimization problem, instead of solving the multi-objective optimization problem directly. Therefore, it is a challenge to design a distributed algorithm for directly solving multi-objective optimization problems and to verify its convergence, which is also worth exploring.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Wang, G.G., Cai, X., Cui, Z., Min, G. and Chen, J. (2020) High Performance Computing for Cyber Physical Social Systems by Using Evolutionary Multi-Objective Optimization Algorithm. *IEEE Transactions on Emerging Topics in Computing*, **8**, 20-30. <https://doi.org/10.1109/TETC.2017.2703784>
- [2] Singh, D., Kumar, V., Vaishali, Z. and Kaur, M. (2020) Classification of COVID-19 patients from Chest CT Images Using Multi-Objective Differential Evolution-Based Convolutional Neural Networks. *European Journal of Clinical Microbiology & Infectious Diseases*, **39**, 1379-1389. <https://doi.org/10.1007/s10096-020-03901-z>
- [3] Abualigah, L. and Diabat, A. (2021) A Novel Hybrid Antlion Optimization Algorithm for Multi-Objective Task Scheduling Problems in Cloud Computing Environments. *Cluster Computing*, **24**, 205-223. <https://doi.org/10.1007/s10586-020-03075-5>
- [4] Wu, C.Y., Wang, J.Z., Chen, X.J., Du, P. and Yang, W.D. (2020) A Novel Hybrid System Based on Multi-Objective Optimization for Wind Speed Forecasting. *Renewable Energy*, **146**, 149-165. <https://doi.org/10.1016/j.renene.2019.04.157>
- [5] Jahn, J. (2011) Vector Optimization: Theory, Applications, and Extensions. 2nd Edition, Springer, Heidelberg. <https://doi.org/10.1007/978-3-642-17005-8>
- [6] Fliege, J. and Svaiter, B.F. (2000) Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, **51**, 479-494. <https://doi.org/10.1007/s001860000043>
- [7] Fliege, J., Graña Drummond, L.M. and Svaiter, B.F. (2009) Newtons Method for Multiobjective Optimization. *SIAM Journal on Optimization*, **20**, 602-626. <https://doi.org/10.1137/08071692X>
- [8] Wang, J.H., Hu, Y.H., Yu, C.K.W., Li, C. and Yang, X.Q. (2019) Extended Newton Methods for Multiobjective Optimization: Majorizing Function Technique and Convergence Analysis. *SIAM Journal on Optimization*, **29**, 2388-2421. <https://doi.org/10.1137/18M1191737>
- [9] Bonnel, H., Iusem, A.N. and Svaiter, B.F. (2005) Proximal Methods in Vector Opti-

- mization. *SIAM Journal on Optimization*, **15**, 953-970.  
<https://doi.org/10.1137/S1052623403429093>
- [10] Bento, G.C., Ferreira, O.P. and Sousa, J.V.L. (2018) Proximal Point Method for a Special class of Nonconvex Multiobjective Optimization Functions. *Optimization Letters*, **12**, 311-320. <https://doi.org/10.1007/s11590-017-1114-0>
- [11] Drummond, L.M.G. and Iusem, A.N. (2004) A Projected Gradient Method for Vector Optimization Problems. *Computational Optimization and Applications*, **28**, 5-29. <https://doi.org/10.1023/B:COAP.0000018877.86161.8b>
- [12] Fazio, N.S. and Schuverdt, M.L. (2018) Convergence Analysis of a Nonmonotone Projected Gradient Method for Multiobjective Optimization Problems. *Optimization Letters*, **13**, 1365-1379. <https://doi.org/10.1007/s11590-018-1353-8>
- [13] Fukuda, E.H., Drummond, L.M.G. and Raupp, F.M.P. (2016) An External Penalty-Type Method for Multicriteria. *TOP*, **24**, 493-513. <https://doi.org/10.1007/s11750-015-0406-8>
- [14] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. <https://doi.org/10.1109/4235.996017>
- [15] Zhang, H., Zhou, A., Song, S., Zhang, Q., Gao, X. and Zhang, J. (2016) A Self-Organizing Multiobjective Evolutionary Algorithm. *IEEE Transactions on Evolutionary Computation*, **20**, 792-806. <https://doi.org/10.1109/TEVC.2016.2521868>
- [16] Liang, J.J., Xu, W.W., Yue, C.T., Yu, K.J., Song, H., Crisalle, O.C. and Qu, B.Y. (2018) Multimodal Multiobjective Optimization with Differential Evolution. *Swarm and Evolutionary Computation*, **44**, 1028-1059. <https://doi.org/10.1016/j.swevo.2018.10.016>
- [17] Yin, L.F. and Sun, Z.X. (2021) Multi-Layer Distributed Multi-Objective Consensus Algorithm for Multi-Objective Economic Dispatch of Large-Scale Multi-Area Interconnected Power Systems. *Applied Energy*, **300**, Article ID: 117391. <https://doi.org/10.1016/j.apenergy.2021.117391>
- [18] Chen, S.X., Garcia, A. and Shahrampour S. (2022) On Distributed Nonconvex Optimization: Projected Subgradient Method for Weakly Convex Problems in Networks. *IEEE Transactions on Automatic Control*, **67**, 662-675. <https://doi.org/10.1109/TAC.2021.3056535>
- [19] Droge, G., Kawashima, H. and Egerstedt, M.B. (2014) Continuous-Time Proportional-Integral Distributed Optimisation for Networked Systems. *Journal of Control and Decision*, **1**, 191-213. <https://doi.org/10.1109/TAC.2013.2278132>
- [20] Gharesifard, B. and Cortés, J. (2014) Continuous-Time Distributed Convex Optimization on Weight-Balanced Digraphs. *IEEE Transactions on Automatic Control*, **59**, 781-786. <https://doi.org/10.1109/TAC.2013.2278132>
- [21] Zhu, Y.N., Yu, W.W., Wen, G.H., Chen, G.R. and Ren, W. (2019) Continuous-time Distributed Subgradient Algorithm for Convex Optimization with General Constraints. *IEEE Transactions on Automatic Control*, **64**, 1694-1701. <https://doi.org/10.1109/TAC.2018.2852602>
- [22] Zhu, W. and Tian, H.B. (2021) Distributed Convex Optimization via Proportional-Integral-Differential Algorithm. *Measurement and Control*. <https://doi.org/10.1177/00202940211029332>
- [23] Nedic, A. and Ozdaglar, A. (2009) Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, **54**, 48-61. <https://doi.org/10.1016/j.automatica.2017.06.011>
- [24] Xi, C. and Khan, U.A. (2017) Distributed Subgradient Projection Algorithm over Directed Graphs. *IEEE Transactions on Automatic Control*, **62**, 3986-3992.

- <https://doi.org/10.1016/j.automatica.2017.06.011>
- [25] Liu, S., Zhang, Z.R. and Xie, L.H. (2017) Convergence Rate Analysis of Distributed Optimization with Projected Subgradient Algorithm. *Automatica*, **83**, 162-169. <https://doi.org/10.1016/j.automatica.2017.06.011>
- [26] Jakovetić, D., Xavier, J. and Moura, J.M.F. (2011) Cooperative Convex Optimization in Networked Systems: Augmented Lagrangian Algorithms with Directed Gossip Communication. *IEEE Transactions on Signal Processing*, **59**, 3889-3902. <https://doi.org/10.1109/TSP.2011.2146776>
- [27] Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2010) Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, **3**, 1-122. <https://doi.org/10.1561/22000000016>
- [28] Zhang, R. and Kwok, J.T. (2014) Asynchronous Distributed ADMM for Consensus Optimization. *Proceedings of the 31st International Conference on Machine Learning*, Beijing, 21-26 June 2014, 3689-3697.
- [29] Wei, E. and Ozdaglar, A. (2012) Distributed Alternating Direction Method of Multipliers. *Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control*, Maui, 10-13 December 2011, 5445-5450. <https://doi.org/10.1109/CDC.2012.6425904>
- [30] Yan, J.Q., Guo, F.H., Wen, C.Y. and Li, G.Q. (2020) Parallel Alternating Direction Method of Multipliers. *Information Sciences*, **507**, 185-196. <https://doi.org/10.1016/j.ins.2019.08.039>
- [31] Shi, W., Ling, Q., Yuan, K., Wu, G. and Yin, W. (2014) On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Transactions on Signal Processing*, **62**, 1750-1761. <https://doi.org/10.1109/TSP.2014.2304432>
- [32] Hong, M., Davood, H. and Zhao, M. (2017) Prox-PDA: The Proximal Primal-Dual Algorithm for Fast Distributed Nonconvex Optimization and Learning over Networks. *Proceedings of the 34th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 2402-2433.
- [33] Wang, Y., Yin, W. and Zeng, J. (2015) Global Convergence of ADMM in Nonconvex Nonsmooth Optimization. *Journal of Scientific Computing*, **78**, 26-93. <https://doi.org/10.1007/s10915-018-0757-z>
- [34] Simonetto, A. and Leus, G. (2014) Distributed Maximum Likelihood Sensor Network Localization. *IEEE Transactions on Signal Processing*, **62**, 1424-1437. <https://doi.org/10.1109/TSP.2014.2302746>