

Received November 18, 2020, accepted December 2, 2020, date of publication December 8, 2020, date of current version December 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043371

A Quick Algorithm to Determine 2-Optimality Consensus for Collectives

DAI THO DANG¹, NGOC THANH NGUYEN^{2,3}, (Senior Member, IEEE), AND DOSAM HWANG¹

¹Department of Computer Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea

²Department of Applied Informatics, Wrocław University of Science and Technology, 50-370 Wrocław, Poland

³Faculty of Information Technology, Nguyen Tat Thanh University, Ho Chi Minh City 70000, Vietnam

Corresponding authors: Dosam Hwang (dosamhwang@gmail.com) and Ngoc Thanh Nguyen (ngoc-thanh.nguyen@pwr.edu.pl)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B4009410.

ABSTRACT Nowadays, to solve a problem, people/systems typically use knowledge from different sources. A binary vector is a useful structure to represent knowledge states, and determining the consensus for a binary vector collective is helpful in many areas. However, determining a consensus that satisfies postulate 2-Optimality is an NP-hard problem; therefore, many heuristic algorithms have been proposed. The basic heuristic algorithm is the fastest in the literature, and most widely used to solve this problem. The computational complexity of the basic heuristic algorithm is $O(m^2n)$. In this study, we propose a quick algorithm (called QADC) to determine the 2-Optimality consensus. The QADC algorithm is developed based on a new approach for calculating the distances from a candidate consensus to the collective members. The computational complexity of the QADC algorithm has been reduced to $O(mn)$, and the consensus quality of QADC algorithm and the basic heuristic algorithm is the same.

INDEX TERMS Collective intelligence, collective knowledge, consensus, heuristic algorithm, 2-optimality consensus.

I. INTRODUCTION

Using knowledge from different sources for decision-making is getting popular [1]. For example, to decide on a problem in our life, people typically search for information on the Internet or ask the opinions of experts. In this way, they generally reach a suitable solution. Another example is a distributed detection system where agents share their logical options on events and make decisions of the system afterwards [2].

In general, a set of knowledge states from different sources is considered as a collective or profile. A collective consists of knowledge states of different agents, experts, or individuals referring to the same problem [3], [4]. Knowledge states of a collective are frequently in conflict [5]. For example, many meteorological stations forecast the weather for the same region. The forecasts from these meteorological stations are different. A collective consisting of knowledge states in conflict is called a conflict collective or conflict profile [6]. Besides, in a collective, the uncertainty of knowledge states often exists [7]. Thus, integrating knowledge states of a

collective into one consistent state, which is termed as a collective knowledge or consensus, is a complicated task.

Uncertainty is generally a situation that involves unknown or imperfect information [8], and it is an attribute of information [9]. To treat data with uncertainty, many theories have been introduced, such as rough set theory [10], soft set theory [11], fuzzy set theory [12].

A conflict takes place when at least two bodies have different options on the same subject [6]. Pawlak introduced the first formal model for conflict analysis. In this model, a set of tools for conflict analysis was presented [13]. This model is straightforward, and it does not allow the agents to express complex opinions [6]. Pawlak's model was enhanced by Skowron *et al.* [14]. This enhanced model defines local states of agents and examines various levels of conflicts. The attribute values in this model are atomic.

In [15], Nguyen *et al.* presented a general model for conflict and knowledge inconsistency. In this model, such factors as conflict representation and consistency measures for conflict collective are considered. Attribute values are multi-valued. To measure the consistency, eight postulates are proposed, and five consistency functions are defined. Besides,

The associate editor coordinating the review of this manuscript and approving it for publication was Ching-Ter Chang¹.

a methodology for using inconsistency of knowledge in a collective to determine its consensus is presented [5]. Determining consensus plays an essential role in many research areas [4], [16]. Consensus methods have been proposed for consensus determination [17].

In consensus methods, many postulates are used to define consensus functions in which the postulates 1-Optimality (or Kemeny median) and 2-Optimality are essential postulates [6]. However, no consensus function concurrently satisfies both the postulate 1-Optimality and the postulate 2-Optimality. The consensus satisfying the postulate 1-Optimality or the postulate 2-Optimality is the best representative of the collective. Besides, the consensus satisfying the postulate 2-Optimality is more uniform than that satisfying the postulate 1-Optimality. Although the criterion 2-Optimality is not well-known, in many cases, it is better than the criterion 1-Optimality [3], [6]. Determining the 2-Optimality consensus is an NP-hard problem [6], and heuristic algorithms have been developed for different data structures, such as a complex tree, partition, ontology, and binary vector [6], [18], [19].

In this study, a collective containing binary vectors is called a binary vector collective. Using binary vector to represent knowledge states and determining the consensus for binary vector collectives are used in distributed networks [2], [20], transport systems [21], [22], e-commerce [23], [24], bioinformatics [25]–[27] medicines [28], [29].

We consider the following situation. In this situation, n experts are asked about their opinions for a given problem. Each of them gives answers to m different questions. They can choose one from two possible answers: “Yes” or “No” (“Yes” is corresponding 1, and “No” is corresponding 0). The collective, which consists of opinions of experts, is represented by n binary vectors of the length equal to m . The second example refers to disease-disease relationships. A group of n patients has hepatocellular carcinoma. Each patient is expressed as a vector with m attributes, and each attribute represents a specific disease. If a patient has a specific disease, the value of the corresponding component is 1, otherwise 0. The collective, which consists of diseases of patients, is also stored as n binary vectors of the length equal to m . The problem of determining the consensus of these collectives is very useful in our lives. Therefore, in this study, we consider determining the consensus of such situations.

Heuristic algorithms have been introduced to determine the consensus for binary vector collectives [19], [30], [31]. The basic heuristic algorithm is the fastest and best algorithm, and it is the most widely used to solve this task. The computational complexity of this algorithm is $O(m^2n)$ where m is the length of members and n is the size of collectives [19].

Increasing the number of knowledge sources in the world and rapid development in information technology have facilitated the use of these sources for finding solutions to different problems; as a result, the collective size is increasing [32]. Determining the 2-Optimality consensus of a binary vector collective with a large size may require more time [4]. In this

scenario, there is a need to develop a quick algorithm to determine the 2-Optimality consensus of such a collective.

A. MOTIVATION

The motivation for this study is to find a new algorithm with lower computational complexity, whereas its consensus quality is higher than or the same as the consensus quality of the basic heuristic algorithm.

B. CONTRIBUTIONS OF THIS STUDY

The main contributions of this study are as follows:

- 1) In this study, we proposed a new method to calculate the sum of squared distances from a candidate consensus to members of the collective. The new method reduces the computational complexity of calculating this sum. Based on the advanced calculating method, we propose a quick algorithm whose computational complexity $O(mn)$.
- 2) We prove that the consensus quality of the QADC algorithm and that of the basic heuristic algorithm is the same both in theory and in the experiment.

The remainder of this paper is organized as follows. Section II provides a short review of the consensus problem and the 2-Optimality consensus. In section III, notions are introduced. Section IV presents a new approach to calculate the distance from a candidate consensus to collective members and the QADC algorithm. In section V, we measure the consensus quality of the QADC algorithm. Besides, we compare the QADC algorithm and the best algorithm of previous studies at both two aspects: running time and consensus quality. Finally, conclusions and future directions are provided in VI.

II. RELATED WORK

In computer science, the consensus problem has a long history [33], [34]. During recent years, it has become an attractive research area of interest [35]–[37]. The consensus problem forms the foundation of the field of distributed computing [34], in which all the agents communicate and update their states to achieve an agreement in the network. The consensus concept was introduced to control a community for describing the collective behavior of a group of systems, called multi-agent systems [38]–[40]. The consensus problem in multi-agent systems has been intensively studied in the control community and has been applied in vehicle formation, sensor networks, social networks [39], [41]. IoT has been rapidly developed during recent years. Its applications have the potential to affect every aspect of daily human life [42], many consensus problems exist in this field, for example, resource allocation [43], task allocation [44], [45], decision making in service-oriented [46].

There are numerous examples of consensus in economics. Prediction is an essential activity in several business processes, but it becomes challenging in the case that historical data are not available, such as forecasting demand for a new product. A prediction market that aggregates the opinions of the crowd is an efficient approach to solve

such problems [47]. Another example is the blockchain that creates a rapid change in economics, such as transactions, accounting, contracts, and records [48], [49]. The consensus problem forms the foundation of blockchain [50], [51], and consensus algorithms maintain the existence of blockchain, for example, PoW, PoS, and PoB [52].

In the medicines field, in the 1950s, consensus problems were born from a wish to synthesize clinician and expert opinions on clinical practice and research programs [53]. Nowadays, the consensus is widely used in this field, and it plays an essential role in diagnosis, management, treatment, health service, etc. [28], [29].

In bioinformatics there are numerous consensus problems, for example, in gene prediction [25], microRNA target ranking [27], disease-related gene ranking [27], drug-drug interaction [54], protein structure prediction [55], etc.

In general, there are three approaches to resolve consensus problems in previous studies [6]. In the axiomatic approach, many axioms are used to specify the conditions that should be met by consensus functions. First, seven conditions for the consensus functions were presented. Then, a set of ten postulates for consensus functions were defined, in which the postulates 1-Optimality and 2-Optimality are essential for determining consensus. The constructive approach resolves the consensus problem in two aspects: the relationship between elements and the structure of them. In the optimization approach, some optimality rules are used to define consensus functions [6].

In general, the consensus is the reasonable choice if the conflict participants refer to the same problem [3]. Assume that the opinions included in the conflict content reflect an unknown solution to a problem. We call this solution the proper solution to the problem, and the following two cases may take place [3]:

- 1) The proper solution is independent of the participants' opinions, such as the problem of predicting the GDP of a country.
- 2) The proper solution is dependent on the participants' opinions, such as the problem of the US presidential election.

In the first case, the proper solution to the problem exists; however, the participants do not know it. Thus, the participants may "guess" the proper solution [3]. In the second case, the participants' opinions decide the solution. In the two cases mentioned above, the consensus seems to have to satisfy the following conditions:

- It should best reflect the given versions, and/or
- It should be a good compromise that could be acceptable to the participants.

The first condition is appropriate for the first case. The reason is the opinions given by the participants that reflect the "hidden" and independent solution, but it is not known to what degree. The consensus should best reflect the participants' opinions. The best criterion for consensus choice is the criterion 1-Optimality. The second condition refers to the second case in which the proper problem solution is

dependent on those given by the participants. Thus, consensus should best represent the participants' opinions, and it should reflect the participants' opinions to an equal degree. The best criterion for consensus choice is the criterion 2-Optimality. Consensus chosen by the criterion 2-Optimality is more uniform than that chosen by the criterion 1-Optimality [3], [5].

The postulate 2-Optimality is used for many applications [56], [57]. In bioinformatics, for example, this postulate is applied for the multiple structure alignment problem that is described as follows: "Given a set of proteins X , compute a transformation (i.e., rotation and translation) for each protein, and generate a 2-Optimality consensus" [58]. Heuristic algorithms have been proposed for solving this problem in [58], [59]. Let n be the maximum length of the k proteins; then, the time complexities of the algorithms are $O(m^2k^2)$ or $O(kn^2 + kn^2)$, depending on the initial consensus [58], [59]. The problem of generating a consensus tree from a given collective of phylogenetic trees is used to reconstruct the evolutionary history of a set of organisms, and the postulate 2-Optimality is used for determining the consensus tree [60]. The time complexity of the algorithm MW and ADDTREE is $O(n^5)$; that of algorithm FITCH is $O(n^4)$; and that of the algorithms UNJ and NJ are $O(n^3)$, where n is the number of organisms [61].

In e-commerce, decision-making has become a necessary component of business activity [62]. In a typical case, the structure of decision representation is a set of decision elements that describes an economic problem. These elements are ordered in sequence to process during decision realization. The 2-Optimality consensus is used to develop multi-agent decision support systems. In [57], a heuristic algorithm with a computational complexity of $O(n^2m) + O(3nm)$ is introduced. This algorithm is implemented in the stock exchange and a-Trader multi-agent system [62], [63].

Determining the 2-Optimality consensus for binary vector collectives is an NP-hard problem, and the computational complexity of the brute-force algorithm to determine the optimal consensus is $O(n2^m)$; thus, heuristic algorithms have been developed [19]. First, the basic heuristic algorithm was introduced, and the computational complexity of this algorithm is $O(m^2n)$. Then, two additional heuristic algorithms were developed based on the basic heuristic algorithm: algorithms H2 and H3. In the algorithm H2, the initial consensus is set to the 1-Optimality consensus of collective X . The time complexity of algorithms H2 and H3 are $O(m^2n)$. During experiments, the basic heuristic algorithm is 3.8% faster than the algorithm H2 and 3.71% faster than the algorithm H3 [19]. Besides, the difference among consensus quality of the basic heuristic algorithm, that of the algorithm H2, and that of the algorithm H3 are not statistically significant [19].

Genetic algorithms were used to determine the 2-Optimality consensus for a binary vector collective. In genetic algorithm Gen1, the authors applied a roulette wheel choice, where the fitness function is $g(x) = 1/(d(x, X) + 1)$. In genetic algorithm Gen2, the authors utilized a tournament choice, where $g(x) = d(x, X)$ is the

fitness function. These two genetic algorithms were compared with the basic heuristic algorithm. During the experiment, the algorithm Gen1 is 99.47% slower than the basic heuristic algorithm, and the algorithm Gen2 is 99.56% slower than the basic heuristic algorithm. Besides, the consensus quality of the algorithm Gen1 is 1.34% lower than that of the basic heuristic algorithm, and the consensus quality of the algorithm Gen2 is 0.01% higher than that of the basic heuristic algorithm. However, both two genetic algorithms are not practical [19] because of their colossal running time.

In [31], a heuristic algorithm was based on the basic heuristic algorithm and vertical partition. First, the vertical partition was used to divide the collection into two parts. Then, the 2-Optimality consensus of these two parts were determined. Finally, the 2-Optimality consensus of the collective was determined by combining the 2-Optimality consensus of the two parts. The time complexity of this algorithm is $O(m^2n)$.

The basic heuristic algorithm is the fastest in the literature. The consensus quality of the algorithm Gen2 is higher (0.01% only) than that of the basic heuristic algorithm. However, the algorithm Gen2 is not practical because of its unacceptable running time [19]. The basic heuristic algorithm is the best algorithm for determining the consensus for binary vector collectives, and it is used to compare with the QADC algorithm.

III. BASIC NOTIONS

A. COLLECTIVE AND COLLECTIVE KNOWLEDGE

Let U be a finite set of objects representing all potential knowledge states for a given problem. In the set U , elements can contradict each other. Let $\Pi_b(U)$ denote the set of all b -element subsets with repetitions of set U for $b \in \mathbb{N}$, and let

$$\Pi(U) = \cup_b \Pi_b(U)$$

Thus, $\Pi(U)$ is the finite set of all nonempty subsets with repetitions of set U . A set $X \in \Pi(U)$ is considered a collective, where each element $x \in X$ represents the knowledge state of a collective member [3], [5].

Collective knowledge or consensus of a collective is understood as a representative of this collective. The two most popular criteria for determining consensus are 1-Optimality and 2-Optimality. For a given collective $X \in \Pi(U)$, the consensus of X is determined by the following:

- *Criterion1 – Optimality* if $d(x^*, X) = \min_{y \in U} d(y, X)$
- *Criterion2 – Optimality* if $d^2(x^*, X) = \min_{y \in U} d^2(y, X)$

where x^* is the consensus of X , $d(x^*, X)$ is the sum of distances from the consensus x^* to members of the collective X , and $d^2(x^*, X)$ is the sum of squared distances from the consensus x^* to members of the collective X [6].

Definition 1: A binary vector collective is defined as follows:

$$X = \{x_1, x_2, \dots, x_n\}$$

where n is the number of collective members, and x_i ($i = 1, 2, \dots, n$) is a binary vector of length m .

Each element $x_i \in X$ is described as follows: $x_i = (x_i^1, x_i^2, \dots, x_i^m)$, $x_i^j = \{0, 1\}$ for $j = 1, 2, \dots, m$. For $s, t \in U$, the distance function is described as follows:

$$d(s, t) = \sum_{j=1}^m |s^j - t^j| \quad (1)$$

where $s = (s^1, s^2, \dots, s^m)$, and $t = (t^1, t^2, \dots, t^m)$.

Definition 2: The sum of distances from vector x_c with a length of m to a binary vector collective X is defined as follows:

$$d(x_c, X) = \sum_{i=1}^n d(x_c, x_i) \quad (2)$$

where

$$\begin{aligned} \sum_{i=1}^n d(x_c, x_i) &= \sum_{j=1}^m |x_c^j - x_1^j| + \sum_{j=1}^m |x_c^j - x_2^j| \\ &\quad + \dots + \sum_{j=1}^m |x_c^j - x_n^j| \end{aligned}$$

Definition 3: The sum of squared distances from vector x_c with a length of m to the binary vector collective X is defined as follows:

$$\begin{aligned} d^2(x_c, X) &= \left(\sum_{j=1}^m |x_c^j - x_1^j| \right)^2 + \left(\sum_{j=1}^m |x_c^j - x_2^j| \right)^2 + \dots \\ &\quad + \left(\sum_{j=1}^m |x_c^j - x_n^j| \right)^2 \quad (3) \end{aligned}$$

B. BASIC HEURISTIC ALGORITHM

Firstly, this algorithm randomly generates one initial candidate consensus x_c . Then it determines the value of components of consensus x_c . The basic heuristic algorithm to determine the 2-Optimality consensus for a binary vector collective is presented as follows:

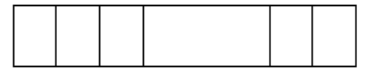
The value of $d(x_c, x_i)$ is calculated by (1) for $i = 1, 2, \dots, n$. The value of $d^2(x_c, X)$ is calculated by (3), the computational complexity of line {2} is $O(mn)$. Because line {6} repeats the calculation of $d^2(x_c, X)$ in m times, its computational complexity is $O(m^2n)$. The computational complexity of the basic heuristic algorithm is $O(m^2n)$.

IV. QUICK ALGORITHM TO DETERMINE 2-OPTIMALITY CONSENSUS

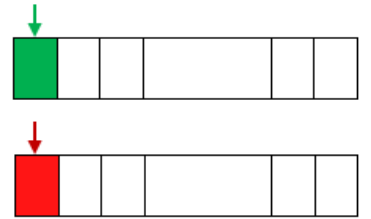
This section introduces a new method to calculate the distances from a candidate consensus to the members of a binary vector collective. Based on this method, we propose a new algorithm to determine the 2-Optimality consensus. The computational complexity of the QADC algorithm is $O(mn)$. The consensus quality of the basic heuristic algorithm and that of the QADC algorithm is the same if their initial candidate consensus are the same.

Step

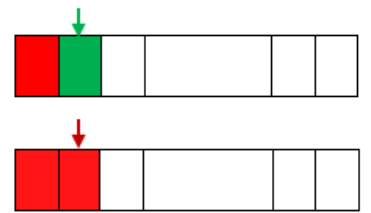
- Initialization:**
- Randomly creating vector c
 - Calculating values of $d(c, x_i)$ by (2.1) for $i = 1, \dots, n$
 - Calculating $d^2(c, X)$



- Step 1:**
- Changing value of the 1st component of c
 - Using Theorem 2 and values of $d(c, x_i)$ in step Initialization to calculate values of $d(c, x_i)$ for $i = 1, \dots, n$
 - Calculating $d^2(c, X)$
 - Choosing value of the 1st component correspond to smaller value of $d^2(c, X)$ in step Initialization and that in this step



- Step 2:**
- Changing value of the 2nd component of c
 - Using Theorem 2 and values of $d(c, x_i)$ in step 1 to calculate values of $d(c, x_i)$ for $i = 1, \dots, n$
 - Calculating $d^2(c, X)$
 - Choosing value of the 2nd component correspond to smaller value of $d^2(c, X)$ in step 1 and that in this step



- Step m:**
- Changing value of the m^{th} component of c
 - Using Theorem 2 and values of $d(c, x_i)$ in step $m-1$ to calculating values of $d(c, x_i)$ for $i = 1, \dots, n$
 - Calculating $d^2(c, X)$
 - Choosing value of the m^{th} component correspond to smaller value of $d^2(c, X)$ in step $m-1$ and that in this step

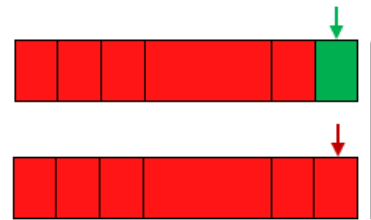


FIGURE 1. The QADC algorithm.

A. METHOD FOR CALCULATING DISTANCES

The sum of squared distances from a candidate consensus x_c to collective members is computed by (3). The computational complexity of calculating value $d^2(x_c, X)$ depends on the computational complexity of calculating values $d(x_c, x_i)$. The distance between x_c and x_i for $i = 1, 2, \dots, n$ is computed by (1). The computational complexity of calculating value $d^2(x_c, X)$ can be reduced if the computational complexity of calculating values $d(x_c, x_i)$ reduces. Thus, an efficient method for calculating values $d(x_c, x_i)$ need to be studied.

Theorem 1: Let $x_v = (x_v^1, x_v^2, \dots, x_v^m)$, $x_u = (x_u^1, x_u^2, \dots, x_u^m)$, and $x_w = (x_w^1, x_w^2, \dots, x_w^m)$ be binary vectors of the same length. Assume that if x_u and x_w only differ in the k^{th} component, then

$$d(x_v, x_u) = \begin{cases} d(x_v, x_w) - 1 & \text{if } x_v^k = x_u^k \\ d(x_v, x_w) + 1 & \text{if } x_v^k \neq x_u^k \end{cases}$$

Example 1: Given $x_v = (1, 1, 1, 0, 1, 1)$, $x_w = (0, 1, 0, 1, 1, 0)$. We have $d(x_v, x_w) = 4$.

If $x_u = (0, 0, 0, 1, 1, 0)$, x_u and x_w differ in the 2nd component: $x_w^2 \neq x_u^2$. Because $x_v^2 \neq x_u^2$, then $d(x_v, x_u) = d(x_v, x_w) + 1 = 5$. If $x_u = (0, 1, 1, 1, 1, 0)$, x_u and x_w only differ in the 3rd component: $x_w^3 \neq x_u^3$. Because $x_v^3 = x_u^3$, then $d(x_v, x_u) = d(x_v, x_w) - 1 = 3$. ■

Theorem 2: Given a binary vector collective $X = \{x_1, x_2, \dots, x_n\}$, the length of each member is m . Assume that x_u and x_w are binary vectors of length m and differ in the k^{th} component. If $d(x_i, x_w)$ is determined, then $d(x_i, x_u)$ can be determined for $i = 1, 2, \dots, n$.

Example 2: Given $x_w = (0, 1, 0, 1)$, $x_u = (1, 1, 0, 1)$ and $X = \{x_1, x_2, x_3\}$, where $x_1 = (1, 1, 0, 1)$, $x_2 = (0, 0, 1, 0)$, $x_3 = (1, 0, 1, 1)$.

We have $d(x_1, x_w) = 1$, $d(x_2, x_w) = 3$, $d(x_3, x_w) = 3$. As x_u and x_w only differ in the 1st component ($k = 1$), based on Theorem 2, we have:

$$\begin{aligned} d(x_1, x_u) &= d(x_1, x_w) - 1 = 0 \text{ because } x_1^1 = x_u^1. \\ d(x_2, x_u) &= d(x_2, x_w) + 1 = 4 \text{ because } x_2^1 \neq x_u^1. \\ d(x_3, x_u) &= d(x_3, x_w) - 1 = 2 \text{ because } x_3^1 = x_u^1. \end{aligned}$$

Algorithm 1 Basic Heuristic Algorithm

```

Input: Collective  $X = \{x_1, x_2, \dots, x_n\}$ 
Output: 2-Optimality consensus  $x_c = (x_c^1, x_c^2, \dots, x_c^m)$ 
Begin
// Initialization
1. randomly choosing binary vector  $x_c$ 
2. calculate  $tO_2 = d^2(x_c, X)$ 
// Choice
3. for  $k = 1$  to  $m$  do
4.    $\oplus x_c^k = x_c^k 1$ ;
5.   calculate  $d^2(x_c, X)$ ;
6.   if  $d^2(x_c, X) < tO_2$  then
7.     else  $tO_2 = d^2(x_c, X)$ ;
8.   end  $\oplus x_c^k = x_c^k 1$ ;
9. end
12. END

```

B. QADC ALGORITHM

In the proposed algorithm, components of a candidate consensus x_c are sequentially determined from 1st to m^{th} . Firstly, x_c is randomly created, and values of $d(x_c, x_i)$ for $i = 1, 2, \dots, n$ are computed by (1). In the next steps, the distances between x_c and collective members are computed based on Theorem 2 and the results of the previously adjacent step.

The QADC algorithm is described in Fig. 1. In this figure, the white boxes represent the values of components randomly created at step Initialization, the green box in each step presents a component that its value is changed, the red box in each step represents that the value of this component is determined.

Example 3: Given $x_c = (0, 1, 0, 1)$ and $X = \{x_1, x_2, x_3\}$, where $x_1 = (1, 1, 0, 1)$, $x_2 = (0, 0, 1, 0)$, $x_3 = (1, 0, 1, 1)$. The basic heuristic algorithm determines the 2-Optimality consensus for the collective X as the following:

Step Initialization:

Assume that $x_c = (0, 1, 0, 1)$. We have

$$\begin{aligned}
 d(x_c, x_1) &= \sum_{j=1}^4 |x_c^j - x_1^j| = 1 \\
 d(x_c, x_2) &= \sum_{j=1}^4 |x_c^j - x_2^j| = 3 \\
 d(x_c, x_3) &= \sum_{j=1}^4 |x_c^j - x_3^j| = 3 \\
 d^2(x_c, X) &= \sum_{i=1}^3 d^2(x_c, x_i) = 19
 \end{aligned}$$

Step 1 (k = 1) :

Changing the value of 1st component, $x_c = (1, 1, 0, 1)$.

We have

$$\begin{aligned}
 d(x_c, x_1) &= \sum_{j=1}^4 |x_c^j - x_1^j| = 0 \\
 d(x_c, x_2) &= \sum_{j=1}^4 |x_c^j - x_2^j| = 4 \\
 d(x_c, x_3) &= \sum_{j=1}^4 |x_c^j - x_3^j| = 2
 \end{aligned}$$

$$d^2(x_c, X) = \sum_{i=1}^3 d^2(x_c, x_i) = 20.$$

Value of $d^2(x_c, X)$ with $x_c = (1, 1, 0, 1)$ is larger than that with $x_c = (0, 1, 0, 1)$. Thus, $x_c = (0, 1, 0, 1)$ is chosen.

Step 2 (k = 2) :

Changing the value of 2nd component, $x_c = (0, 0, 0, 1)$. We have

$$\begin{aligned}
 d(x_c, x_1) &= \sum_{j=1}^4 |x_c^j - x_1^j| = 2 \\
 d(x_c, x_2) &= \sum_{j=1}^4 |x_c^j - x_2^j| = 2 \\
 d(x_c, x_3) &= \sum_{j=1}^4 |x_c^j - x_3^j| = 2 \\
 d^2(x_c, X) &= \sum_{i=1}^3 d^2(x_c, x_i) = 12.
 \end{aligned}$$

Value of $d^2(x_c, X)$ with $x_c = (0, 0, 0, 1)$ is smaller than that with $x_c = (0, 1, 0, 1)$. Therefore, $x_c = (0, 0, 0, 1)$ is chosen.

Step 3 (k = 3) :

Changing the value of 3rd component, $x_c = (0, 0, 1, 1)$. We have

$$\begin{aligned}
 d(x_c, x_1) &= \sum_{j=1}^4 |x_c^j - x_1^j| = 3 \\
 d(x_c, x_2) &= \sum_{j=1}^4 |x_c^j - x_2^j| = 1 \\
 d(x_c, x_3) &= \sum_{j=1}^4 |x_c^j - x_3^j| = 1 \\
 d^2(x_c, X) &= \sum_{i=1}^3 d^2(x_c, x_i) = 11.
 \end{aligned}$$

Value of $d^2(x_c, X)$ with $x_c = (0, 0, 1, 1)$ is smaller than that with $x_c = (0, 0, 0, 1)$. Thus, $x_c = (0, 0, 1, 1)$ is chosen.

Step 4 (k = 4) :

Changing the value of 4th component, $x_c = (0, 0, 1, 0)$. We have

$$\begin{aligned}
 d(x_c, x_1) &= \sum_{j=1}^4 |x_c^j - x_1^j| = 4 \\
 d(x_c, x_2) &= \sum_{j=1}^4 |x_c^j - x_2^j| = 0 \\
 d(x_c, x_3) &= \sum_{j=1}^4 |x_c^j - x_3^j| = 2 \\
 d^2(x_c, X) &= \sum_{i=1}^3 d^2(x_c, x_i) = 20
 \end{aligned}$$

Value of $d^2(x_c, X)$ with $x_c = (0, 0, 1, 0)$ is larger than that with $x_c = (0, 0, 1, 1)$. Thus, $x_c = (0, 0, 1, 1)$ is chosen. It is also the 2-Optimality consensus of the collective X generated by the proposed algorithm. ■

The schema of the QADC is described in Fig. 2. In this algorithm, tO_2 is the sum of squared distances from a current candidate consensus x_c to collective members, and nO_2 is the sum of squared distances from a new candidate consensus to collective members. In each step, tO_2 and nO_2 are computed. If nO_2 is larger than or equal to tO_2 , the new candidate consensus is discarded. Otherwise, the new candidate consensus is considered as the current candidate consensus.

After step m , the current consensus is the consensus of the collective X .

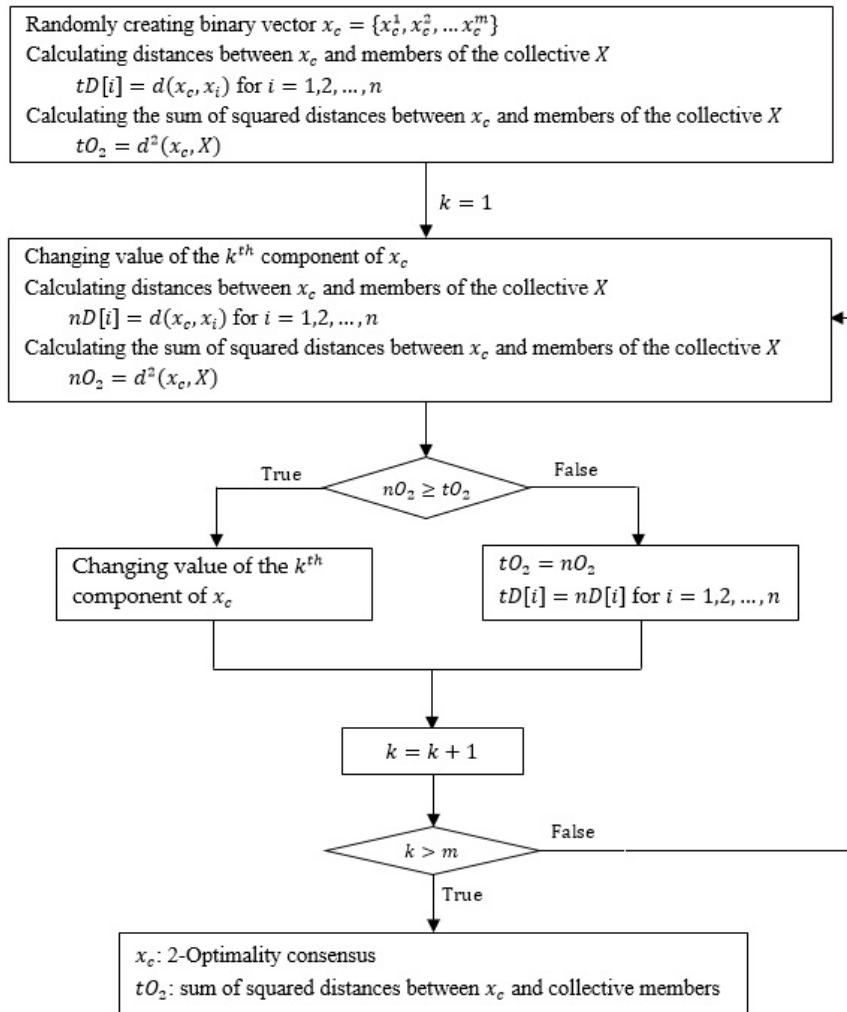


FIGURE 2. Schema of the QADC algorithm.

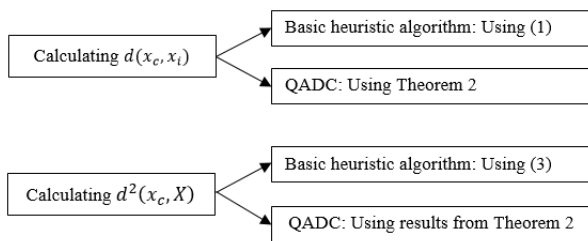


FIGURE 3. Calculation of $d(x_c, x_i)$ and $d^2(x_c, X)$ in Algorithm 1 and 2.

The QADC algorithm is presented as follows:

The main difference in the calculation of each step (from step 1 to step m) between the basic heuristic algorithm and the QADC algorithm is shown in Fig. 3.

Theorem 3: The computational complexity of the QADC algorithm is $O(mn)$.

Theorem 4: The 2-Optimality consensus determined by the basic heuristic algorithm and that by the QADC algorithm is the same if initial candidate consensuses are the same.

V. EXPERIMENT AND ANALYSIS

This section assesses the efficiency of the QADC algorithm. We compare the QADC algorithm and the basic heuristic algorithm. The basic heuristic algorithm is chosen because it is the best algorithm for previous studies, and it is being used widely. These two algorithms are compared both in two respects: running time and consensus quality. The main tasks in this section are described as follows:

- Measuring the consensus quality of the QADC algorithm,
- Comparing the consensus quality of the QADC algorithm and that of the basic heuristic algorithm,
- Comparing the running time of the QADC algorithm and the running time of the basic heuristic algorithm.

The consensus quality of a heuristic algorithm is estimated as follows:

$$QC = 1 - \frac{|d^2(x^*, X) - d^2(x_{opt}, X)|}{d^2(x_{opt}, X)} \quad (4)$$

where x^* is the consensus determined by the heuristic algorithm, and x_{opt} is the optimal consensus determined by the brute-force algorithm.

Algorithm 2 QADC Algorithm

Input: Collective $X = \{x_1, x_2, \dots, x_n\}$
Output: Consensus $x_c = (x_c^1, x_c^2, \dots, x_c^m)$
Begin

```

// Initialization
1. randomly generating an initial candidate
   consensus  $x_c$ ;
2.  $tO_2 = 0$ ;
3. for  $i = 1$  to  $n$  do
4.    $tD[i] = 0$ ;
5.   for  $j = 1$  to  $m$  do
6.      $tD[i] = tD[i] + |x_i^j - x_c^j|$ ;
7.   end
8.    $tO_2 = tO_2 + (tD[j])^2$ ;
9. end
// Choice
10. for  $k = 1$  to  $m$  do
11.    $\oplus x_c^k = x_c^k 1$ ;
12.    $nO_2 = 0$ ;
13.   for  $i = 1$  to  $n$  do
14.     if  $x_i^k = x_c^k$  then
15.        $nD[i] = tD[i] - 1$ ;
16.     else
17.        $nD[i] = tD[i] + 1$ ;
18.     end
19.      $nO_2 = nO_2 + (nD[j])^2$ ;
20.   end
21.   if  $nO_2 \geq tO_2$  then
22.      $\oplus x_c^k = x_c^k 1$ ;
23.   else
24.      $tO_2 = nO_2$ ;
25.     for  $i = 1$  to  $n$  do
26.        $tD[i] = nD[i]$ ;
27.     end
28.   end
29. end
END

```

In this study, the significance level is chosen to be 0.05 ($\alpha = 0.05$).

A. MEASURING THE CONSENSUS QUALITY

We have used the hepatocellular carcinoma dataset¹ in this study. This dataset is heterogeneous and hence comprised of 205 real patients diagnosed with hepatocellular carcinoma. Each patient has 23 binary attributes, and each attribute represents a specific disease. We consider that each patient is represented as a binary vector with a length of 23. If a patient has a specific disease, the value of corresponding component is 1, otherwise 0.

¹ <https://www.kaggle.com/mrsantos/hcc-dataset>

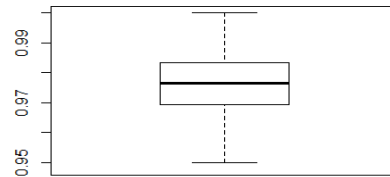


FIGURE 4. Consensus quality of the QADC algorithm.

If we perform the QADC algorithm on this dataset one time, one consensus is generated, and its quality is computed by (4). In the experiment, a set of consensus qualities create a sample, and this sample is used to measure the consensus quality of the QADC algorithm. We need to determine how many times to perform the QADC algorithm. The number of times performed by the QADC algorithm on this dataset is also the size (sz) of the sample.

We need to compute the consensus quality of the QADC algorithm with a margin of error (E) of 0.002 and a confidence interval of 95%. The sample deviation of the pilot survey (s) was 0.02. The sample size is determined below [64]:

$$sz \geq \left(\frac{Z \times s}{E} \right)^2$$

The confidence interval was 95%, then $Z = 1.96$. We have

$$sz \geq \left(\frac{Z \times s}{E} \right)^2 = \left(\frac{1.96 * 0.02}{0.002} \right)^2 = 484.16$$

We can choose $sz = 485$.

We perform the QADC algorithm on the hepatocellular carcinoma dataset 485 times. After every run of the QADC algorithm, a consensus is generated. The quality of this consensus is determined by (4). Finally, we obtain a consensus quality sample with a size of 485. This sample is shown in Table 1. The mean of the sample is 0.976, so the consensus quality of the QADC algorithm is 0.976 ± 0.001 .

The boxplot of the sample consensus is shown in Fig. 4. The maximum consensus quality is 1.000, and the minimum consensus quality is 0.951. The mean of the sample is 0.976.

B. EVALUATING THE CONSENSUS QUALITY

In this section, we continue our experiment by comparing the consensus qualities of the QADC and the basic heuristic algorithm. The sample consensus of the QADC algorithm has been presented in Table 1.

The sample consensus in basic heuristic algorithm is computed as follows. We perform the basic heuristic algorithm to find the consensus for the hepatocellular carcinoma dataset. This algorithm is performed on the hepatocellular carcinoma dataset 485 times. After each run of the algorithm, a consensus is generated. The quality of this consensus is computed by (4). Finally, we obtain a consensus quality sample of the basic heuristic algorithm with a size of 485, and this sample is shown in Table 2.

We utilize the Shapiro-Wilk test to find the distribution of the two consensus quality samples. The significance level is chosen to be 0.05. The p-values of two samples are less

TABLE 1. Consensus quality of the QADC algorithm.

0.9660630	0.9830417	0.9809866	0.9672474	0.9763153	0.9788934	0.9711732	0.9571367	0.9615797	0.9646080	0.9641471	0.9775616
0.9862679	0.9640264	0.9656104	0.9854117	0.9780556	0.9669578	0.9997473	1.0000000	0.9865432	0.9774362	0.9770092	0.9783363
0.9835106	0.9846637	0.9737378	0.9777059	0.9744465	0.9782632	0.9797464	0.9799439	0.9700267	0.9650045	0.9582002	0.9795559
0.9609083	0.9846415	0.9798143	0.9653650	0.9816829	0.9834808	0.9913843	0.9668643	0.9838946	0.9850352	0.9818984	0.9597814
0.9750485	0.9640138	0.9681648	0.9621195	0.9861068	0.9866110	0.9749029	0.9848675	0.9911649	0.9908378	0.9869661	0.9782071
0.9670405	0.9823949	0.9818299	0.9968672	0.9708202	0.9738715	0.9811147	0.9793337	0.9740823	0.9955072	0.9904566	0.9791045
0.9552925	1.0000000	0.9628349	0.9692150	0.9730590	0.9750636	0.9781396	0.9742936	0.9788388	0.9839760	0.9567485	0.9617889
0.9775012	0.9704278	0.9479023	0.9574443	0.9799784	0.9751329	0.9741388	0.9841883	0.9584893	0.9944327	0.9801277	0.9855537
0.9752079	0.9844490	0.9806497	0.9813326	0.9704978	0.9904146	0.9747961	0.9798837	0.9678133	0.9884642	0.9919337	0.9838417
0.9750263	0.9681780	0.9844054	0.9791023	0.9769127	0.9872507	0.9861283	0.9715406	0.9765621	0.9746813	0.9718598	0.9989962
0.9781615	0.9871071	0.9731415	0.9624617	0.9563829	0.9843440	0.9739854	0.9783111	0.9789909	0.9807704	0.9738785	0.9775670
0.9848247	0.9784982	0.9810413	0.9565704	0.9648010	0.9731266	0.9556232	0.9824751	0.9684060	0.9662801	0.9756346	0.9764673
0.9780560	0.9925192	0.9644408	0.9748370	0.9627224	0.9797324	0.9740411	0.9795628	1.0000000	0.9741480	0.9796502	0.9769659
0.9698356	0.9614103	0.9747285	0.9873940	0.9674638	0.9800329	0.9813979	0.9694199	0.9714161	0.9882096	0.9826516	0.9766977
1.0000000	0.9754870	0.9565848	0.9823612	0.9690670	0.9655833	0.9821646	0.9845520	0.9766424	0.9814128	0.9891782	0.9575153
0.9746820	0.9707307	0.9878118	0.9710706	0.9798231	0.9587170	0.9821657	0.9875294	0.9824979	1.0000000	0.9750451	0.9592887
0.9791002	0.9740274	0.9945991	0.9676581	0.9858211	0.9824183	0.9590790	0.9787627	0.9757398	0.9656187	0.9779628	0.9752246
0.9656032	0.9697042	0.9867401	0.9787107	0.9687262	0.9607069	0.9796874	0.9774410	0.9695643	0.9699549	1.0000000	0.9701893
0.9741569	0.9676616	0.9757265	0.9775735	0.9660316	0.9760168	0.9856786	0.9752437	0.9864531	0.9683539	0.9803110	0.9765474
0.9856727	0.9817872	0.9756667	0.9822971	0.9655831	0.9785025	0.9887658	0.9976142	0.9921555	0.9799530	0.9778875	0.9548879
0.9749172	0.9651242	0.9608393	0.9720420	0.9718541	0.9816387	0.9737504	0.9787632	0.9757031	0.9660949	0.9625776	0.9610130
0.9690158	0.9908403	0.9839039	0.9849935	0.9736097	0.9778943	0.9727811	0.9744171	0.9816227	0.9816204	0.9760286	0.9649852
0.9732405	0.9641379	0.9738927	0.9676919	0.9808362	0.9686715	0.9908784	0.9509208	0.9750259	0.9648358	0.9737867	0.9858606
0.9871465	0.9770108	0.9694326	0.9726946	0.9727868	0.9858637	0.9593207	0.9603472	0.9861646	0.9942853	0.9758895	0.9650151
0.9815004	0.9813299	0.9618797	0.9834081	0.9552151	0.9933863	0.9716317	0.9752233	0.9644383	0.9806059	0.9702458	0.9680049
0.9883668	0.9818674	0.9730024	0.9858997	0.9750857	0.9848118	0.9805746	0.9780629	0.9992086	0.9689900	0.9691318	0.9767987
0.9773910	0.9729825	0.9675094	0.9566150	0.9878719	0.9565635	0.9598223	0.9787687	0.9699647	0.9784105	0.9687923	0.9727725
0.9801028	0.9767950	0.9720297	0.9770719	0.9879160	0.9899958	0.9787963	1.0000000	1.0000000	0.9724755	0.9738550	0.9774642
0.9704154	0.9856126	0.9638240	1.0000000	0.9681104	0.9754394	0.9947786	0.9740585	0.9724908	0.9797115	0.9896813	0.9990506
0.9820537	0.9614353	0.9928759	0.9614918	0.9605222	0.9812491	0.9759594	0.9881459	0.9774671	0.9784353	0.9864909	0.9647540
0.9709367	0.9681826	0.9758400	0.9808063	1.0000000	1.0000000	0.9732155	0.9667848	0.9922362	1.0000000	1.0000000	0.9729453
0.9617943	0.9792040	0.9867495	0.9677183	0.9743758	0.9750331	0.9807491	0.9639156	0.9851121	0.9680869	0.9591408	0.9708324
0.9772799	0.9715522	0.9499830	1.0000000	0.9750255	0.9752474	0.9732093	0.9637101	0.9774246	0.9789959	0.9675542	0.9911240
0.9954585	0.9897000	0.9714680	0.9813848	0.9802057	0.9861916	0.9841340	0.9834230	0.9621052	0.9923905	0.9714224	0.9683052
0.9840091	0.9827325	0.9692452	0.9836905	0.9598596	0.9831160	0.9850444	0.9751708	0.9673396	0.9868462	0.9770364	
0.9876525	0.9767217	0.9637707	0.9772072	0.9687178	0.9859026	0.9760269	0.9838982	0.9743672	0.9697543	0.9693739	
0.9795886	0.9609224	0.9914661	0.9846365	0.9605956	0.9998293	0.9642331	0.9733229	1.0000000	0.9842592	0.9960668	
0.9699144	0.9762610	0.9618472	0.9898051	0.9690691	0.9826442	0.9628178	0.9700811	0.9573977	0.9755143	0.9732773	
0.9739776	0.9728358	0.9791839	0.9956625	0.9771885	0.9780738	0.9700700	0.9723165	0.9873105	0.9790131	0.9638606	
0.9732675	0.9749765	1.0000000	0.9757160	0.9623529	0.9538937	0.9839738	0.9574738	0.9707277	0.9786036	0.9745874	
0.9713130	0.9641844	0.9777819	0.9535095	0.9818998	1.0000000	0.9564179	0.9643038	0.9926599	1.0000000	0.9659462	

than 0.05 ($p - value = 0.0015$ and $p - value = 0.0381$ for the sample consensus of the QADC algorithm and the sample consensus of the basic heuristic algorithm, respectively). Thus, there is evidence that these two samples do not come from a normal distribution.

The hypotheses to compare the consensus quality of these two algorithms are presented as follows:

- *Hypothesis H_0* : The difference in consensus quality between the QADC algorithm and the basic heuristic algorithm is not significant.
- *Hypothesis H_1* : The difference in consensus quality between the QADC algorithm and the basic heuristic algorithm is significant.

Since the two samples do not come from a normal distribution, the Wilcoxon rank-sum is utilized for this test.

We obtain $p - value = 0.0952$. The $p - value$ is greater than 0.05; thus, the hypothesis H_0 cannot be rejected. It means that the difference in consensus quality between the QADC algorithm and the basic heuristic algorithm is not statistically significant.

In Theorem 4, we prove that if initial candidate consensus x_c are the same, then the 2-Optimality consensus determined by the QADC algorithm and that by the basic heuristic algorithm are the same. Besides, the analysis of experimental results shows that the difference between the consensus quality determined by the QADC algorithm and that the basic heuristic algorithm is not statistically significant.

Similar to the basic heuristic algorithm, the QADC algorithm keeps track of one initial candidate consensus to determine the consensus of a collective. Keeping track of one

TABLE 2. Consensus quality of the basic heuristic algorithm.

0.9729274	0.9684165	0.9825651	0.9741981	0.9713020	0.9848223	0.9754268	0.9731926	0.9804861	0.9759250	0.9811398	0.9797696
0.9669790	0.9656411	0.9714600	0.9901024	0.9734959	0.9746663	0.9885675	0.9833410	0.9656636	0.9907431	1.0000000	0.9716828
0.9822707	0.9854816	0.9700614	0.9824347	0.9946866	0.9647932	0.9918745	0.9721336	0.9747480	0.9815414	0.9660146	0.9955906
0.9872036	0.9851416	0.9772034	0.9677874	0.9939042	0.9806119	0.9791948	1.0000000	0.9852866	0.9780366	0.9902082	0.9617154
0.9972721	0.9630127	0.9739055	0.9605408	0.9649892	0.9912414	0.9798159	0.9793954	0.9846834	0.9931904	1.0000000	0.9961129
0.9796611	0.9802438	1.0000000	0.9677345	0.9743189	0.9803447	0.9735636	0.9604092	0.9653200	0.9868296	0.9656022	0.9724841
0.9672522	0.9648745	0.9858970	0.9763453	0.9897528	0.9779200	0.9764805	0.9772167	0.9916581	0.9724915	0.9863103	0.9895712
0.9862447	0.9654893	0.9927557	0.9848807	0.9859829	0.9694376	0.9619545	0.9738958	0.9656519	0.9619651	0.9750585	0.9560082
0.9850476	0.9812541	0.9851497	0.9772034	0.9887660	0.9816840	0.9788993	0.9804994	0.9776452	0.9739820	0.9774181	0.9855608
0.9736175	0.9691398	0.9874426	0.9862320	0.9652825	0.9652943	0.9706445	0.9656884	0.9704435	0.9747322	0.9882224	0.9847643
0.9604215	0.9859348	0.9850288	0.9760546	1.0000000	0.9708840	0.9793468	0.9635508	0.9746138	0.9865921	0.9781367	0.9632878
1.0000000	0.9763852	1.0000000	0.9816978	0.9646654	0.9755665	0.9725402	1.0000000	0.9948284	0.9643260	0.9674863	0.9683168
0.9872914	0.9813615	0.9701747	0.9594674	1.0000000	0.9756541	0.9693838	0.9892877	0.9847367	1.0000000	0.9712959	0.9779352
0.9730489	0.9707637	0.9706707	0.9693335	0.9774128	0.9996506	0.9738089	0.9789268	0.9584870	0.9908812	0.9828614	0.9874384
0.9813624	0.9644878	0.9599916	0.9715177	0.9719629	0.9638365	0.9723310	0.9707306	0.9634858	0.9895867	0.9785516	0.9683400
0.9732411	0.9851475	0.9759418	0.9864389	0.9722058	0.9777091	0.9869458	0.9730142	0.9936455	0.9876321	0.9869524	1.0000000
0.9828232	0.9783807	0.9849936	0.9862817	0.9660860	0.9840505	0.9780921	0.9768724	0.9716610	0.9926152	0.9603285	0.9917135
0.9748271	0.9736093	0.9863192	0.9803509	0.9922266	0.9865059	0.9803249	0.9761096	0.9712438	0.9780403	0.9762193	0.9647265
0.9725532	0.9766924	0.9769513	0.9920421	0.9768951	0.9758928	0.9635915	0.9571857	0.9921658	0.9701812	0.9740964	0.9854779
0.9771162	0.9892591	0.9705237	0.9708459	0.9785922	0.9685674	0.9909682	0.9642430	0.9905013	0.9815520	0.9889307	0.9804877
0.9731659	0.9690183	1.0000000	0.9861254	0.9780963	0.9753422	0.9775937	0.9855256	0.9870902	0.9865872	0.9778885	0.9649419
0.9700898	0.9685059	0.9833669	0.9756406	0.9722483	0.9953976	0.9674372	0.9730943	0.9772641	1.0000000	0.9770194	0.9693213
0.9728406	0.9698038	0.9902058	0.9693266	0.9646598	0.9808274	0.9790905	0.9727149	0.9714819	0.9713951	0.9829814	0.9838328
0.9759185	0.9601501	0.9726232	0.9852338	0.9785373	0.9555552	0.9847043	0.9957270	0.9646337	0.9787324	0.9677299	0.9784896
0.9780750	0.9841963	0.9756204	0.9898110	0.9550636	0.9902346	0.9621632	0.9535131	0.9780979	0.9825479	0.9740410	0.9902510
0.9913242	0.9779237	0.9804861	0.9847825	0.9618143	0.9814050	0.9929011	0.9843822	0.9787916	0.9754540	0.9877758	0.9699822
0.9624200	0.9780717	0.9927652	0.9709060	0.9652360	0.9756643	0.9744197	0.9753808	0.9752383	0.9604218	0.9828347	0.9588551
0.9740038	0.9755665	0.9728853	0.9713021	0.9652132	0.9758214	0.9872117	0.9821392	0.9899466	0.9834150	0.9632450	0.9864783
0.9823152	0.9708984	0.9845362	0.9897768	0.9770719	0.9745021	0.9767226	0.9605391	0.9776453	0.9682091	0.9823796	0.9699138
0.9936202	0.9677658	0.9550519	0.9795283	0.9919849	0.9785656	0.9726758	0.9748761	0.9766924	0.9810586	0.9622241	0.9757821
0.9802601	0.9845186	0.9709275	0.9842957	0.9608468	0.9709613	0.9576508	0.9917785	0.9891432	0.9850755	1.0000000	0.9731098
0.9758625	0.9617382	0.9630799	0.9726130	0.9803368	0.9787701	0.9649983	0.9772997	0.9874626	0.9888396	0.9881092	0.9543739
0.9729244	0.9804030	0.9871336	0.9886103	0.9849954	0.9666864	0.9755866	0.9702910	0.9735742	0.9604214	0.9713951	0.9740744
0.9801431	0.9680739	0.9743555	0.9679124	0.9661047	0.9780015	0.9842785	0.9810570	0.9835954	0.9868174	0.9582099	0.9689849
0.9858906	0.9788231	0.9720963	0.9822535	0.9754720	1.0000000	0.9571832	0.9668540	0.9678521	1.0000000	0.9890137	
0.9741614	0.9685931	0.9896910	0.9678283	0.9842361	0.9810921	0.9897544	0.9775183	0.9884371	0.9631098	0.9807280	
0.9776376	0.9707666	0.9871627	0.9513742	0.9734449	0.9863375	0.9899899	0.9675083	0.9743175	0.9714423	0.9725925	
0.9781694	0.9936937	0.9670198	0.9625704	0.9737932	0.9650913	0.9645668	0.9673967	0.9906981	0.9756734	0.9735821	
0.9832928	0.9826828	0.9802787	0.9773630	0.9790773	0.9765479	0.9807230	0.9841328	0.9987347	0.9923676	0.9897875	
0.9871138	0.9545510	0.9637156	0.9848292	0.9753999	0.9821725	0.9744897	0.9804861	0.9714497	0.9727096	0.9726112	
0.9802778	0.9631064	0.9497067	0.9772615	0.9804861	0.9694543	0.9750628	1.0000000	0.9836110	0.9931904	0.9762014	

initial candidate consensus results in the moderate consensus quality of the QADC algorithm. That is the main limitation of the QADC algorithm. One approach to solve this limitation keeps track of many initial candidate consensuses. Besides, the diversity of initial candidate consensuses is carefully considered.

C. EVALUATING THE RUNNING TIME

The basic heuristic algorithm is the fastest in the literature. Therefore, we compare the running time of the QADC algorithm and that of the basic heuristic algorithm. Two datasets are generated randomly.

- Dataset 1 consists of 14 binary vector collectives. The vector length is 23. The sizes of collectives are 15,000; 20,000; 25,000; 30,000; 35,000; 40,000; 45,000; 50,000; 55,000; 60,000; 65,000; 70,000; 75,000; 80,000.

- Dataset 2 contains 14 binary vector collectives. The vector length is 30. The sizes of collectives are 14,000; 19,000; 24,000; 29,000; 34,000; 39,000; 44,000; 49,000; 54,000; 59,000; 64,000; 69,000; 74,000; 79,000.

First, these two algorithms are performed on dataset 1. We obtain two samples: one sample running time of the QADC algorithm and one sample running time of the basic heuristic algorithm. Table 3 shows the two samples and the ratio of the basic heuristic algorithm’s running time to the QADC algorithm’s running time.

We utilize the Shapiro-Wilk test to find the distribution of the two samples. The p-values of the two samples are larger than 0.05 ($p - value = 0.8881$ and $p - value = 0.7852$ for the sample running time of the QADC algorithm and the sample running time of the basic heuristic algorithm, respectively). It means that these samples come from a normal distribution.

TABLE 3. Running time comparison on dataset 1 (in seconds).

Collective size	Basic heuristic algorithm	QADC algorithm	Speed-up ratio
15,000	0.110	0.0192	5.7292
20,000	0.147	0.0258	5.6977
25,000	0.185	0.0321	5.7632
30,000	0.235	0.0384	6.1198
35,000	0.258	0.0438	5.8904
40,000	0.292	0.0552	5.2899
45,000	0.338	0.0576	5.8681
50,000	0.369	0.0630	5.8571
55,000	0.406	0.0669	6.0688
60,000	0.446	0.0723	6.1687
65,000	0.494	0.0837	5.9020
70,000	0.518	0.0865	5.9884
75,000	0.552	0.0895	6.1676
80,000	0.593	0.1026	5.7797

TABLE 4. Running time comparison on dataset 2 (in seconds).

Collective size	Basic heuristic algorithm	QADC algorithm	Speed-up ratio
14,000	0.1890	0.0261	7.2344
19,000	0.2483	0.0345	7.2031
24,000	0.3125	0.0467	6.6855
29,000	0.3977	0.0558	7.1254
34,000	0.4342	0.0610	7.1147
39,000	0.4918	0.0693	7.1011
44,000	0.5715	0.0806	7.0868
49,000	0.6203	0.0889	6.9764
54,000	0.6862	0.0977	7.0215
59,000	0.7502	0.0982	7.6417
64,000	0.8315	0.1108	7.5055
69,000	0.8807	0.1218	7.2299
74,000	0.9334	0.1353	6.8966
79,000	1.0041	0.1400	7.1707

The hypotheses to compare the running time of the two algorithms are stated as follows:

- Hypothesis H_0 : The difference in running time between the QADC algorithm and the basic heuristic algorithm is not significant.
- Hypothesis H_1 : The difference in running time between the QADC algorithm and the basic heuristic algorithm is significant.

Paired samples come from a normal distribution; therefore, we use the paired t-test for this test. We obtain $p - value = 0.000007$. Since the $p - value$ is less than 0.05, the hypothesis H_0 is rejected. It means that the difference in running time between the QADC algorithm and the basic heuristic algorithm is significant. Thus, their means are compared.

The mean of the sample running time of the QADC algorithm and that of the basic heuristic algorithm are 0.059757 and 0.353071, respectively. The average running time for the QADC algorithm occupies 16.92% ($\frac{0.059757}{0.353071} \times 100\%$) that of the basic heuristic algorithm. In simple words, the running

time of the QADC algorithm equals to 16.92% that of the basic heuristic algorithm.

Second, we perform the QADC algorithm and the heuristic algorithm on dataset 2. We obtain a sample running time of the QADC algorithm and a sample running time of the basic heuristic algorithm. Table 4 shows these two samples and the ratio of the running time of the basic heuristic algorithm to that of the QADC algorithm.

The Shapiro-Wilk test is utilized to find the distribution of the two samples. The p-values of the two samples are larger than 0.05 ($p - value = 0.9142$ and $p - value = 0.7804$ for the sample running time of the QADC algorithm and that of the basic heuristic algorithm, respectively). It means that these samples come from a normal distribution.

The hypotheses for comparison of the running time of the two algorithms are stated as follows:

- Hypothesis H_0 : The difference in running time between the QADC algorithm and the basic heuristic algorithm is not significant.
- Hypothesis H_1 : The difference in running time between the QADC algorithm and the basic heuristic algorithm is significant.

The paired t-test is used for this test. We obtain $p - value = 0.000005$. Because the $p - value$ is less than 0.05, the hypothesis H_0 is rejected. The difference in running time between the QADC algorithm and the basic heuristic algorithm is significant. Their means are compared.

The mean of the sample running time of the QADC algorithm and that of the basic heuristic algorithm are 0.083336 and 0.596529, respectively. The average running time for the QADC algorithm occupies 13.97% ($\frac{0.083336}{0.596529} \times 100\%$) that of the basic heuristic algorithm. In other words, the running time of the QADC algorithm equals to 13.97% that of the basic heuristic algorithm.

The computational complexity of the basic heuristic algorithm is $O(m^2n)$, and the computational complexity of the QADC algorithm is $O(mn)$. In the case of dataset 1, the length of the vector is 23, so the computational complexity of the basic heuristic algorithm and that of the QADC algorithm are $O(23^2n)$ and $O(23n)$, respectively. The running time of the QADC algorithm equals to 16.9% that of the basic heuristic algorithm. In the case of dataset 2, the length of the vector is 30, so the computational complexity of the basic heuristic algorithm and that of the QADC algorithm are $O(30^2n)$ and $O(30n)$, respectively. The running time of the QADC algorithm equals to 13.97% that of the basic heuristic algorithm. The QADC algorithm is faster than earlier basic heuristic algorithm. The achieved results of these experiments conform the efficiency and the computational complexity of the two algorithms.

VI. CONCLUSION

In this study, we proposed a fast algorithm to determine the 2-Optimality consensus for binary vector collectives based on a new method to calculate the distance from a candidate consensus to members of the collective. The computational

complexity of the QADC algorithm was reduced to $O(mn)$. In the experiment, the case length of the collective members is 30, and the running time of the QADC algorithm equals to 13.97% that of the basic heuristic algorithm. The running time of the QADC algorithm equals to 16.9% that of the basic heuristic algorithm in the case length of collective members is 23. Besides, the difference between the consensus quality of our proposed QADC algorithm and the basic heuristic algorithm, the best algorithm in the literature, is not statistically significant.

In future work, we will consider keeping track of many initial candidate consensus with high diversity to enhance the consensus quality of the QADC algorithm.

APPENDIX

In this section, we present proofs of Theorem 1, 2, 3 and 4.

Proof of Theorem 1: We have

$$d(x_v, x_u) = |x_v^1 - x_u^1| + |x_v^2 - x_u^2| + \dots + |x_v^m - x_u^m|$$

$$d(x_v, x_w) = |x_v^1 - x_w^1| + |x_v^2 - x_w^2| + \dots + |x_v^m - x_w^m|$$

Therefore,

$$d(x_v, x_u) - d(x_v, x_w) = |x_v^1 - x_u^1| + |x_v^2 - x_u^2| + \dots + |x_v^m - x_u^m| - (|x_v^1 - x_w^1| + |x_v^2 - x_w^2| + \dots + |x_v^m - x_w^m|)$$

Because x_u and x_w are only different in the k^{th} component, we obtain

$$d(x_v, x_u) - d(x_v, x_w) = |x_v^k - x_u^k| - |x_v^k - x_w^k|$$

or

$$d(x_v, x_u) = d(x_v, x_w) + |x_v^k - x_u^k| - |x_v^k - x_w^k|$$

If $x_v^k = x_u^k$ then $x_v^k \neq x_w^k$. Thus,

$$|x_v^k - x_u^k| - |x_v^k - x_w^k| = 0 - 1 = -1$$

We have

$$d(x_v, x_u) = d(x_v, x_w) + |x_v^k - x_u^k| - |x_v^k - x_w^k| = d(x_v, x_w) - 1$$

If $x_v^k \neq x_u^k$ then $x_v^k = x_w^k$. Thus,

$$|x_v^k - x_u^k| - |x_v^k - x_w^k| = 1 - 0 = 1$$

We have

$$d(x_v, x_u) = d(x_v, x_w) + |x_v^k - x_u^k| - |x_v^k - x_w^k| = d(x_v, x_w) + 1$$

Thus, we have the following:

$$d(x_v, x_u) = \begin{cases} d(x_v, x_w) - 1 & \text{if } x_v^k = x_u^k \\ d(x_v, x_w) + 1 & \text{if } x_v^k \neq x_u^k \end{cases}$$

Proof of Theorem 2: Distance $d(x_1, x_w)$ is determined. Two vectors x_w and x_u are only different in the k^{th} component. From Theorem 1, we have

$$d(x_1, x_u) = \begin{cases} d(x_1, x_w) - 1 & \text{if } x_1^k = x_u^k \\ d(x_1, x_w) + 1 & \text{if } x_1^k \neq x_u^k \end{cases}$$

Distance $d(x_2, x_w)$ is determined. Two vectors x_w and x_u are only different in the k^{th} component. From Theorem 1, we have

$$d(x_2, x_u) = \begin{cases} d(x_2, x_w) - 1 & \text{if } x_2^k = x_u^k \\ d(x_2, x_w) + 1 & \text{if } x_2^k \neq x_u^k \end{cases} \dots \dots$$

Distance $d(x_n, x_w)$ is determined. Two vectors x_w and x_u are only different in the k^{th} component. From Theorem 1, we have

$$d(x_n, x_u) = \begin{cases} d(x_n, x_w) - 1 & \text{if } x_n^k = x_u^k \\ d(x_n, x_w) + 1 & \text{if } x_n^k \neq x_u^k \end{cases}$$

From the aforementioned analyses, it is clear that

$$d(x_i, x_u) = \begin{cases} d(x_i, x_w) - 1 & \text{if } x_i^k = x_u^k \\ d(x_i, x_w) + 1 & \text{if } x_i^k \neq x_u^k \end{cases} \text{ for } i = 1, 2, \dots, n.$$

It means that distances between x_u and members of the collective X can be determined based on distances between x_w and members of the collective X . ■

Proof of Theorem 3: The computational complexity of line {2} is $O(1)$. The computational complexity of line {4} and line {8} is $O(n)$. The computational complexity of lines {6}, {15}, {17}, {19}, and {26} is $O(mn)$. The computational complexity of lines {1}, {11}, {12}, {22}, and {24} is $O(m)$. Thus, the computational complexity of the QADC algorithm is $\max\{O(1), O(n), O(m), O(mn)\} = O(mn)$.

Proof of Theorem 4: In step Initialization, the value of x_c in the basic heuristic algorithm and the value of x_c in the QADC algorithm are equal. After step Initialization, the sum of squared distances between x_c and the collective's members in the basic heuristic algorithm equals to that in the QADC algorithm.

At step 1 ($k = 1$), after changing the values of the 1st component of vectors x_c , the vectors x_c in both algorithms are equal. Thus, the sum of squared distances between x_c and collective members in the basic heuristic algorithm equals to that in the QADC algorithm. Therefore, in the vectors x_c , the chosen values of the 1st component in two algorithms are the same. After step 1, the vectors x_c in two algorithms are equal.

At step 2 ($k = 2$), after changing the values of the 2nd component of vectors x_c , the vectors x_c in both algorithms are equal. Thus, the sum of squared distances between x_c and the collective members in the basic heuristic algorithm equals to that in the QADC algorithm. Therefore, in the vectors x_c , the chosen values of the 2nd component in two algorithms are the same. After step 2, the vectors x_c in two algorithms are equal.

Assume that after step p ($p \leq m - 1$), the vectors x_c in two algorithms are equal. Now, we prove that after step $p + 1$, the values of vectors x_c in two algorithms are equal.

At step $p + 1$, after changing the values of the $(p + 1)^{th}$ component of vectors x_c , the values of x_c in both algorithms are equal. Thus, the sum of squared distances between x_c and the collective members in the basic heuristic algorithm equals to that in the QADC algorithm. Therefore, in the vectors x_c , the chosen values of the $(p + 1)^{th}$ component in the two algorithms are the same. After step $p + 1$, the vectors x_c in two algorithms are equal.

REFERENCES

- [1] Y. Dong, Q. Zha, H. Zhang, G. Kou, H. Fujita, F. Chiclana, and E. Herrera-Viedma, "Consensus reaching in social network group decision making: Research paradigms and challenges," *Knowl.-Based Syst.*, vol. 162, pp. 3–13, Dec. 2018.
- [2] A. Fagiolini and A. Bicchi, "On the robust synthesis of logical consensus algorithms for distributed intrusion detection," *Automatica*, vol. 49, no. 8, pp. 2339–2350, Aug. 2013.
- [3] N. T. Nguyen, "Inconsistency of knowledge and collective intelligence," *Cybern. Syst.*, vol. 39, no. 6, pp. 542–562, Aug. 2008.
- [4] D. T. Dang, N. T. Nguyen, and D. Hwang, "Multi-step consensus: An effective approach for determining consensus in large collectives," *Cybern. Syst.*, vol. 50, no. 2, pp. 208–229, Feb. 2019.
- [5] N. T. Nguyen, "Processing inconsistency of knowledge in determining knowledge of a collective," *Cybern. Syst.*, vol. 40, no. 8, pp. 670–688, Nov. 2009.
- [6] N. T. Nguyen, *Advanced Methods for Inconsistent Knowledge Management*. London, U.K.: Springer, 2008.
- [7] A. Taghavi, E. Eslami, E. Herrera-Viedma, and R. Ureña, "Trust based group decision making in environments with extreme uncertainty," *Knowl.-Based Syst.*, vol. 191, pp. 1–10, Mar. 2020.
- [8] F. Knight, *Risk, Uncertainty and Profit, Library of Economics and Liberty*. Boston, MA, USA: Houghton Mifflin Company, 1921.
- [9] L. A. Zadeh, "Toward a generalized theory of uncertainty (GTU)—An outline," *Inf. Sci.*, vol. 172, nos. 1–2, pp. 1–40, 2005.
- [10] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.
- [11] D. Molodtsov, "Soft set theory—First results," *Comput. Math. Appl.*, vol. 37, nos. 4–5, pp. 19–31, Feb. 1999.
- [12] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.
- [13] Z. Pawlak, "An inquiry into anatomy of conflicts," *Inf. Sci.*, vol. 109, nos. 1–4, pp. 65–78, Aug. 1998.
- [14] A. Skowron and R. Deja, "On some conflict models and conflict resolutions," *Romanian J. Inf. Sci. Technol.*, vol. 3, nos. 1–2, pp. 69–82, 2002.
- [15] N. T. Nguyen and M. Malowicki, "Consistency measures for conflict profiles," in *Transactions on Rough Sets I* (Lecture Notes in Computer Science), vol. 3100. Berlin, Germany: Springer, 2004, pp. 169–186.
- [16] Y. Zhang, W. Zhang, J. Pei, X. Lin, Q. Lin, and A. Li, "Consensus-based ranking of multivalued objects: A generalized borda count approach," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 83–96, Jan. 2014.
- [17] N. T. Nguyen, "Consensus system for solving conflicts in distributed systems," *Inf. Sci.*, vol. 147, nos. 1–4, pp. 91–122, Nov. 2002.
- [18] A. Ali and M. Meilä, "Experiments with Kemeny ranking: What works when?" *Math. Social Sci.*, vol. 64, no. 1, pp. 28–40, Jul. 2012.
- [19] A. Kozierekiewicz and M. Sitarczyk, "Heuristic algorithms for 2-optimality consensus determination," in *Intelligent Information and Database Systems* (Lecture Notes in Computer Science), vol. 10751. Cham, Switzerland: Springer, 2018, pp. 48–58.
- [20] D. Gadjev and L. Pavel, "A passivity-based approach to Nash equilibrium seeking over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1077–1092, Mar. 2019.
- [21] M. Le Pira, G. Inturri, M. Ignaccolo, A. Pluchino, and A. Rapisarda, "Simulating opinion dynamics on Stakeholders' networks through agent-based modeling for collective transport decisions," *Procedia Comput. Sci.*, vol. 52, pp. 884–889, 2015.
- [22] M. Le Pira, G. Inturri, M. Ignaccolo, and A. Pluchino, "Analysis of AHP methods and the pairwise majority rule (PMR) for collective preference rankings of sustainable mobility solutions," *Transp. Res. Procedia*, vol. 10, pp. 777–787, 2015.
- [23] C. C. Aggarwal, C. Procopiuc, and P. S. Yu, "Finding localized associations in market basket data," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 51–62, 2002.
- [24] A. Griva, C. Bardaki, K. Pramataris, and D. Papakiriakopoulos, "Retail business analytics: Customer visit segmentation using market basket data," *Expert Syst. Appl.*, vol. 100, pp. 1–16, Jun. 2018.
- [25] N. Goel, S. Singh, and T. C. Aseri, "A comparative analysis of soft computing techniques for gene prediction," *Anal. Biochemistry*, vol. 438, no. 1, pp. 14–21, Jul. 2013.
- [26] G. Glazko, A. Gordon, and A. Mushegian, "The choice of optimal distance measure in genome-wide datasets," *J. Bioinform.*, vol. 21, pp. 3–11, Nov. 2005.
- [27] D. Sengupta, A. Pyne, U. Maulik, and S. Bandyopadhyay, "Reformulated kemeny optimal aggregation with application in consensus ranking of microRNA targets," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 3, pp. 742–751, May 2013.
- [28] K. De Boeck, C. Castellani, and J. S. Elborn, "Medical consensus, guidelines, and position papers: A policy for the ECFS," *J. Cystic Fibrosis*, vol. 13, no. 5, pp. 495–498, Sep. 2014.
- [29] S. T. Kandukuri, A. Klausen, H. R. Karimi, and K. G. Robbersmyr, "A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management," *Renew. Sustain. Energy Rev.*, vol. 53, pp. 697–708, Jan. 2016.
- [30] A. Kozierekiewicz-Hetmańska, "Comparison of one-level and two-level consensus methods satisfying the 2-Optimality criterion," in *Proc. ICCCI*. Cham, Switzerland: Springer, 2012, pp. 1–10.
- [31] D. T. Dang, N. T. Nguyen, and D. Hwang, "A new heuristic algorithm for 2-optimality consensus determination," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 70–75.
- [32] M. Maleszka and N. T. Nguyen, "Integration computing and collective intelligence," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 332–340, Jan. 2015.
- [33] M. J. Fischer, "The consensus problem in unreliable distributed systems (a brief survey)," in *Foundations of Computation Theory* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 1983, pp. 127–140.
- [34] R. Olfati-Saber, J. Alex Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [35] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, and H. Ning, "Analysis of blockchain solutions for IoT: A systematic literature review," *IEEE Access*, vol. 7, pp. 58822–58835, 2019.
- [36] M. Du, Q. Chen, and X. Ma, "MBFT: A new consensus algorithm for consortium blockchain," *IEEE Access*, vol. 8, pp. 87665–87675, 2020.
- [37] J. Ling, J. Liang, and M. Du, "Bipartite consensus problems on second-order signed networks with heterogeneous topologies," *IEEE Access*, vol. 8, pp. 39420–39427, 2020.
- [38] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4972–4983, Jun. 2017.
- [39] J. Zhang, Z. Wang, H. Zhang, and O. C. May, "Data-based optimal control of multiagent systems: A reinforcement learning design approach," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4441–4449, Sep. 2018.
- [40] Y. Xu, M. Fang, Z.-G. Wu, Y.-J. Pan, M. Chadli, and T. Huang, "Input-based event-triggering consensus of multiagent systems under denial-of-service attacks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 4, pp. 1455–1464, Apr. 2020.
- [41] R. Abdulghafor, S. S. Abdullah, S. Turaev, and M. Othman, "An overview of the consensus problem in the control of multi-agent systems," *Automatica*, vol. 59, no. 2, pp. 143–157, Apr. 2018.
- [42] R. Porkodi and V. Bhuvaneshwari, "The Internet of Things (IoT) applications and communication enabling technology standards: An overview," in *Proc. Int. Conf. Intell. Comput. Appl.*, Mar. 2014, pp. 324–329.
- [43] Z. Dou, G. Si, Y. Lin, and M. Wang, "An adaptive resource allocation model with anti-jamming in IoT network," *IEEE Access*, vol. 7, pp. 93250–93258, 2019.
- [44] G. Colistra, V. Pilloni, and L. Atzori, "The problem of task allocation in the Internet of Things and the consensus-based approach," *Comput. Netw.*, vol. 73, pp. 98–111, Nov. 2014.
- [45] M. Mudassar, Y. Zhai, L. Liao, and J. Shen, "A decentralized latency-aware task allocation and group formation approach with fault tolerance for IoT applications," *IEEE Access*, vol. 8, pp. 49212–49223, 2020.
- [46] S. Li, G. Oikonomou, T. Tryfonas, T. M. Chen, and L. Da Xu, "A distributed consensus algorithm for decision making in service-oriented Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1461–1468, May 2014.

- [47] A. Bassambo, R. Cui, and A. Moreno, "Wisdom of crowds in operations: Forecasting using prediction markets," *SSRN Electron. J.*, pp. 1–31, Oct. 2015.
- [48] J. Wu, F. Xiong, and C. Li, "Application of Internet of Things and blockchain technologies to improve accounting information quality," *IEEE Access*, vol. 7, pp. 100090–100098, 2019.
- [49] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.
- [50] B. Orostica and F. Nunez, "Robust gossiping for distributed average consensus in IoT environments," *IEEE Access*, vol. 7, pp. 994–1005, 2019.
- [51] X. Zhu, Y. Li, L. Fang, and P. Chen, "An improved proof-of-trust consensus algorithm for credible crowdsourcing blockchain services," *IEEE Access*, vol. 8, pp. 102177–102187, 2020.
- [52] N. O. Nawari and S. Ravindran, "Blockchain and the built environment: Potentials and limitations," *J. Build. Eng.*, vol. 25, pp. 1–16, Sep. 2019.
- [53] B. Kea and B. C.-A. Sun, "Consensus development for healthcare professionals," *Internal Emergency Med.*, vol. 10, no. 3, pp. 373–383, Apr. 2015.
- [54] R. Ferdousi, R. Safdari, and Y. Omid, "Computational prediction of drug-drug interactions based on drugs functional similarities," *J. Biomed. Informat.*, vol. 70, pp. 54–64, Jun. 2017.
- [55] Y. Zhang, "Protein structure prediction: When is it useful?" *Anal. Biochem.*, vol. 19, no. 2, pp. 145–155, 2008.
- [56] I. Ilinkin, J. Ye, and R. Janardan, "Multiple structure alignment and consensus identification for proteins," *BMC Bioinf.*, vol. 11, no. 1, p. 71, Dec. 2010.
- [57] J. Sobieska-karpinska and M. Hernes, "Consensus determining algorithm in multiagent decision support system with taking into consideration improving agent's knowledge," in *Proc. FedCSIS*, 2012, pp. 1035–1040.
- [58] J. Ma and S. Wang, "Algorithms, applications, and challenges of protein structure alignment," *Adv. Protein Chem. Struct. Biol.*, vol. 94, pp. 121–175, 2014.
- [59] J. Ye and R. Janardan, "Approximate multiple protein structure alignment using the Sum-of-Pairs distance," *J. Comput. Biol.*, vol. 11, no. 5, pp. 986–1000, Oct. 2004.
- [60] F.-J. Lapointe and G. Cucumel, "The average consensus procedure: Combination of weighted trees containing identical or overlapping sets of taxa," *Systematic Biol.*, vol. 46, no. 2, pp. 306–312, Jun. 1997.
- [61] V. Makarenkov, "An algorithm for the fitting of a tree metric according to a weighted least-squares criterion," *J. Classification*, vol. 16, no. 1, pp. 3–26, Jan. 1999.
- [62] M. Hernes and J. Sobieska-Karpińska, "Application of the consensus method in a multiagent financial decision support system," *Inf. Syst. e-Bus. Manage.*, vol. 14, no. 1, pp. 167–185, Feb. 2016.
- [63] J. Sobieska-Karpinska and M. Hernes, "The postulates of consensus determining in financial decision support systems," in *Proc. FedCSIS*, 2013, pp. 1165–1168.
- [64] G. D. Israel, "Determining sample size," in *Proc. PEOD*. Gainesville, FL, USA: Univ. Florida Inst. Food Agricult. Sci., 2013, pp. 1–5.



NGOC THANH NGUYEN (Senior Member, IEEE) is currently a Full Professor with the Wrocław University of Science and Technology, and the Head of the Information Systems Department, Faculty of Computer Science and Management. He has edited more than 30 special issues in international journals, 52 books, and 35 conference proceedings. He is the author or coauthor of five monographs and more than 350 journal articles and conference papers. His research interests

include collective intelligence, knowledge integration methods, inconsistent knowledge processing, and multi-agent systems. He was a General Chair or a Program Chair of more than 40 international conferences. He serves as an Expert of the National Center of Research and Development and European Commission in evaluating research projects in several programs, such as Marie Skłodowska-Curie Individual Fellowships, FET, and EUREKA. He has given 20 plenary and keynote speeches for international conferences, and more than 40 invited lectures in many countries. In 2009, he was granted of title Distinguished Scientist of ACM. He was also a Distinguished Visitor of IEEE and a Distinguished Speaker of ACM. He also serves as the Chair for IEEE SMC Technical Committee on Computational Collective Intelligence. He also serves as the Editor-in-Chief for *International Journal of Information and Telecommunication* (Taylor & Francis), *Transactions on Computational Collective Intelligence* (Springer), and *Vietnam Journal of Computer Science* (Springer). He is also an associate editor of several prestigious international journals.



DOSAM HWANG received the Ph.D. degree from Kyoto University, Kyoto, Japan. He has served as the Head for the Yeungnam University's Computer Engineering Department for five years from 2005 to 2009. He held a position as a Principal Researcher at the Korea Institute of Science and Technology (KIST) and has been a Visiting Professor with the Korea Advanced Institute of Science and Technology (KAIST). He is currently a Full Professor with the Department of Computer

Engineering, Yeungnam University, Republic of Korea. His research interests include natural language processing, ontology, knowledge engineering, information retrieval, and machine translation. He has so far been not only a co-chair of several international conferences but also a steering committee member of ICCCI and ACIIDS, and MISSI international conferences. More specifically, for example, he had been the Assistant Secretary of ISO/TC37/SC4 for language resource management from 2005 to 2007, and the Secretary of Korean TC for ISO/TC37/SC4 at the same time. In 2006, he was the Director of the Korean Society for Cognitive Science (KSCS), and the Korean Information Science Society (KISS), he has also been serving as the Society's Director and the Mentor of a knowledge engineering study group since 2007. In addition to this, he has also participated in several Korean national research projects, such as a project on machine translation system from 1985 to 1990, and the national IT ontology infrastructure and technology development project called "CoreOnto" from 2006 to 2009, and "Exobrain" from 2013 to 2014, the project focused on the construction of deep knowledge base and question-answering platform. He has also been in charge of an intelligent service integration based on IoT Big Data as part of Korea's another principal national research project "BK+" since 2014. In recognition of his such great commitment and contribution to the relative fields of study, he had also been honored as a Distinguished Researcher of KIST in 1988 by the Korea's Ministry of Science and Technology (MoST) and awarded a prize for Good Conduct from the Kyunghee High School in 1973. He had more than 50 publications.



DAI THO DANG received the M.S. degree in computer science from the University of Nice Sophia Antipolis, Nice, France. He is currently pursuing the Ph.D. degree in computer science with Yeungnam University, Republic of Korea, in cooperation with the Wrocław University of Science and Technology, Poland. After working in the industry for several years, he joined the University of Danang, Vietnam, as a Lecturer, in 2011. His research interests include collective intelligence,

knowledge integration methods, algorithm, consensus theory, and inconsistent knowledge processing. He is an active Reviewer of IEEE TRANSACTIONS ON CYBERNETICS.

...