Relaxed Byzantine Vector Consensus[*][†]

Zhuolun Xiang[1] and Nitin H. Vaidya[2]

1 **Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA**
xiangzl@illinois.edu

2 **Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, USA**
nhv@illinois.edu

Abstract

Byzantine vector consensus requires that non-faulty processes reach agreement on a decision (or output) that is in the convex hull of the inputs at the non-faulty processes. Recent work has shown that, for n processes with up to f Byzantine failures, when the inputs are d-dimensional vectors of reals, $n \geq \max(3f+1, (d+1)f+1)$ is the tight bound for synchronous systems, and $n \geq (d+2)f+1$ is tight for approximate consensus in asynchronous systems.

Due to the dependence of the lower bound on vector dimension d, the number of processes necessary becomes large when the vector dimension is large. With the hope of reducing the lower bound on n, we propose relaxed versions of Byzantine vector consensus: k-relaxed Byzantine vector consensus and (δ, p)-relaxed Byzantine vector consensus. k-relaxed consensus only requires consensus for projections of inputs on every subset of k dimensions. (δ, p)-relaxed consensus requires that the output be within distance δ of the convex hull of the non-faulty inputs, where distance is defined using the L_p-norm. An input-dependent δ allows the distance from the non-faulty convex hull to be dependent on the maximum distance between the non-faulty inputs.

We show that for k-relaxed consensus with $k > 1$, and for (δ, p)-relaxed consensus with constant $\delta \geq 0$, the bound on n is identical to the bound stated above for the original vector consensus problem. On the other hand, when $k = 1$ or δ depends on the inputs, we show that the bound on n is smaller when $d \geq 3$. Input-dependent δ may be of interest in practice. In essence, input-dependent δ scales with the spread of the inputs.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Byzantine consensus, vector inputs, relaxed validity conditions

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2016.26

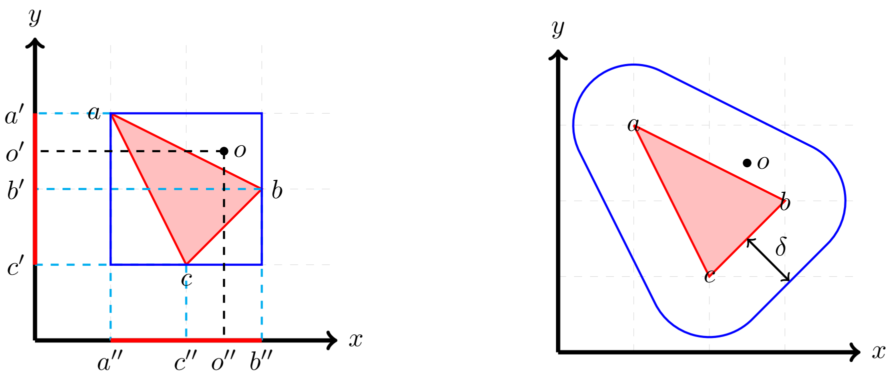
1 Introduction

Byzantine vector consensus requires that non-faulty processes reach agreement on a decision (or output) that is in the convex hull of the inputs at the non-faulty processes. This paper considers Byzantine consensus in a complete network consisting of n processes of which up to f processes may be Byzantine faulty [5]. Recent work has shown that when the inputs are d-dimensional vectors of reals, $n \geq \max(3f+1, (d+1)f+1)$ is the tight bound on the number of processes n to be able to achieve exact Byzantine consensus in a synchronous system [6, 11, 7].

[*] This research is supported in part by National Science Foundation award 1421918. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agency or the U.S. government.

[†] A brief announcement summarizing the results appeared at SPAA 2016.

20th International Conference on Principles of Distributed Systems (OPODIS 2016).
Editors: Panagiota Fatourou, Ernesto Jiménez, and Fernando Pedone; Article No. 26; pp. 26:1–26:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) k-relaxed convex hull.

(b) (δ, p)-relaxed convex hull.

Figure 1 Illustrations for relaxed convex hull.

Exact Byzantine vector consensus is defined as follows:

▶ **Definition 1** (Exact Byzantine vector consensus (exact BVC))**.** Exact BVC must satisfy the following three conditions [11, 7]:

1. *Agreement:* The decision (or output) vector at all the non-faulty processes must be identical.
2. *Validity:* The decision vector at each non-faulty process must be in the convex hull of the input vectors at the non-faulty processes.
3. *Termination:* Each non-faulty process must terminate after a finite amount of time.

Due to the dependence of the above bound on vector dimension d, the number of processes necessary becomes large when the vector dimension is large. With the hope of reducing the lower bound on n, we consider *relaxed* versions of Byzantine vector consensus: k-relaxed Byzantine vector consensus and (δ, p)-relaxed Byzantine vector consensus (we often refer to these as k-relaxed consensus and (δ, p)-relaxed consensus, respectively). For (δ, p)-relaxed consensus, we consider two formulations: constant δ and input-dependent δ, respectively.

For brevity, this paper first focuses on exact vector consensus in synchronous systems, and its relaxations. Analogous results for relaxations of approximate Byzantine vector consensus in asynchronous systems are summarized in Section 6.

Our relaxed versions of the Byzantine vector consensus problem are defined by replacing the convex hull in the above Validity condition by a *relaxed convex hull*, in particular, *k-relaxed convex hull* or *(δ, p)-relaxed convex hull* as defined later in Section 4.1 and 5.1. Intuitively, a k-relaxed convex hull, illustrated in Figure 1a, consists of points which are contained in the convex hull of the projections of inputs at non-faulty processes, where the projections are taken on every subset of k dimensions. (δ, p)-relaxed convex hull, illustrated in Figure 1b, consists of points within the distance δ of the convex hull of the inputs of non-faulty processes, where distance is defined using the L_p-norm. Formal definitions of the two *relaxations* of a convex hull are presented in Section 4.1 and 5.1.

The original vector consensus problem (Definition 1) is obtained as a special case of the two relaxed versions, by choosing $k = d$ in k-relaxed consensus, and $\delta = 0$ in (δ, p)-relaxed consensus. Also note that, when $d = 1$, the inputs are scalar, and all the L_p norms are identical. For the case of $d = 1$, (δ, p)-relaxed consensus with constant $\delta > 0$ is similar to a consensus problem addressed in prior work [2]. In particular, [2] showed (for scalar inputs) that even if a valid output is allowed to be outside the range of non-faulty inputs by up to δ, the number of processes necessary to achieve consensus cannot be smaller than $3f+1$. Thus, our work on (δ, p)-relaxed consensus extends the above prior work [2] to higher dimensions.

With the exception of Section 6, the rest of the paper assumes that the system is synchronous. For synchronous systems, we obtain the following results:

- We show that $n \geq \max(3f+1, (d+1)f+1)$ is the tight bound on n for k-relaxed consensus for $1 < k \leq d$. That is, when $k > 1$, the relaxation does not reduce the number of processes necessary. However, when $k = 1$, $n \geq 3f+1$ is the tight bound for all dimensions d. Thus, $k = 1$ significantly reduces the tight bound on n when d is large.
- For any *constant* $\delta \geq 0$ that is *independent* of the inputs, we show that $n \geq \max(3f+1, (d+1)f+1)$ remains the tight bound on n for (δ, p)-relaxed consensus. That is, the relaxation does not lower the bound.
- For values of δ specified as a function of the inputs of the non-faulty processes, we show that (δ, p)-consensus can be achieved using a smaller number of processes than the above bound for the case of constant δ. We establish a relationship between n and an achievable value of δ. For instance, for $f = 1$ and $d \geq 3$, we show that $(\frac{e_{max}}{n-2}, 2)$-consensus and $(\frac{e_{min}}{2}, 2)$-consensus is achievable with $4 \leq n \leq d+1$ processes, where e_{max} (e_{min}) is the maximum (minimum) distance between the inputs of any two fault-free processes. We also obtain results for some other values of f, n and p, and propose a conjecture for the open cases.

Section 6 summarize analogous results for asynchronous systems.

2 Related Work

Lamport, Shostak and Pease [5] developed the initial results on Byzantine fault-tolerant agreement. As noted above, for the special case of $d = 1$, our (δ, p)-relaxed consensus is similar to the so-called "$(\epsilon, \delta, \gamma)$-agreement" problem addressed in prior work [2]. *Byzantine vector consensus* (BVC) (also called *multidimensional* consensus) was introduced by Mendes and Herlihy [6] and Vaidya and Garg [11]. Tight bounds on number of processes n for Byzantine vector consensus have been obtained for synchronous [11] and asynchronous [6, 11] systems both, when the network is a complete graph. A necessary condition and a sufficient condition for *iterative* BVC in incomplete graphs were derived by Vaidya [10], however, there is a gap between these necessary and sufficient conditions.

A related problem of Convex Hull Consensus was introduced by Tseng and Vaidya [9], wherein the goal for the non-faulty processes is to try to learn the largest possible subset of the convex hull of the non-faulty inputs. For this problem, fault-tolerant algorithms have been proposed for asynchronous systems under crash faults [9] and Byzantine faults [8], respectively.

Herlihy et al. [3] introduce the *(d, ϵ)-solo approximate agreement* problem in the context of a d-solo execution model, which yields the message-passing model and the traditional shared memory model as special cases. For (d, ϵ)-solo approximate agreement, the inputs are d-dimensional vectors of reals, and the outputs must be in the convex hull of the inputs. Up to d processes may potentially choose as their outputs any arbitrary points in the convex hull of all inputs (not necessarily approximately equal to each other), while each remaining process must choose as its output a point within distance ϵ of the convex hull of the outputs of these d processes (all outputs must be within the convex hull of the inputs). Although Herlihy et al. [3] only consider crash failures, their problem formulation can be easily extended to the Byzantine fault model. The relaxed consensus formulations considered in our work are distinct from (d, ϵ)-solo agreement.

3 Notations and Terminology

The total number of processes is n, with up to f processes suffering Byzantine failures. The processes are numbered as $1, 2, \cdots, n$. Each process can communicate directly with all the processes (i.e., the network is a complete graph). The input at each process is a d-dimensional vector of reals, $d \geq 1$. We view each input as a *column* vector. Dimensions (or coordinates) of a d-dimensional vector are indexed as $1, 2, \cdots, d$. Transpose of vector u is denoted u^T. We often view a *vector* as a *point* in an appropriate space. The i-th element (or i-th coordinate) of vector v is denoted as $v[i]$. The set $\{1, 2, \cdots, d\}$ is denoted as $[1, d]$. For $u, v \in \mathbb{R}^d$, distance $\|u - v\|_p$ using L_p-norm is defined as $\|u - v\|_p = \left(\sum_{i=1}^{d} |u[i] - v[i]|^p\right)^{1/p}$. By definition, L_∞-norm is defined as $\|u - v\|_\infty = \max_{i=1,\cdots,d}(|u[i] - v[i]|)$.

A multiset may potentially contain repetitions of an element. Let $\mathcal{H}(S)$ denote the convex hull of a multiset S. For a multiset Y, when we write $X \subseteq Y$, X is a multiset in which frequency of each element is no greater than its frequency in multiset Y. The size of the multiset S, denoted $|S|$, is the number of elements in S, counting all repetitions. For a multiset Y with $|Y| \geq f$, define $\Gamma(Y)$ as

$$\Gamma(Y) = \bigcap_{X \subseteq Y, |X| = |Y| - f} \mathcal{H}(X) \tag{1}$$

In Section 4, we consider (δ, p)-relaxed Byzantine vector consensus, and Section 5 focuses on k-relaxed Byzantine vector consensus.

4 (δ, p)-Relaxed Byzantine Vector Consensus

4.1 Definition

To be able to define (δ, p)-relaxed consensus, we first define a relaxed notion of a convex hull.

▶ **Definition 2.** For $\delta \geq 0$ and $p \geq 1$, (δ, p)-relaxed convex hull $H_{(\delta,p)}$ of $S \subseteq \mathbb{R}^d$ is

$$H_{(\delta,p)}(S) = \{u \mid \|u - v\|_p \leq \delta,\ v \in \mathcal{H}(S)\}.$$

As an example, see Figure 1b. In the figure a, b, c are 2-dimensional inputs of three non-faulty processes. Let $p = 2$. The red triangle in the figure is the convex hull of a, b, c, while the area within the blue curve is the (δ, p)-relaxed convex hull of a, b, c, where δ is the length shown in the figure.

(δ, p)-relaxed consensus must satisfy the *Agreement* and *Termination* conditions stated in Section 1, and the *relaxed* validity condition below.

(δ, p)-relaxed validity: The decision vector at each non-faulty process must be in the *(δ, p)-relaxed convex hull* of the set of input vectors at the non-faulty processes.

We consider two ways to specify δ: (i) δ may be specified as a constant (Section 4.3), or (ii) δ may be input-dependent, in particular, specified as a function of the distance between the inputs at the non-faulty processes (Section 4.4).

Consider Figure 1b again, with a, b, c being the inputs of non-faulty processes. Instead of the δ shown in the figure, suppose that we choose an input-dependent δ. Specifically, let δ = minimum distance between non-faulty inputs. Then in this example, δ will equal the length of segment bc, resulting in a larger *relaxed* convex hull than that encompassed by the blue curve in Figure 1b. On the other hand, if we were to have $a = b = c$, then the minimum distance would be 0, resulting in δ being 0 as well. In this manner, we can use input-dependence to scale δ with the "spread" of the non-faulty inputs.

4.2 Preparation

As noted before, for brevity, the following discussion assumes that the systems is synchronous. Results for asynchronous systems are summarized in Section 6.

The following lemma can be proved easily.

▶ **Lemma 3.** *Solving (δ, p)-Relaxed BVC implies solving (δ', p)-Relaxed BVC where $\delta \leq \delta'$. That is, a necessary condition for (δ', p)-Relaxed BVC is also necessary for (δ, p)-Relaxed BVC, and a sufficient condition for (δ, p)-Relaxed BVC is also sufficient for (δ', p)-Relaxed BVC.*

The proof of Lemma 3 is provided in our full version [12].

We make some simple observations about two special cases of (δ, p)-relaxed BVC.

- When $\delta = 0$, the problem formulation become identical to the original exact BVC problem (Definition 1). Thus, $n \geq \max(3f+1, (d+1)f+1)$ is the necessary and sufficient condition in this special case.
- When $\delta = \infty$, the validity condition for (∞, p)-relaxed consensus is vacuous, allowing the processes to choose any fixed vector in $\mathbb{R}^d$ as the output (e.g., the processes may always choose the all-0 vector as their output and still satisfy the validity condition with $\delta = \infty$).

4.3 Results for constant δ

▶ **Theorem 4.** $n \geq \max(3f+1, (d+1)f+1)$ *is necessary and sufficient for (δ, p)-Relaxed Exact BVC in a synchronous system, where $0 < \delta < \infty$ and $1 \leq p$.*

Proof. When $d = 1$, the inputs are scalar, and all the L_p norms are identical. For the case of $d = 1$, (δ, p)-relaxed consensus is similar to a problem that was addressed in prior work [2]. For this case, it can be shown similarly that $n \geq 3f + 1$ is necessary and sufficient. Therefore, in the rest of the proof, we assume $d \geq 2$.

Sufficiency: Due to the equivalence of the original Exact BVC and $(0, p)$-Relaxed Exact BVC, for $d \geq 2$ and $1 \leq p$, $n \geq (d+1)f + 1$ is sufficient for $(0, p)$-Relaxed Exact BVC. Then by Lemma 3, this condition is also sufficient for (δ, p)-Relaxed BVC where $0 < \delta < \infty$.

Necessity: We first prove that $n \geq d + 2$ is necessary for $f = 1$ and $p = \infty$. The proof is by contradiction. Suppose that $n = d + 1$ and (δ, ∞)-Relaxed Exact BVC is achievable using a certain algorithm.

Let us suppose that exactly one process is Byzantine faulty, but the faulty process correctly follows any specified algorithm. Due to this restricted behavior, it is possible for all the processes to correctly learn the input of all the other processes. If we can show that $d + 1$ processes are insufficient despite the above constraint on the faulty process, then $d + 1$ are insufficient when arbitrary behaviors are allowed for the faulty process. Hereafter, we assume that all the processes follow the specified algorithm.

Let the i^{th} column of the following $d \times (d+1)$ matrix S be an input vector of the i^{th} process, where $x > 2d\delta$.

$$S = \begin{pmatrix} x & 0 & \cdots & \cdots & 0 & 0 \\ 0 & x & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & x & 0 & 0 \\ 0 & \cdots & \cdots & 0 & x & 0 \end{pmatrix}$$

For $1 \leq i \leq d$, the i-th coordinate of the i-th input is x, and the rest of the coordinates are 0. The $d+1$-th input is all-0. Let Y denote the set of all inputs specified in matrix S. If N is the set of non-faulty processes, then the output must be in $H_{(\delta,\infty)}(N)$. However, since the identity of any faulty process is unknown, the decision vector must be in

$$\bigcap_{T \subseteq Y, |T| = |Y| - f} H_{(\delta,\infty)}(T)$$

where $f = 1$.

Now we consider different choices of T:

- *Observation 1:* Consider T as the set of all inputs except the input of process i, $1 \leq i \leq d$. Then the i^{th} element of each of the d inputs in T is 0. Therefore the i^{th} element of all the vectors in $H_{(\delta,\infty)}(T)$ – and consequently in the output – must be less than or equal to δ due to the definition of (δ, ∞)-relaxed validity.
- *Observation 2:* Consider T as the set of all inputs except the input of process $(d+1)$. The vectors in $H_{(\delta,\infty)}(T)$ are within distance δ (where the distance is measured using the L_∞ norm) of the convex hull $\mathcal{H}(T)$. In each convex combination of elements in T used to obtain the convex hull $\mathcal{H}(T)$, at least one of the weights must be $\geq \frac{1}{d}$. Hence at least one element of each vector in $\mathcal{H}(T)$ must be $\geq \frac{x}{d}$. Thus, at least one element of each vector in $H_{(\delta,\infty)}(T)$ – and consequently the output – must be $\geq \frac{x}{d} - \delta > \delta$ (recall that $x > 2d\delta$).

Thus, Observation 1 and 2 contradict each other, proving that $n = d+1$ is not sufficient for $f = 1$.

For $f > 1$, we can use the well-known simulation approach to show $n = (d+1)f$ is not sufficient [5]. Therefore, $n \geq (d+1)f + 1$ is necessary for (δ, ∞)-*Relaxed Exact BVC* with $f > 1$, completing the proof.

Now, for any vector x, $\|x\|_\infty \leq \|x\|_p$, for $1 \leq p < \infty$ [4]. Therefore, we have

$$H_{(\delta,p)} \subseteq H_{(\delta,\infty)}\,.$$

Then, the argument above for (δ, ∞)-consensus would imply that $n \geq (d+1)f + 1$ is also necessary for (δ, p)-*Relaxed Exact BVC.* ◀

Theorem 4 shows a disappointing result. Specifically, when δ is a constant, the relaxed validity condition of (δ, p)-relaxed consensus does not yield a reduction in the number of processes necessary to solve the problem.

On the other hand, as shown in Section 4.4 below, the tight bound on n can be lower when δ is input-dependent. In general, the results for input-dependent δ are more challenging to prove than the results presented above.

4.4 Results for input-dependent δ

In contrast to **constant** δ, if the relaxation parameter δ depends on the non-faulty inputs themselves, then the (δ, p)-Relaxed Exact BVC problem may be solvable when $3f+1 \leq n \leq (d+1)f$. Interpreting a d-dimensional vector as a point in the d-dimensional Euclidean space, define E_+ as the set of edges between the inputs at the non-faulty processes in any given execution. The input-dependent δ will be defined below using the edge set E_+. In particular, we prove the following results for $p = 2$. An extension to general L_p-norm is provided in our full version [12].

▶ **Theorem 5.** *Let E_+ denote the set of edges between the inputs at non-faulty processes. When* (i) *$f = 1$ and $4 \leq n \leq d+1$, or* (ii) *$f \geq 2$ and $3f+1 \leq n = (d+1)f$, $(\widehat{\delta}, 2)$-relaxed consensus is achievable where*

$$\widehat{\delta} = \frac{\max_{e \in E_+} \|e\|_2}{\lfloor \frac{n}{f} \rfloor - 2}.$$

Observe that $\widehat{\delta}$ depends on the inputs of non-faulty processes – however, for brevity, our notation $\widehat{\delta}$ does not make that dependence explicit. If inputs of all non-faulty processes happen to be identical, then $\widehat{\delta}$ would be 0. On the other hand, if the non-faulty inputs are far apart, then $\widehat{\delta}$ would be accordingly larger (larger $\widehat{\delta}$ allows the output to be farther away from the convex hull of the non-faulty inputs).

The cases considered in Theorem 5 satisfy the constraint $3f+1 \leq n \leq (d+1)f$. It is well-known that at least $3f+1$ processes are necessary for scalar Byzantine consensus. A similar argument, as presented in our full version [12], shows that $n \geq 3f+1$ is also necessary for (δ, p)-relaxed consensus with input-dependent δ for all d. Secondly, if $n > (d+1)f$, then $\delta = 0$ is achievable (i.e., the "unrelaxed" problem is solvable). Hence the constraint $3f+1 \leq n \leq (d+1)f$ is meaningful. Note that this constraint can only be met when $d \geq 3$.

The above theorem considers two special cases. When $f = 1$, the above expression becomes $\widehat{\delta} = \frac{\max_{e \in E_+} \|e\|_2}{n-2}$, and when $n = (d+1)f$, it becomes $\widehat{\delta} = \frac{\max_{e \in E_+} \|e\|_2}{d-1}$. The above theorem does not make any claims about the case when $f \geq 2$ and $3f+1 \leq n < (d+1)f$. We conjecture that $\widehat{\delta}$ specified in the theorem is achievable even in these cases. For the case of $f = 1$, we are able to strengthen the above result, as stated next.

▶ **Theorem 6.** *When $f = 1$ and $4 \leq n \leq d+1$, $(\widehat{\delta}, 2)$-relaxed consensus is achievable where*

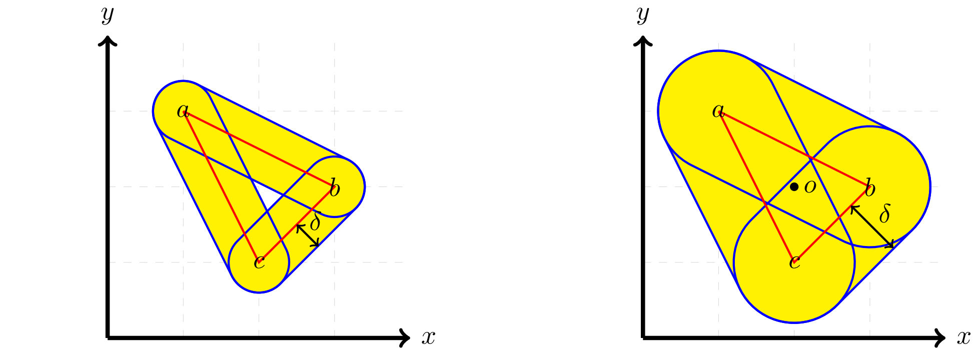
$$\widehat{\delta} = \min\left(\frac{\min_{e \in E_+} \|e\|_2}{2}, \frac{\max_{e \in E_+} \|e\|_2}{n-2}\right)$$

Our proof of Theorems 5 and 6 is constructive. We show that the *Relaxed BVC* (R-BVC) algorithm presented below satisfies the claims in these theorems. While algorithm R-BVC is quite straightforward, our key contribution here is to show that the algorithm achieves $(\widehat{\delta}, 2)$-relaxed consensus, as claimed in Theorems 5 and 6.

Algorithm R-BVC

Let v_i denote the d-dimensional input at process i, $1 \leq i \leq n$.

- *Step 1:* Each process i performs a Byzantine broadcast of its input v_i. Byzantine broadcast of each element of the vector v_i can be performed separately by using any Byzantine broadcast algorithm, such as [5]. $n \geq 3f+1$ suffices for the correctness of Byzantine



(a) Smaller δ.

(b) Larger δ.

Figure 2 Impact of δ on $\Gamma_{(\delta,2)}(\{a,b,c\})$.

broadcast in a completely connected network. At the completion of Step 1, each process will receive a multiset $S = \{a_i \mid 1 \leq i \leq n\}$, where for a non-faulty process i, $a_i = v_i$, the input of process i, and for a faulty process j, a_j may be any arbitrary point in the d-dimensional space. Importantly, all non-faulty processes obtain identical multiset S. The points in S received from non-faulty processes are said to *non-faulty inputs*, and the remaining points are said to be *faulty inputs*.

- *Step 2:* Each process determines the smallest value δ such that

$$\Gamma_{(\delta,2)}(S) = \bigcap_{T\subseteq S, |T|=|S|-f} H_{(\delta,2)}(T)$$

is non-empty, and for this value of δ, the process deterministically chooses a point in $\Gamma_{(\delta,2)}(S)$ as its output. All processes use the same deterministic function to choose the output from $\Gamma_{(\delta,2)}(S)$.

Let $\delta^*(S)$ denote the smallest value of δ for which $\Gamma_{(\delta,2)}(S)$ is non-empty. When the set S is clear from context, we will abbreviate $\delta^*(S)$ simply as δ^*. δ^* is well-defined, because by choosing δ sufficiently large, it is always possible to ensure that $\Gamma_{(\delta,2)}(S)$ is non-empty. This is illustrated in Figures 2a and 2b for the case when $d = 2$, $S = \{a, b, c\}$, and $f = 1$. Note that the cylinder around each red edge is the $(\delta, 2)$-relaxed convex hull of the endpoints of that edge. With the smaller value of δ used in Figure 2a, $\Gamma_{(\delta,2)}(S)$ is empty, but it is non-empty in Figure 2b with a larger value of δ.

Recall that, for $(\widehat{\delta}, 2)$-relaxed consensus, the validity condition requires the output to be in the relaxed convex hull $H_{(\widehat{\delta},2)}$ of the non-faulty inputs. However, since a non-faulty process does not know which processes are faulty, and let $\delta^*(S)$ denote the smallest value of δ for which $\Gamma_{(\delta,2)}(S)$ is non-empty. $\widehat{\delta}$ depends on inputs of non-faulty processes, it is not possible for the non-faulty processes to compute $\widehat{\delta}$ explicitly. Instead, the above algorithm chooses an output in $\Gamma_{(\delta^*,2)}(S)$ (where $\delta^*(S)$ is defined above). We will show that $\delta^*(S) \leq \widehat{\delta}$. By Lemma 3, this implies that $(\widehat{\delta}, 2)$-relaxed consensus is achieved.

4.5 Proof of Theorem 6

We now prove Theorem 6 stated previously.

Proof. Note that the discussion below makes frequent references to set S of inputs collected in Step 1 of algorithm R-BVC. Recall that E_+ is the set of edges between non-faulty inputs in S. Let E denote the set of edges between *all* points in S.

Consider set $S = \{a_1, \cdots, a_n\}$ obtained in Step 1 of algorithm R-BVC. If the n points in S are **not** affinely independent, then the $n-1$ vectors in the set $\{a_i - a_n \mid i \neq n, 1 \leq i \leq n\}$ are **not** linearly independent. In this case, it is easy to show that $\delta^*(S) = 0$. The proof of this claim is presented in our full version [12]. Thus, in this case, $(0, 2)$-relaxed consensus (i.e., "unrelaxed" version) is achievable.

Hereafter, we assume that the n points in S **are** affinely independent, thus, the $n-1$ vectors in $\{a_i - a_n \mid i \neq n, 1 \leq i \leq n\}$ are linearly independent.

Recall that Theorem 6 assumes $f = 1$. We consider two cases separately: (I) $n = d+1$ and (II) $4 \leq n < d+1$.

4.5.1 Case I: $n = d + 1$

We begin with a useful lemma.

▶ **Lemma 7.** *Let $d \geq 2$. When the points in $S = \{a_1, \cdots, a_{d+1}\}$ are affinely independent, let r be the radius of the inscribed sphere of the simplex(contained within the simplex and tangent to each of the simplex's faces) formed by the points in S. Let E denote the set of edges between every pair of vertices of this simplex, and let π_k denote the facet of the simplex that contains $\{a_i \mid i \neq k, \ 1 \leq i \leq d+1\}$ (i.e., all vertices except a_k), $k = 1, \cdots, d+1$. Then π_k itself is a simplex in a $(d-1)$-dimensional subspace. Let r_k be the radius of the $(d-1)$-dimensional inscribed sphere of π_k in this $(d-1)$-dimensional subspace. Then,*

$$r = \delta^*(S), \tag{2}$$

$$r < \min_{1 \leq k \leq d+1} r_k, \quad \textit{and} \tag{3}$$

$$r < \frac{\max_{e \in E} \|e\|_2}{d} \tag{4}$$

The proof of the three claims in the above lemma are presented in our full version [12].

Proof that $\delta^*(S) < \frac{\min_{e \in E_+} \|e\|_2}{2}$

This claim will be proved here by induction for any simplex in dimensions ≥ 2. Consider a simplex in 2 dimensions. Then the simplex is simply a triangle, and it can be easily shown that the radius of its inscribed sphere is $<$ half the length of the shortest edge in the triangle. Our full version [12] presents the proof of this claim.

Now, suppose that, for every simplex of dimension $k \geq 2$, the radius of its inscribed sphere is $<$ half the length of its shortest edge, and consider a simplex A of dimension $k+1$. Equation (3) in Lemma 7 then implies that the radius of the inscribed sphere of A is also $<$ half the length of the shortest edge in A. Inductively, for the simplex formed by S, this proves that $r < \frac{\min_{e \in E} \|e\|_2}{2}$. Since $E_+ \subseteq E$, it then follows that $r < \frac{\min_{e \in E_+} \|e\|_2}{2}$. Then by equation (2) of Lemma 7, we have that

$$\delta^*(S) < \frac{\min_{e \in E_+} \|e\|_2}{2}. \tag{5}$$

Proof that $\delta^*(S) < \frac{\max_{e \in E_+} \|e\|_2}{d-1}$

Without loss of generality, assume that process 1 is faulty, and thus $a_1 \in S$ is the only faulty input in S. Let π_1 be the facet of the simplex formed by the points in $S - \{a_1\}$, and r_1 be the radius of $(d-1)$-dimensional inscribed sphere of π_1. Observe that π_1 is isomorphic to a simplex in $d-1$ dimensions. By equation (4) of Lemma 7 (when applied to $d-1$ dimensional points), we have $r_1 < \frac{\max_{e\in E'}\|e\|_2}{d-1}$, where E' is the set of edges between the input corresponding to π_1 (i.e., inputs in $S-\{p_1\}$). Now, since p_1 is the input of the only faulty process, it follows that E' equals E_+ (i.e., the set of edges between non-faulty inputs). Thus, $r_1 < \frac{\max_{e\in E_+}\|e\|_2}{d-1}$. By equations (2) and (3) of Lemma 7, we then have $\delta^*(S) = r < r_1 < \frac{\max_{e\in E_+}\|e\|_2}{d-1}$.

This result, in conjunction with (5) proves that

$$\delta^*(S) \quad < \quad \min\left(\frac{\min_{e\in E_+}\|e\|_2}{2}, \frac{\max_{e\in E_+}\|e\|_2}{d-1}\right). \tag{6}$$

Note that, since $n=d+1$, $d-1$ equals $n-2$, thus (6) proved Theorem 6 when $n=d+1$ and $f=1$.

4.5.2 Case II: $4 \le n < d+1$

Since the vectors in $\{a_i - a_n \mid 1 \le i < n\}$ are linearly independent, these vectors form a $n-1$ dimensional subspace W (where $n-1<d$). Then we can find a projection matrix P that projects these d-dimensional vectors into a $(n-1)$-dimensional space, while preserving the distances between the points in $S=\{a_1,\cdots,a_n\}$. Then the n points $Pa_1,\cdots,Pa_n$ form a simplex in a $(n-1)$-dimensional subspace. By the results in Case I, and substituting d by $n-1$, the claim follows in Case II.

Thus, we have proved that algorithm R-BVC achieves $(\delta^*(S),2)$-relaxed consensus, where $\delta^*(S) < \widehat{\delta} = \min\left(\frac{\min_{e\in E_+}\|e\|_2}{2}, \frac{\max_{e\in E_+}\|e\|_2}{n-2}\right)$. Then, by Lemma 3, R-BVC also achieves $(\widehat{\delta},2)$-relaxed consensus, proving Theorem 6. ◀

4.6 Proof of Theorem 5

We presented Theorem 5 earlier. The proof of Theorem 5 is significantly more complex than Theorem 6. For lack of space, we only sketch the proof here. A complete proof is presented in our full version [12].

Proof Sketch. Theorem 5 considers the case when $3f+1 \le n = (d+1)f$. Since the case of $f=1$ is provided by Theorem 6, here we focus on case (ii) where $n=(d+1)f$ and $f\ge 2$.

For brevity below, we often refer to $\delta^*(S)$ simply as δ^*. Let P_i be the subsets of S of size $(n-f)=df$, $i=1,\cdots,\binom{n}{f}$. Recall that δ^* denotes the smallest value of δ for which $\Gamma_{(\delta,2)}(S)$ is non-empty. Equivalently, there exists a point in $\Gamma_{(\delta,2)}(S)$, whose largest distance to any $\mathcal{H}(P_i)$ is minimized compared to any other points, and this largest distance is exactly δ^*. Hence, it is easy to see that $\delta^* = \min_{p\in\mathbb{R}^d}\left(\max_{i=1,\cdots,\binom{n}{f}} dist(p,\mathcal{H}(P_i))\right)$. Let $p_0 \in \arg\min_{p\in\mathbb{R}^d}\left(\max_{i=1,\cdots,\binom{n}{f}} dist(p,\mathcal{H}(P_i))\right)$. Also, suppose that m of the P_i's are distance exactly δ^* from p_0; let Q_j, $j=1\cdots,m$, denote these m subsets (P_i's). The rest of the subsets are at distance $>\delta^*$. Now we consider the following two cases.

- *Case 1:* $1\le m\le d$:
  In this case, we can show that $\delta^*=0$ by contradiction. Suppose $\delta^*>0$. We can then move p_0 towards a suitable direction to find a new point p' which is closer to all $\mathcal{H}(P_i)$ for all i, when compared to p_0, contradicting the definition of p_0. Thus, $\delta^*=0$. Therefore, $(0,2)$-relaxed consensus is achievable, and the theorem is true in Case 1.

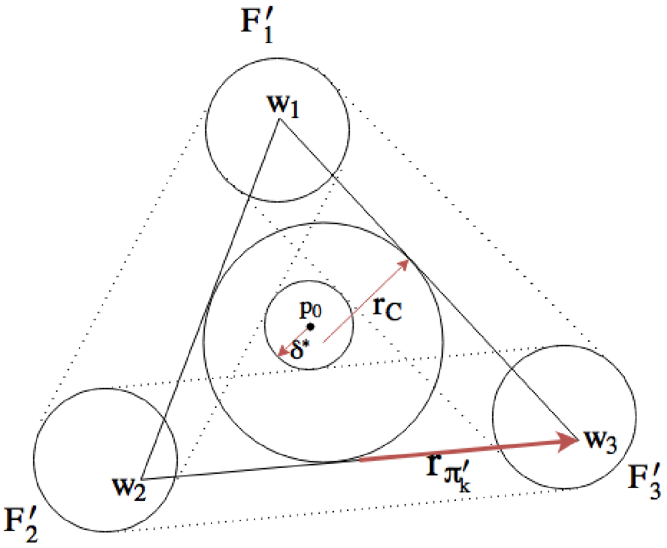


Figure 3 Illustration of the proof.

- *Case 2:* $m \geq d+1$:
  In this case, we can first show that there exist $d+1$ distinct sets $Q'_1, \cdots, Q'_{d+1}$ in $\{Q_j\}$, such that $\bigcap_{i=1}^{d+1} \mathcal{H}(Q'_i) = \emptyset$. Let $F'_i = S - Q'_i$, $i = 1, \cdots, d+1$. We can show that these F'_i's are disjoint, and form a partition of S.
  Then, we can show the following three claims to prove that p_0 is contained in the simplex A formed by $W = \{w_i \mid 1 \leq i \leq d+1\}$, with each $w_i \in F'_i$, as illustrated in the Figure 3 (here $d = 2$ as an simple example).

  ▶ **Claim 8.** *Consider a set Z of size $d+1$ consisting of one point each in F'_i. Then the $d+1$ points in Z are affinely independent, and $A = \mathcal{H}(Z)$ is a simplex in d-dimensions.*
  ▶ **Claim 9.** $\mathcal{H}(S) - \bigcup_{i=1}^{d+1} \mathcal{H}(Q'_i) \subseteq A$.
  ▶ **Claim 10.** *p_0 is contained in the simplex A formed by $W = \{w_i \mid 1 \leq i \leq d+1\}$, i.e., $p_0 \in \mathcal{H}(W)$.*

  In Figure 3, $A = \mathcal{H}(Z)$ is the triangle (which is a simplex in dimension 2) formed by w_1, w_2, w_3 (Claim 8). $\mathcal{H}(S) - \bigcup_{i=1}^{d+1} \mathcal{H}(Q'_i)$ is the dotted triangle in the center, which is contained in the triangle A (Claim 9). It is clear from the figure that p_0 is contained in A (Claim 10).

  Then, by Claim 10, p_0 is contained in the simplex A formed by $W = \{w_i \mid 1 \leq i \leq d+1\}$ with each $w_i \in F'_i$, where $|F'_i| = f$ and $\cup_{i=1}^{d+1} F'_i = S$.
  Let π'_k denote the facet of simplex A that contains $\{w_i \mid i \neq k,\ 1 \leq i \leq d+1\}$. Let r_A be the radius of the sphere inscribed in the simplex A. Since p_0 is contained in the simplex A, it can be shown that $r_A \geq \delta^*$. Then considering the distribution of faulty inputs among sets F'_i. There are two cases:
  1. There exists k, $1 \leq k \leq d+1$, such that all the faulty inputs are contained in F'_k. Then π'_k is the convex hull of a subset of non-faulty inputs. By equation (3) of Lemma 7, we have $r_A < r_{\pi'_k}$. Then by equation (4) of Lemma 7, $r_{\pi'_k} < \frac{\max_{e \in E'} \|e\|_2}{d-1}$, where E' is the set of edges between vertices of π'_k. Since π'_k consists of only non-faulty inputs,

we have $r_{\pi'_k} < \frac{\max_{e \in E'} \|e\|_2}{d-1} \le \frac{\max_{e \in E_+} \|e\|_2}{d-1}$. Hence $\delta^* \le r_A < r_{\pi'_k} < \frac{\max_{e \in E_+} \|e\|_2}{d-1}$. The relationship between δ^* and $r_{\pi'_k}$ is shown intuitively in Figure 3.

2. There does not exist k such that all the faulty inputs are contained in F'_k. Then we can take one non-faulty input from each F'_i and form a simplex C which contains p_0. By equation (4) in Lemma 7, we know that $r_C < \frac{\max_{e \in E''} \|e\|_2}{d}$, where E'' is the set of edges between the vertices of C. Since vertices of C are all non-faulty inputs, we have $r_C < \frac{\max_{e \in E''} \|e\|_2}{d} \le \frac{\max_{e \in E_+} \|e\|_2}{d}$. Hence, $\delta^* \le r_C < \frac{\max_{e \in E_+} \|e\|_2}{d} < \frac{\max_{e \in E_+} \|e\|_2}{d-1}$. The relationship between δ^* and r_C is also shown intuitively in Figure 3.

Then, by Lemma 3, R-BVC achieves $(\widehat{\delta}, 2)$-relaxed consensus, proving Theorem 5 ◀

4.7 A Conjecture

While Theorem 6 covers all the interesting cases for $f = 1$, Theorem 5 leaves some cases undecided for $f \ge 2$. We conjecture that the claim of Theorem 5 is also true for $f = 2$ and $3f + 1 \le n < (d + 1)f$.

5 *k*-Relaxed Byzantine Vector Consensus

5.1 Definition

To be able to define k-relaxed consensus, we first introduce other definitions.

▶ **Definition 11** (D-projection). Let $D = \{d_1, d_2, \cdots, d_k\}$ where $1 \le d_i < d_j \le d$ for $1 \le i < j \le k$. For $u \in \mathbb{R}^d$ define projection $g_D(u) = v$ where $v \in \mathbb{R}^k$ and $v[i] = u[d_i]$. For multiset S consisting of points in $\mathbb{R}^d$, define $g_D(S) = \{g_D(u) \mid u \in S\}$.

Thus, D-projection g_D defined above projects a given vector on the specified set of k coordinates. When a set is provided as an argument, g_D returns D-projection of each vector in that set. While g_D is not a one-to-one function, with an abuse of terminology, we will define its inverse.

▶ **Definition 12** (Inverse of D-projection). For $v \in \mathbb{R}^k$, define $g_D^{-1}(v) = U$ where $U \subset \mathbb{R}^d$, such that $u \in U$ if and only if $g_D(u) = v$. For multiset S consisting of points in $\mathbb{R}^k$, define $g_D^{-1}(S) = \bigcup_{v \in S} g_D^{-1}(v)$.

For example, suppose that $d = 4$, $D = \{1, 3\}$, $u = (7, -4, -2, 0)^T$, and $v = (7, -2)^T$. Then $g_D(u) = (7, -2)^T$, and $g_D^{-1}(v) = \{(7, a, -2, b)^T \mid a, b \in \mathbb{R}\}$.

▶ **Definition 13.** The k-relaxed convex hull H_k of $S \subset \mathbb{R}^d$ is defined as

$$H_k(S) = \{u \mid g_D(u) \in \mathcal{H}(g_D(S)), \forall D \in \mathcal{D}_k\}$$

where $\mathcal{D}_k = \{D \mid D \subseteq [1, d],\ |D| = k\}$. Equivalently,

$$H_k(S) = \bigcap_{D \in \mathcal{D}_k} g_D^{-1}(\mathcal{H}(g_D(S)))\,.$$

As an example, see Figure 1a. In this example, we consider inputs a, b, c of dimension 2. Let $k = 1$. The red triangle is the convex hull of a, b, c, while the area within the blue rectangle is the k-relaxed convex hull of a, b, c (for $k = 1$). Any point in H_k, o for instance, is contained in the convex hull formed by projections of a, b, c on each dimension (because $k = 1$). o' and o'' are projections of o on each of the dimensions. Clearly, o' is contained in the convex hull formed by a', b', c', and o'' is contained in the convex hull formed by a'', b'', c'' (projections of a, b, c).

Now we define k-relaxed consensus. In particular, k-relaxed consensus must satisfy the *Agreement* and *Termination* conditions in Section 1, and the *relaxed* validity condition below.

> **k-relaxed validity:** The decision vector at each non-faulty process must be in the *k-relaxed convex hull* of the set of input vectors at the non-faulty processes.

5.2 Results

This section presents our key results on k-Relaxed Byzantine Vector Consensus. As noted before, results for asynchronous systems are summarized in Section 6.

We begin with some simple observations about a few special cases of relaxed BVC.

- For *k-relaxed BVC* with $k = d$, the problem formulation becomes identical to the original exact BVC problem(Definition 1). Thus, $n \geq \max(3f + 1, (d + 1)f + 1)$ is the necessary and sufficient condition in these special cases.
- When $k = 1$, k-relaxed consensus (i.e., 1-relaxed consensus) can be achieved using any Byzantine consensus algorithm for scalar inputs (such as [1]). In particular, the processes perform d instances of a scalar Byzantine consensus algorithm. The input of any process j for the i-th instance is the i-th coordinate of process j's d-dimensional input for 1-relaxed consensus. Each process j uses the output of the i-th instance of scalar Byzantine consensus above to be the i-th coordinate of its output vector for 1-relaxed consensus. It is easy to verify that this solution correctly achieves 1-relaxed consensus. Thus, the tight bound on n for $k = 1$ is identical to the well-known bound for scalar Byzantine consensus, namely, $n \geq 3f + 1$ [5].

▶ **Theorem 14.** *$n \geq (d + 1)f + 1$ is necessary and sufficient for k-relaxed consensus in a synchronous system when $2 \leq k \leq d - 1$.*

The proof of Theorem 14 is provided in our full version [12]. Analogous to Theorem 4, the above theorem also shows a negative result.

6 Results for Asynchronous Systems

In this section, we briefly present our results for relaxed Byzantine vector consensus in asynchronous systems.

Approximate Byzantine vector consensus in asynchronous systems must satisfy the *Validity* and *Termination* conditions stated in Definition 1, and the *ϵ-Agreement condition* below.

> **ϵ-Agreement:** The decision (or output) vectors at any two non-faulty processes must be within distance ϵ of each other, where $\epsilon > 0$.

From previous studies, $n \geq (d+2)f+1$ is necessary and sufficient for the above approximate Byzantine consensus [6, 11, 7]. As we will see soon, the results for relaxed Byzantine vector consensus in asynchronous systems are analogous to those of synchronous systems.

6.1 (δ, p)-Relaxed Byzantine Vector Consensus

(δ, p)-relaxed approximate Byzantine vector consensus in asynchronous systems must satisfy the *ϵ-Agreement condition* above, the *(δ, p)-relaxed Validity* condition in Section 4.1 and the *Termination* condition.

With Constant δ

▶ **Theorem 15.** *$n \geq (d+2)f+1$ is necessary and sufficient for (δ, p)-Relaxed Approximate BVC in an asynchronous system, where $0 < \delta < \infty$ and $1 \leq p$.*

The proof of Theorem 15 is analogous to the one of Theorem 4, and is provided in our full version [12].

The condition on number of processes $n \geq (d+2)f+1$ remains unchanged when we relax the validity condition, compared with the original Byzantine vector consensus problem.

With Input-dependent δ

For brevity, we only present the results. The algorithm is provided in our full version [12].

▶ **Theorem 16.** *Suppose $(\widehat{\delta}, p)$-Relaxed Exact BVC is achievable where*

$$\widehat{\delta} = \kappa(n, f, d, p) \max_{e \in E_+} \|e\|_p$$

where $\kappa(n, f, d, p)$ is a finite constant that may depend on number of processes n, number of failures f, dimension of the inputs d and L_p norm. E_+ is the set of edges between pairs of non-faulty inputs in S.

Then $(\widehat{\delta}, p)$-Relaxed Approximate BVC is achievable where

$$\widehat{\delta} = \kappa(n-f, f, d, p) \max_{e \in E_+} \|e\|_p$$

where $\kappa(n, f, d, p)$, S and E_+ are defined above.

The proof of Theorem 16 is provided in our full version [12].

By Theorem 16 and the previous results in Section 4.4, (δ, p)-relaxed approximate consensus can be solved with fewer number of processes. Moreover, we established a formula between the size of feasible δ for solving synchronous case and that for solving asynchronous case. Namely, we can infer the feasible δ for asynchronous case from that for synchronous case.

6.2 k-Relaxed Byzantine Vector Consensus

k-relaxed approximate Byzantine vector consensus in asynchronous systems must satisfy the *ϵ-Agreement condition* above, the *k-relaxed Validity* condition in Section 5.1 and the *Termination* condition.

For $k = 1$, $n \geq 3f+1$ is necessary and sufficient, and for $k = d$, $n \geq (d+2)f+1$ is necessary and sufficient. Bounds for $d = 1, 2$ are included in the above results.

▶ **Theorem 17.** *$n \geq (d+2)f+1$ is necessary and sufficient for $2 \leq k \leq d-1$ in k-Relaxed Approximate BVC in an asynchronous system.*

The proof of Theorem 17 is analogous to the one of Theorem 14, and is provided in our full version [12].

Similar to Theorem 15 and Theorem 14, the result for k-consensus in asynchronous systems is also negative.

7 Summary

This paper studies *k-relaxed Byzantine vector consensus*, and (δ, p)*-relaxed Byzantine vector consensus* with constant and input-dependent δ both. For k-relaxed consensus and (δ, p)-relaxed consensus with constant δ, the tight necessary and sufficient condition on number of processes is shown to be identical to that for the original ("unrelaxed") consensus problem. For the case of input-dependent δ, we show that the problem is solvable with fewer processes.

References

1 Brian A. Coan and Jennifer L. Welch. Modular construction of a byzantine agreement protocol with optimal message bit complexity. *Information and Computation*, 97(1):61–85, 1992.

2 Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. *Easy impossibility proofs for distributed consensus problems*. Springer, 1990.

3 Maurice Herlihy, Sergio Rajsbaum, Michel Raynal, and Julien Stainer. Computing in the presence of concurrent solo executions. In *LATIN 2014: Theoretical Informatics*, pages 214–225. Springer, 2014.

4 Gottfried Köthe. *Topological vector spaces*. Springer, 1983.

5 Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

6 Hammurabi Mendes and Maurice Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 391–400. ACM, 2013.

7 Hammurabi Mendes, Maurice Herlihy, Nitin H. Vaidya, and Vijay K. Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.

8 Lewis Tseng and Nitin H. Vaidya. Byzantine convex consensus: An optimal algorithm. *arXiv preprint arXiv:1307.1332*, 2013.

9 Lewis Tseng and Nitin H. Vaidya. Asynchronous convex hull consensus in the presence of crash faults. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, pages 396–405. ACM, 2014.

10 Nitin H. Vaidya. Iterative byzantine vector consensus in incomplete graphs. In *Distributed Computing and Networking*, pages 14–28. Springer, 2014.

11 Nitin H. Vaidya and Vijay K. Garg. Byzantine vector consensus in complete graphs. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, pages 65–73. ACM, 2013.

12 Zhuolun Xiang and Nitin H Vaidya. Relaxed byzantine vector consensus. *arXiv preprint arXiv:1601.08067*, 2016.