<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>New Constructions for Nondominated k-Coteries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Choi, Eun Hye; Tsuchiya, Tatsuhiro; Kikuno, Tohru</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>IEICE transactions on information and systems. E83-D(7) P.1526-P.1532</td>
</tr>
<tr>
<td><strong>Issue Date</strong></td>
<td>2000-07-25</td>
</tr>
<tr>
<td><strong>Text Version</strong></td>
<td>publisher</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/11094/27249">http://hdl.handle.net/11094/27249</a></td>
</tr>
<tr>
<td><strong>DOI</strong></td>
<td></td>
</tr>
<tr>
<td><strong>rights</strong></td>
<td>Copyright © 2000 The Institute of Electronics, Information and Communication Engineers</td>
</tr>
</tbody>
</table>

**Osaka University Knowledge Archive : OUKA**

https://ir.library.osaka-u.ac.jp/repo/ouka/all/

Osaka University
New Constructions for Nondominated $k$-Coteries

Eun Hye CHOI†, Nonmember, Tatsuhiro TSUCHIYA†, and Tohru KIKUNO†, Regular Members

SUMMARY The $k$-mutual exclusion problem is the problem of guaranteeing that no more than $k$ computing nodes enter a critical section simultaneously. The use of a $k$-coterie, which is a special set of node groups, is known as a robust approach to this problem. In general, $k$-coteries are classified as either dominated or nondominated, and a mutual exclusion mechanism has maximal availability when it employs a nondominated $k$-coterie. In this paper, we propose two new schemes called VOT and D-VOT for constructing nondominated $k$-coteries. We conduct a comparative evaluation of the proposed schemes and well-known previous schemes. The results clearly show the superiority of the proposed schemes.

key words: $k$-mutual exclusion, distributed systems, $k$-coteries, nondominated coteries, availability

1. Introduction

The distributed mutual exclusion problem is recognized as one of the most fundamental problems in distributed computing. The distributed $k$-mutual exclusion problem is its generalization and is the problem of guaranteeing that no more than $k$ computing nodes can enter a critical section (CS) simultaneously. The solution to this problem is useful for various applications in a distributed environment. For example, it can be used to restrict the number of broadcasting nodes for congestion control. It can be also useful in the replicated databases that allow bounded ignorance [10]. In such databases, more than one updates are allowed to proceed simultaneously for achieving high concurrency.

Several methods have been proposed for solving the $k$-mutual exclusion problem. Among them, the use of a $k$-coterie is known as a reliable approach [1], [3], [4], [7]–[9], [11], [12]. A $k$-coterie is a special set of subsets of nodes. Each element in a $k$-coterie is called a quorum. For any $k+1$ quorums in a $k$-coterie, there is always a node that is shared by at least two of the $k+1$ quorums. Each node has to gain permissions from all nodes of a quorum before it is allowed to enter the CS, and thus it is guaranteed that more than $k$ nodes never enter the CS simultaneously.

In the presence of failures, $k$-coterie-based mutual exclusion mechanisms provide fault-tolerance capability: The CS can be still entered if all nodes of a quorum are operational. It is then clear that the availability of a $k$-coterie-based mechanism depends on the $k$-coterie adopted by the mechanism.

In general, $k$-coteries are classified as either dominated or nondominated [12]. By the definition of domination, $k$-coteries provide higher availability than the $k$-coteries that they dominate. Thus nondominated $k$-coteries can be considered as a class of the most resilient $k$-coteries to failures.

So far, several constructions for $k$-coteries have been proposed [1], [3], [4], [7], [8], [11]–[13]. However most of them, including MAJ [4], [8], [13] and DIV [1], [4], can generate nondominated $k$-coteries only when certain conditions hold.

In this paper, we propose two new schemes, VOT and D-VOT, for constructing $k$-coteries. VOT is based on voting, and D-VOT is based on a composition of $1$-coteries constructed by VOT. First, we show that $k$-coteries constructed by VOT dominate those by MAJ, and $k$-coteries constructed by D-VOT dominate those by DIV (unless they are identical). Therefore the new $k$-coteries constructed by VOT and D-VOT provide higher availability than those by MAJ and DIV, respectively. Furthermore, we show that the $k$-coteries constructed by both VOT and D-VOT are always nondominated.

The remainder of this paper is organized as follows: In Sect. 2, the formal definition of a $k$-coterie is given. MAJ and DIV $k$-coteries are also described in this section. Then, the proposed scheme VOT is explained in Sect. 3. In Sect. 4, the other proposed scheme D-VOT is explained. Comparative evaluation of the new constructions with MAJ and DIV is performed in Sect. 5. Finally, a brief summary is given in Sect. 6.

2. Preliminaries

Let $V = \{v_1, \ldots, v_n\}$ denote the set of all nodes in a distributed system where $n$ is the number of nodes. We assume that each node is either operational or nonoperational, and that they are connected via a reliable network. Let $p$ be the reliability of each node (i.e., the probability that a node is operational).

Definition 1 ($k$-coterie and quorum): A nonempty...
set \( C \) of nonempty subsets \( Q \) of \( V \) is called a \( k \)-coterie under \( V \) if and only if the following three properties hold:

1. Nonintersection property: For any \( h(<k) \) elements \( Q_1, Q_2, \ldots, Q_h \in C \) such that \( Q_i \cap Q_j = \emptyset (i \neq j, 1 \leq i, j \leq h) \), there exists an element \( Q \in C \) such that \( Q \cap Q_l = \emptyset (1 \leq l \leq h) \).
2. Intersection property: For any \( k+1 \) elements \( Q_1, Q_2, \ldots, Q_{k+1} \in C \), there exists a pair \( Q_i \) and \( Q_j \) such that \( Q_i \cap Q_j \neq \emptyset (1 \leq i, j \leq k+1) \).
3. Minimality property: For any \( Q \in C \), there is no other element \( Q' \in C \) such that \( Q' \subset Q \).

Each element \( Q \) of a \( k \)-coterie \( C \) is called a quorum.

Obviously, if \( k > n \), no \( k \)-coterie exists. This case is not of interest, so we assume that \( k \leq n \) always holds throughout the paper.

To illustrate this concept, let us introduce two well-known constructions for \( k \)-coteries, MAJ and DIV.

**Definition 2** (MAJ[4],[8],[13]): Suppose \( w \leq n/k \) where \( w = \lfloor (n+1)/(k+1) \rfloor \). An MAJ \( k \)-coterie under \( V \) is a set of all subsets of \( V \) that have exactly \( \lfloor (n+1)/(k+1) \rfloor \) nodes.

**Definition 3** (DIV[1],[4]): Suppose that \( n \bmod k = 0 \) and all nodes of \( V \) are partitioned into \( k \) subsets of \( V \), \( V_1, V_2, \ldots, V_k \), such that \( |V_1| = |V_2| = \cdots = |V_k| = n/k \). Let \( C_i \) be an MAJ 1-coterie under \( V_i (1 \leq i \leq k) \). A DIV \( k \)-coterie under \( V \) is an union of \( C_i (1 \leq i \leq k) \).

**Example 1:** Let \( V = \{v_1, v_2, v_3, v_4, v_5, v_6\} \). Now consider the following sets of subsets of \( V \), \( C_1, C_2, \) and \( C_3 \). They are all \( 2 \)-coteries. Among them, \( C_2 \) is an MAJ 2-coterie. \( C_3 \) is a DIV 2-coterie when the partition of \( V \) is \( \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\} \).

\[
C_1 = \{\{v_1, v_2, v_4\}, \{v_3, v_5, v_6\}\},
C_2 = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\},
C_3 = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\}.
\]

**Example 2:** Consider \( C_1, C_2 \) and \( C_3 \) discussed in Example 1. Here \( C_2 \) dominates \( C_1 \), and \( C_3 \) dominates \( C_2 \). Furthermore one can prove that \( C_3 \) is a nondominated \( 2 \)-coterie[12].

By Definition 4, it is easy to see that \( k \)-coteries are more resilient to failures than their dominated coteries. Let \( C_j \) be a \( k \)-coterie that is dominated by another \( k \)-coterie \( C_i \). Then, for any quorum \( Q \) in \( C_j \), there is a quorum \( Q' \) in \( C_i \) such that \( Q' \subseteq Q \). Thus, if there exists a quorum in \( C_j \) such that all its nodes are operational, then there also exists such a quorum in \( C_i \). As mentioned above, the CS can be entered if all nodes of a quorum are operational. Therefore, the availability of a \( k \)-coterie is higher than or at least equal to that of any other coteries that the \( k \)-coterie dominates.

3. **Scheme VOT**

In this section, we explain one of the two new constructions and prove that the new \( k \)-coteries are always superior to MAJ \( k \)-coteries. We call this construction VOT since it uses a technique similar to weighted voting[6].

**3.1 Construction**

First, the values of three integers, \( a, b, \) and \( w \), are determined as follows: Let \( x \) be an integer such that \( (n+1+x) \bmod (k+1) = 0 \) and \( 0 \leq x \leq k \). Let \( y \) be
(n + 1 + x)/(k + 1). (1) If y is even or x < y(y + 1)/2, then let a = x, b = 0 and w = y = [(n + 1)/(k + 1)]; (2) Otherwise (that is, y is odd and x ≥ y(y + 1)/2), let a = 0, w = [(n + 1)/(k + 1)] and b be an integer such that (n + 1 − b) mod (k + 1) = 0 and 0 < b ≤ k.

Next, votes are assigned to nodes in the following way:
(1) Select any a nodes and assign each of them two votes. (2) Select any b nodes and assign each of them zero vote. (3) Assign other n − a − b nodes one vote each.

Finally, a k-coterie is formed as follows: Let W(A) denote the number of votes that the nodes in a subset A of V have. Consider a set \( C_{\text{coterie}} \) of subsets of V such that a subset V' of V is in \( C_{\text{coterie}} \) if and only if W(V') ≥ w and W(V'\) < w for any V' in V. Then, as will be shown later, \( C_{\text{coterie}} \) is a k-coterie.

Example 3: Let V = \{v₁, v₂, v₃, v₄, v₅, v₆\}. Consider the case where k = 2 (then a = 2, b = 0, and w = 3). Assume that two (a) nodes, v₁ and v₂, are assigned two votes and the other nodes are assigned one vote. Then, 2-coterie \( C₂ \) shown below is obtained. (Take Q = \{v₁, v₂\} as an example. Then, W(Q) = 4 ≥ w and there is no Q' \subset Q\ such that W(Q') > w. Thus Q ∈ \( C₁ \).) Similarly, 3-coterie \( C₃ \) is formed as shown below.

\[ C₄ = \{\{v₁, v₂\}, \{v₁, v₃\}, \{v₁, v₄\}, \{v₁, v₅\}, \{v₁, v₆\}, \{v₂, v₃\}, \{v₂, v₄\}, \{v₂, v₅\}, \{v₂, v₆\}, \{v₃, v₄\}, \{v₃, v₅\}, \{v₃, v₆\}, \{v₄, v₅\}, \{v₄, v₆\}, \{v₅, v₆\}\}. \]

\[ C₅ = \{\{v₁\}, \{v₂\}, \{v₃\}, \{v₄\}, \{v₅\}, \{v₆\}\}. \]

Theorem 1: \( C_{\text{coterie}} \) is a k-coterie.

Proof: We show that \( C_{\text{coterie}} \) satisfies the three properties of a k-coterie: the nonintersection property, the intersection property, and the minimality property.

1. Nonintersection property
There are two cases to be considered.

Case 1: There is no node assigned zero vote. (i.e. \( b = 0 \))

Case 2: There is a node assigned zero vote. (i.e. \( b > 0 \))

In Case 1, \( w = (n + 1 + a)/(k + 1) \), and then \( n = kw + (w − a − 1) \). By definition, W(Q) ≥ w for all elements Q in \( C_{\text{coterie}} \). Now assume W(Q) ≥ (w + 2) and let v be any node in Q and Q' = Q – \{v\}. Since each node in Q is assigned one or two votes, Q' \subset Q and W(Q') ≥ (w + 2) − 2 = w. This is a contradiction. Thus W(Q) = w or (w + 1) for all elements Q in \( C_{\text{coterie}} \). Furthermore, every node in Q is assigned two votes if W(Q) = w + 1. (Assume that there is a node v in Q that is assigned one vote. Then W(Q – \{v\}) = w. This is a contradiction.) Thus, if w is even, there is no Q in \( C_{\text{coterie}} \) such that W(Q) = w + 1.

Select \( h(<k) \) mutually disjoint elements \( Q₁, \ldots, Qₙ \) in \( C_{\text{coterie}} \) and let \( S₁ \) denote the union of such \( h \) elements. When w is even, each element in \( S₁ \) has w votes. Thus W(S₁) = hw. Then W(V - S₁) = (kw + w - 1) - hw ≥ w + (w - 1) ≥ w, since W(V) = kw + w - 1 and w ≥ 1. When w is odd, each of the \( h \) elements has either w or w + 1 votes. If it has w + 1 votes, all its subsets are assigned two votes, as mentioned above. Since \( S₁ \) contains no more than a nodes assigned two votes, it contains no more than \( 2a/(w+1) \) elements of \( C_{\text{coterie}} \) that have w + 1 votes. Thus W(S₁) ≤ \( 2a/(w+1)\) + (k - 1 - \( 2a/(w+1)\))w = \( 2a/(w+1)\) + (k - 1)w. Then W(V - S₁) ≥ kw + w - 1 - \( 2a/(w+1)\) + (k - 1)w = w + (w - 1 - \( 2a/(w+1)\)) ≥ w. Since, by definition, \( a < w(w+1)/2 \), W(V - S₁) ≥ w. Thus, whether w is even or odd, W(V - S₁) ≥ w. Then there is Q \subset V - S₁ such that W(Q) ≥ w. Clearly, Q \in \( C_{\text{coterie}} \) and \( Q \cap Q₁ = \emptyset \) (1 ≤ i ≤ h).

In Case 2, since w = (n + 1 + b)/(k + 1), n = kw + (w + b - 1). Select \( h(<k) \) mutually disjoint elements \( Q₁, \ldots, Qₙ \) in \( C_{\text{coterie}} \) and let \( S₁ \) denote the union of such \( h \) elements. Then W(S₁) = hw and W(V - S₁) = kw + (w - 1) - hw ≥ w. Thus there is an element Q in \( V - S₁ \) such that W(Q) ≥ w. Clearly, Q \in \( C_{\text{coterie}} \) and \( Q \cap Q₁ = \emptyset \) (1 ≤ i ≤ h).

In both Case 1 and Case 2, \( C_{\text{coterie}} \) satisfies the nonintersection property.

2. Intersection property
Assume that there are k + 1 mutually disjoint elements in \( C \). Let \( S₁ \) denote the union of such k + 1 elements. In the case where a > 0, W(S₁) ≥ w(k + 1) = n + a + 1 > n + a. This is a contradiction. In the case where a = 0, W(S₁) = w(k + 1) = n + b + 1 > n + b. (Recall that there are b nodes that are assigned zero vote.) Since this is also a contradiction, \( C_{\text{coterie}} \) satisfies the intersection property.

3. Minimality property
Assume that there is a pair of Q and Q' in \( C_{\text{coterie}} \) such that Q \subset Q'. By definition, W(Q) < w. This is a contradiction. Thus \( C_{\text{coterie}} \) satisfies the minimality property.

\[ \Box \]

3.2 Properties
Here, we prove two properties of new k-coterie \( C_{\text{coterie}} \) constructed by VOT. One is that \( C_{\text{coterie}} \) dominates MAJ k-coterie \( C_{maj} \). The other is that \( C_{\text{coterie}} \) is nondominated.

Theorem 2: If \( C_{maj} \) exists and \( C_{\text{coterie}} \neq C_{maj} \), then \( C_{\text{coterie}} \) dominates \( C_{maj} \).

Proof: As mentioned above, \( C_{maj} \) is a set of all y-subsets of V where y = \( [(n + 1)/(k + 1)] \). Recall that \( C_{maj} \) exists only when n ≥ yk. Let x be an integer such that \( (n + 1 + x)/(k + 1) = [(n + 1)/(k + 1)] \). Then n = yk + (y - x - 1). Thus y - x - 1 ≥ 0 if n ≥ yk. If y - x - 1 ≥ 0, then x < y(y + 1)/2. Thus, when
Each of the k-majorities exists, a ≥ 0, b = 0 and w = [(n + 1)/(k + 1)] in VOT, which means that each node is assigned one or two votes in VOT.

Let Q be a quorum in Cmaj. First, consider the case where every node in Q is assigned one vote. In this case, Q is also in Crot since W(Q) = w. Next, consider the case where there is at least one node assigned two votes in Q. Since W(Q) > w in this case, Q is also in Crot if there is no Q′(⊂ Q) such that W(Q′) > w; Otherwise (that is, there is Q′(⊂ Q) such that W(Q′) > w), Q′ is in Crot. Consequently, for all Q ∈ Cmaj, there exists Q′ ∈ Crot such that Q′ ⊆ Q. Thus Crot dominates Cmaj unless Crot = Cmaj.

Theorem 2 means that new k-coterie Crot dominates MAJ k-coterie Cmaj if (n+1) mod (k+1) ≠ 0. (Note that Crot is exactly equal to Cmaj if (n+1) mod (k+1)=0.) As mentioned above, a k-coterie provides higher availability than its dominated coteries. Thus it is guaranteed that the availability of Crot is always higher than or at least equal to that of Cmaj.

Theorem 3: Crot is a nondominated k-coterie.

Proof: We prove this theorem by contradiction. Assume that Crot is a dominated k-coterie. Then, because of the theorem of Neilsen and Mizuno [12], there exists a subset H of V that satisfies the following two properties: (P1) for any quorum Q ∈ Crot, Q ∩ H, and (P2) for any k mutually disjoint quorums Q1, Q2, · · · , Qk, there is Qi (1 ≤ i ≤ k) such that H ∩ Qi ≠ ∅. In the following, we show that V − H contains k mutually disjoint quorums as its subsets, thus contradicting to (P2).

Due to (P1), W(H) < w. Since W(V) = kw+w−1, W(V − H) = W(V) − W(H) ≥ kw. When a = 0, that is, each node is assigned one or zero vote, V − H has at least kw nodes that have one vote. Hence it is clear that V − H contains at least k mutually disjoint quorums.

When a > 0, every node has one or two votes. In this case, if w is even, then at least k mutually disjoint subsets of V − H exist that have exactly w votes.

Finally, consider the case where a > 0 and w is odd. In this case, w ≥ 3 since we assume n ≥ k. Let m denote the number of nodes in H that have one vote, and let m′ denote the number of nodes in V − H that have one vote. Since a ≤ k and m < w, m′ = n − a − m = (kw+w−a−1) − (m ≥ k(w−2) ≥ k). Hence there is a subset T of V − H that consists of exactly k nodes that have one vote. Let S = V − (H ∪ T). Since W(S) = W(V − H) − W(T) ≥ k(w−1) and w−1 is even, there are at least k subsets of S each of which has a total of w − 1 votes. By adding each node in T to each of the k subsets, k mutually disjoint quorums can be obtained. Thus the following holds:

4. Scheme D-VOT (Division Strategy Using VOT)

The other proposed scheme D-VOT is based on VOT and a division strategy used in DIV. As mentioned above, the strategy in DIV partitions all nodes of V into k clusters so as to construct a k-coterie by combining 1-coteries in the clusters. However, since it partitions V equally, its applicability is limited to the cases where n mod k = 0. In our scheme, D-VOT, the division strategy is extended to an arbitrary number of nodes.

4.1 Construction

Let R be the remainder from integer division of n by k. First, V is partitioned into k clusters (nonempty sets of nodes) V1, V2, · · · , Vk with R of clusters containing ⌊n/k⌋ nodes and k−R clusters containing ⌈n/k⌉ nodes. Next, let Crot = C1 ∪ C2 ∪ · · · ∪ Ck where Ci is the 1-coterie under Vi (1 ≤ i ≤ k) that is constructed by scheme VOT\(^1\). Then, Crot is a k-coterie. (We omit the proof since it is trivial.) Note that VOT does not assign zero vote to any nodes in V if k = 1\(^1\), and thus every node is contained in at least one quorum in Crot.

Example 4: Let V = \{v1, v2, v3, v4, v5, v6, v7\} and k = 2. Then R = 1. Suppose that the partition of V is \{\{v1, v2, v3\} (= V1), \{v4, v5, v6, v7\} (= V2)\}, and let C′ and C″ be 1-coteries under V1 and V2 constructed by VOT, respectively. Then, 2-coterie C6 shown below is obtained.

C6 = C′(= \{\{v1, v2\}, \{v1, v3\}, \{v2, v3\}\}) ∪ C″(= \{\{v4, v5\}, \{v4, v6\}, \{v4, v7\}, \{v5, v6, v7\}\}).

4.2 Properties

Let Crot be a k-coterie constructed by D-VOT and let Cdiv be a DIV k-coterie under the same V and partition. Here, we prove two properties of the new k-coterie Crot. One is that Crot dominates Cdiv. The other is that Crot is nondominated.

Theorem 4: If Crot = Cdiv, then Crot dominates Cdiv.

Proof: Let \{V1, V2, · · · , Vr\} be a partition of V into k clusters. By definition, Crot under V is D1 ∪ D2 ∪ · · · ∪ Dr where Di is the MAJ 1-coterie under Vi (1 ≤ i ≤ k). Crot under V is C1 ∪ C2 ∪ · · · ∪ Ck where Ci is the 1-coterie under Vi (1 ≤ i ≤ k) that is constructed by scheme VOT. By Theorem 2, Ci dominates Di if Ci ≠ Ci.

\(^1\)When k = 1, VOT is equivalent to a 1-coterie construction proposed in [5].

\(^2\)Note that, in VOT, a = (0 or 1), b = 0, and w = [(n + 1)/2] if k = 1.
Theorem 4 means that $C_{dVOT}$ dominates $C_{div}$ if $(n/k + 1) \mod 2 \neq 0$. (Note that $C_{div}$ exists only when $n \mod k = 0$, and $C_{dVOT}$ is equal to $C_{div}$ if $(n/k + 1) \mod 2 = 0$.) Thus it is guaranteed that the availability of $C_{dVOT}$ is always higher than or at least equal to that of $C_{div}$.

To prove $C_{dVOT}$ is a nondominated k-coterie, we first introduce a theorem given in [12].

**Theorem 5:** Let $\{V_1, V_2, \ldots, V_k\}$ be a partition of $V$ into $k$ nonempty sets. Let $C = C_1 \cup C_2 \cup \cdots \cup C_k$, and $C_i$ be a nondominated coterie under $V_i(1 \leq i \leq k)$. Then $C$ is a nondominated $k$-coterie under $V$.

**Proof:** By definition, $C_{dVOT} = C_1 \cup C_2 \cup \cdots \cup C_k$ where $C_i$ is the 1-coterie under $V_i(1 \leq i \leq k)$ that is constructed by scheme VOT. By Theorem 3, $C_i(1 \leq i \leq k)$ is a nondominated coterie. Thus, by Theorem 5, $C_{dVOT}$ is nondominated.

5. Experimental Evaluation

In Sects. 3 and 4, we proved that the new constructions VOT and D-VOT provide higher level of availability than MAJ and DIV, respectively. Here, for the purpose of quantitative analysis, we perform experimental evaluation of the four constructions, VOT, D-VOT, MAJ and DIV.

As a measure for evaluation, we consider $(k, r)$-availability. As mentioned above, $(k, r)$-availability is a generic measure to evaluate the availability of a $k$-coterie, and it is defined as the probability that at least $r(1 \leq r \leq k)$ nodes can enter the CS simultaneously.

Table 1 shows the $(k, r)$-availabilities of $k$-coteriers constructed by VOT, D-VOT, MAJ, and DIV when $14 \leq n \leq 17$, $2 \leq k \leq 4$, and $p = 0.9$. First, consider VOT and MAJ. When $(n, k) = (15, 4)$, no MAJ
A new construction for non-dominated \( k \)-coterie is presented. When \((n,k) = (14,2), (14,4), (15,3), \) and \((17,2)\), VOT is equivalent to MAJ since \((n+1) \mod (k+1) = 0\). If \((n+1) \mod (k+1) \neq 0\), the \((k,r)\)-availability of a VOT \( k \)-coterie is higher than that of an MAJ \( k \)-coterie. Especially, as the value of \( r \) increases, the difference of the availabilities between VOT and MAJ becomes larger. For example, when \((n,k,r) = (16,4,4)\), the difference of \((k,r)\)-availabilities is larger than 0.6. Next, consider D-VOT and DIV. DIV \( k \)-coteries exist only when \((n,k) = (14,2), (16,2), (15,3) \) and \((16,4)\). When \((n,k) = (14,2) \) and \((15,3)\), D-VOT is equivalent to DIV since \((n+1) \mod 2 = 0\). In other cases, D-VOT outperforms DIV in \((k,r)\)-availability. Like the relations between VOT and MAJ, as the value of \( r \) increases, the difference of the availabilities between D-VOT and DIV becomes larger. As a result, when \( r \) is small, VOT provides the highest \((k,r)\)-availability among the four constructions, whereas VOT achieves the highest \((k,r)\)-availability when \( r \) is large.

Although \((k,r)\)-availability is a useful and well accepted measure, it may be difficult to compare \( k \)-coteries based on this measure only, since \( r \) can take any value up to \( k \). For that reason, we introduce \( \text{computation availability}[2] \) as a unified measure for evaluation. Computation availability has been used for evaluating gracefully degradable systems in terms of both availability and performance. Here we define the computation availability of a \( k \)-coterie as follows:

\[
\text{Computation availability} = \sum_{r=1}^{k} \alpha_r \times r\text{-availability}
\]

where \( \alpha_r \) denotes the computation capability when \( r \) nodes can enter the CS simultaneously, and \( r\text{-availability} \) denotes the probability that the maximum number of nodes that can enter the CS simultaneously is exactly \( r \)[8]. Assuming that \( \alpha_r \) is proportional to \( r \), we set \( \alpha_r \) to \( \frac{r}{k} \) in the evaluation.

Figure 1 shows the computation availabilities of 4-coteries constructed by VOT, D-VOT, MAJ, and DIV when \( k=4 \).

\( k \)-coterie exists. When \((n,k) = (14,2), (14,4), (15,3), \) and \((17,2)\), VOT is equivalent to MAJ since \((n+1) \mod (k+1) = 0\). If \((n+1) \mod (k+1) \neq 0\), the \((k,r)\)-availability of a VOT \( k \)-coterie is higher than that of an MAJ \( k \)-coterie. Especially, as the value of \( r \) increases, the difference of the availabilities between VOT and MAJ becomes larger. For example, when \((n,k,r) = (16,4,4)\), the difference of \((k,r)\)-availabilities is larger than 0.6. Next, consider D-VOT and DIV. DIV \( k \)-coteries exist only when \((n,k) = (14,2), (16,2), (15,3) \) and \((16,4)\). When \((n,k) = (14,2) \) and \((15,3)\), D-VOT is equivalent to DIV since \((n+1) \mod 2 = 0\). In other cases, D-VOT outperforms DIV in \((k,r)\)-availability. Like the relations between VOT and MAJ, as the value of \( r \) increases, the difference of the availabilities between D-VOT and DIV becomes larger. As a result, when \( r \) is small, VOT provides the highest \((k,r)\)-availability among the four constructions, whereas VOT achieves the highest \((k,r)\)-availability when \( r \) is large.

Although \((k,r)\)-availability is a useful and well accepted measure, it may be difficult to compare \( k \)-coteries based on this measure only, since \( r \) can take any value up to \( k \). For that reason, we introduce computation availability[2] as a unified measure for evaluation. Computation availability has been used for evaluating gracefully degradable systems in terms of both availability and performance. Here we define the computation availability of a \( k \)-coterie as follows:

\[
\text{Computation availability} = \sum_{r=1}^{k} \alpha_r \times r\text{-availability}
\]

where \( \alpha_r \) denotes the computation capability when \( r \) nodes can enter the CS simultaneously, and \( r\text{-availability} \) denotes the probability that the maximum number of nodes that can enter the CS simultaneously is exactly \( r \)[8]. Assuming that \( \alpha_r \) is proportional to \( r \), we set \( \alpha_r \) to \( \frac{r}{k} \) in the evaluation.

Figure 1 shows the computation availabilities of 4-coteries constructed by VOT, D-VOT, MAJ, and DIV. Figures 1 (a1) and (a2) show the results when \( 4 \leq n \leq 30 \), and \( p = 0.9 \) and 0.95, respectively. As the number of nodes increases, the \((k,r)\)-availabilities of VOT and D-VOT become steady and larger. The results thus show the scalability that the proposed schemes exhibit. In the

\( r\text{-Availability} \) is easily obtained from the \((k,r)\)-availabilities. If \( r=k \), then \( r\text{-availability} = (k,r)\text{-availability} \); otherwise, \( r\text{-availability} = (k,r)\text{-availability} - (k,r+1)\text{-availability} \).
case where $p = 0.9$, VOT provides the highest computation availability when $4 \leq n \leq 10$, and D-VOT provides the highest computation availability when $11 \leq n \leq 30$. In the case where $p = 0.95$, VOT provides the highest computation availability when $4 \leq n \leq 11$, while D-VOT has the highest computation availability when $12 \leq n \leq 30$. Figures 1 (b1) and (b2) show the results when $0 \leq p \leq 1$, and $n = 16$ and 40, respectively. We selected the two values of $n$ because the four schemes construct different $k$-coteries when $n$ has these values\(^1\). In both cases, when $0 < p < 0.5$, VOT provides the highest computation availability, while when $0.5 < p < 1$, D-VOT provides the highest computation availability. Interestingly, the computation availability of VOT and D-VOT is exactly equal to $0.5$ when $p = 0.5$.

6. Conclusion

In this paper, we have proposed two new schemes VOT and D-VOT of constructing $k$-coteries. In scheme VOT, $k$-coteries are constructed by using weighted voting. In scheme D-VOT, $k$-coteries are constructed by using the partition of nodes and scheme VOT. We have shown that the $k$-coteries constructed by VOT and D-VOT dominate those by MAJ and DIV, respectively. We have also shown that the $k$-coteries constructed by both VOT and D-VOT are nondominated, which means that the new $k$-coteries provide higher availability than any of their dominated coteries. Furthermore, we have evaluated the four schemes in terms of $(k, r)$-availabilities and the computation availabilities. The superiority of the new $k$-coteries has been also shown by the results.

Acknowledgements

The authors would like to thank anonymous referees for their helpful suggestions.

References


---

\(^1\) Consider other values of $n$. For example, when $n = 20$, D-VOT is equivalent to DIV, and when $n = 24$, VOT is equivalent to MAJ.