Constructing Nondominated Local Coteries for Distributed Resource Allocation

Jehn-Ruey Jiang
Department of Information Management
Hsuan-Chuang University, Taiwan
E-mail: jrjiang@hcu.edu.tw

Cheng-Sheng Chou
Department of Computer Science
National Tsing-Hua University, Taiwan
E-mail: g894371@oz.nthu.edu.tw

Shing-Tsaan Huang
Department of Computer Science and Information Engineering
National Central University, Taiwan
E-mail: sthuang@csie.ncu.edu.tw

Abstract

The resource allocation problem is a fundamental problem in distributed systems. In this paper, we focus on constructing *nondominated* (*ND*) *local coteries* to solve the problem. Distributed algorithms using coteries usually incur low communication overhead and have high degree of fault-tolerance, and ND coteries are candidates for the algorithms to achieve the highest degree of fault-tolerance. We define a new type of coteries, called *p-coteries*, to aid the construction of local coteries. We then develop theorems about the nondomination of p-coteries, and propose an operation, called *pairwise-union* (*p-union*), to help generate ND p-coteries from known ND coteries. ND p-coteries can then be used to generate ND local coteries for solving the distributed resource allocation problem.

1. Introduction

The resource allocation problem is a fundamental problem in distributed systems. Consider a distributed system consisting of a set of processes and a set of distinct resources. The processes can communicate with each other by exchanging messages, and from time to time, a process may request to enter the critical section (*CS*) to access some of the resources. Before entering the *CS*, a process has to wait until all the desired resources are acquired. The resource allocation problem is concerned with how to ensure that all resources are accessed in a mutually-exclusive way and that all processes wishing to enter the *CS* can proceed in finite time.

There are many problems related to the resource allocation problem: the mutual exclusion problem [7], the *k*-mutual exclusion problem [9], the *h*-out of-*k* mutual exclusion problem [25], the dining philosophers problem [8] and the drinking philosophers problem [4]. The mutual exclusion problem deals with the mutually-exclusive sharing of a unique resource among all processes. The *k*-mutual exclusion problem deals with the sharing of *k* identical resources with the restriction that one process can access any one resource at a time. The *h*-out of-*k* mutual exclusion problem deals with the sharing of *k* identical resources with the restriction that one process can access any *h*, $h \le k$, resources at a time. The dining philosophers problem and the drinking philosophers problem describe the resource sharing relation by conflict graph, in which a vertex represents a process and an edge represents the resource shared by the two processes incident to the edge. In the dining philosophers problem, a process can enter the *CS* when it has acquired all the resources represented by the edges incident to it; while in the drinking philosophers problem, a process can enter the *CS* when it has acquired a subset of the resources.

There have been solutions [3, 5, 6, 20, 26] proposed for solving the distributed resource allocation problem. Among them, solutions in [6, 20] utilize a special structure called *local coterie* to solve the problem. The local coterie is one of the extensions of the *coterie* proposed in [10]. A coterie is a collection of mutually disjoint minimal sets, each of which is called a *quorum*. Except the distributed resource allocation problem, the coterie and its extensions are applied to solve many other problems. For example, the coterie is used to solve the mutual exclusion problem [1, 19, 15], and the *k*-coterie, another extension of the coterie, is used to solve the *k*-mutual exclusion problem [11, 13, 14, 17] and the *h*-out of-*k* mutual exclusion problem [16].

The solutions using coterie structures usually incur low communication overhead and can tolerate process and/or communication link failures. Among coterie structures, *nondominated* (ND) coterie structures are candidates for the solutions to achieve the highest degree of fault-tolerance. Thus, we should always concentrate on ND coterie structures if fault-tolerance is significant. There are many researches investigating ND coterie structures; for example, researches in [10, 15, 18] study ND coteries and researches in [12, 13, 22, 23] study ND *k*-coteries.

In this paper, we concentrate on constructing ND local coteries to solve the distributed resource allocation problem. We define a new type of coteries, called *p-coteries*, to aid the construction of local coteries. We then develop theorems about the nondomination of p-coteries, and propose an operation, called *pairwise-union* (*p-union*), to help generate ND p-coteries from known ND coteries, such as the majority coteries [27], the tree coteries [1], the hierarchical coteries [19], the Lovasz coteries [21], the crumbling walls coteries [24] and the cohorts coteries [15], etc. ND p-coteries can then be used to construct ND local coteries for solving the distributed resource allocation problem.

The rest of this paper is organized as follows. In Section 2, we elaborate some preliminaries of the distributed resource allocation problem, including coteries and local coteries. In Section 3, we propose the definition of p-coteries and develop theorems for checking their nondomination. We also show that the p-union operation can help generate ND p-coteries for the construction of ND local coteries. And finally, we conclude this paper in Section 4.

2. Preliminaries

2.1. Distributed Resource Allocation

Consider a distributed system consisting of a set P of processes and a set R of shared resources each of which is of a different type and must be accessed in a mutually exclusive way. Occasionally, processes may request to enter the critical section (*CS*) to access some of the resources. A process p_i, $p_i \in P$, enters the *CS* after it acquires all the requested resources. Afterwards, p_i leaves the *CS* and releases all the acquired resources. Processes are assumed to leave the *CS* in finite time. The resource allocation problem is concerned with how to ensure that all resources are accessed in a mutually exclusive way and that all processes wishing to enter the *CS* can proceed in a finite time.

The process-accessing-resource relation in the resource allocation problem can be represented by a *resource allocation graph* (*RAG*). A *RAG* for the system with process set P and resource set R is a bipartite graph $G=(V, E)$, where $V=P\cup R$ is a set of vertices and E is a set of edges. There is an edge $e=(p, r)\in E$ if and only if process p requests to access resource r. Let $R_i = \{r \mid$ process p_i requests to access resource $r\}$ be the set of all the resources that process p_i requests to access. If $R_i \cap R_j \neq \emptyset$, it means that process p_i and process p_j compete for the same resources.

Figure-1 is an example of RAG for the system with $P=\{p_1, p_2, p_3\}$ and $R=\{r_1, r_2\}$. With respect to the RAG, $R_1=\{r_1\}$, $R_2=\{r_1, r_2\}$ and $R_3=\{r_2\}$, which means that process p_1 requests to access resource r_1, process p_2 requests to access resources r_1 and r_2, and process p_3 requests to access resource r_2.

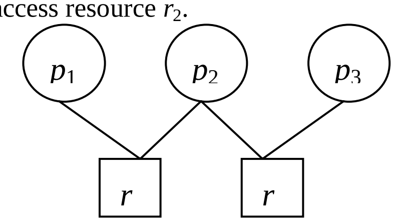

Figure 1. The resource allocation graph (RAG) for the system with $P=\{p_1, p_2, p_3\}$ and $R=\{r_1, r_2\}$.

In [20], Kakugawa and Yamashita introduced the concept of *local coteries* and proposed an algorithm using local coteries to solve the distributed resource allocation problem. In [6], Cheng et al. proposed another algorithm using local coteries to solve the distributed resource allocation problem. Local coteries are the extensions of coteries. Below, we introduce the concept of coteries first. And then we introduce the concept of local coteries.

2.2. Coteries

A *coterie* C under P is a family of subsets of P. Each member in C is called a *quorum* and should observe the following two properties [10]:

Intersection Property: $\forall q_1, q_2 : q_1, q_2 \in C : q_1 \cap q_2 \neq \varnothing$

Minimality Property: $\forall q_1, q_2 : q_1, q_2 \in C : q_1 \not\subset q_2$

For example, $C = \{\{p_1, p_2\}, \{p_1, p_3\}, \{p_2, p_3\}\}$ is a coterie under $P=\{p_1, p_2, p_3\}$ because every pair of quorums (members) in C have a non-empty intersection, and no quorum is a super set of another quorum.

2.3. Local Coteries

Given a RAG of the system with a set P of processes and a set R of resources. A local coterie $LC=(C_1,\ldots,C_{|P|})$ is a list[1] of coteries under P. There is a coterie C_i associated with each process $p_i \in P$, $1 \leq i \leq |P|$ and all of the following conditions should hold [20]:

Non-emptiness Property: $\forall p_i : p_i \in P : C_i \neq \varnothing$

Intersection Property: If $R_i \cap R_j \neq \varnothing$, then $\forall q_1, q_2$: $q_1 \in C_i$, $q_2 \in C_j$: $q_1 \cap q_2 \neq \varnothing$, where $R_i = \{r \mid \text{process } p_i \text{ requests to access resource } r\}$.

Minimality Property: $\forall p_i, q_1, q_2 : p_i \in P, q_1, q_2 \in C_i : q_1 \not\subset q_2$

For example, LC=($\{\{p_1\}\}$, $\{\{p_1, p_3\}\}$, $\{\{p_3\}\}$) is a local coterie for the RAG in Figure-1. The reader can check that there is a coterie associated with every process (for example, $C_1=\{\{p_1\}\}$ for process p_1, $C_2=\{\{p_1, p_3\}\}$ for process p_2 and $C_3=\{\{p_3\}\}$ for process p_3) and every quorum in C_2 intersects with every quorum in C_1 (resp. C_3) because p_2 and p_1 (resp. p_3) compete for the same resource r_1 (resp. r_2).

The local coterie can be used to develop algorithms solving the distributed resource allocation problem. To enter the critical section, a process is required to form a quorum, that is, to receive the permissions from all the processes of some quorum of its associated coterie. If we restrict that every process can grant its permission to only one process at a time, then the mutually exclusive access of resources is guaranteed because any two quorums q_1 and q_2, $q_1 \in C_i$ and $q_2 \in C_j$, must intersect when p_i and p_j compete for the same resources. The reader should note that the minimality property is not necessary for the correctness of resource allocation but is used to enhance efficiency.

Kakugawa and Yamashita proposed an algorithm [20] to construct local coteries. In the algorithm, for each process p_i, its associated coterie C_i is $\{q_i\}$, where $q_i=\{p_j | R_i \cap R_j \neq \varnothing\}$ (i.e., C_i has only one quorum containing all the processes competing resources with p_i). For example, with respect to the RAG in Figure-1, a local coterie constructed by the Kakugawa and Yamashita's algorithm is ($\{\{p_1, p_2\}\}$, $\{\{p_1, p_2, p_3\}\}$, $\{\{p_2, p_3\}\}$). The local coterie is not so efficient since each C_i has only one quorum. It also has the drawback that it prohibits the concurrent *CS* entrances of the processes not competing for the same resources. For example, for the RAG in Figure-1, process p_1 and process p_3 should be able to enter the *CS* concurrently since they use no common resource. However, when we apply the local coterie ($\{\{p_1, p_2\}\}$, $\{\{p_1, p_2, p_3\}\}$, $\{\{p_2, p_3\}\}$) constructed by the Kakugawa and Yamashita's algorithm to solve the distributed resource allocation problem, process p_1 and p_3 are not allowed to enter the *CS* concurrently.

Cheng et al. proposed another algorithm [6] to construct local coteries, which are more efficient than the ones constructed by the Kakugawa and Yamashita's algorithm and can allow no-competing processes to enter the *CS* concurrently. Below, we describe the Cheng et al.'s algorithm briefly. For each resource r_j, the algorithm first finds out $P_j=\{p|$ process p accesses

[1] A local coterie is defined to be a "*set*" of coteries in paper [20]. Since the order of the coteries makes sense, we modify the definition of the local coterie to be a "*list*" of coteries.

resource r_j}, the set of all processes that access resource r_j. Then, for each resource r_j, the algorithm constructs a coterie Cr_j under P_j (note that in this paper, we use the term "*the coterie for resource* r_j" to refer to Cr_j). Afterwards, for each process p_i, a set Q_i of quorums is derived, where

$$Q_i=\{q \mid q=\bigcup_{j=1}^{m} q_j \text{ , } q_j \in Cr_j \text{ and } r_j \in R_i\}.$$

To be more precise, if process p_i accesses resources $r_1,\ldots,r_m$, $m>1$, then each member q of Q_i is of the form $q=q_1\cup\ldots\cup q_m$, where $q_1\in Cr_1,\ldots, q_m\in Cr_m$. At last, the coterie C_i associated with p_i is derived by removing every non-minimal quorum of Q_i (note that a quorum is non-minimal if it is a superset of another quorum).

We observe that the Cheng et al.'s algorithm can be improved. For example, for the RAG in Figure-2, below is a possible local coterie construction by the Cheng et al.'s algorithm:

$P=\{p_1, p_2, p_3, p_4, p_5\}$.
$R=\{r_1, r_2\}$.
$R_1=R_2=R_3= R_4=R_5=\{r_1, r_2\}$.
$Cr_1=\{\{p_1, p_2\}, \{p_1, p_3\}, \{p_1, p_4\}, \{p_1, p_5\}, \{p_2, p_3, p_4, p_5\}\}$.
$Cr_2=\{\{p_1, p_2, p_4\}, \{p_1, p_2, p_5\}, \{p_1, p_3, p_4\}, \{p_1, p_3, p_5\}, \{p_2, p_3, p_4\}, \{p_2, p_3, p_5\}, \{p_4, p_5\}\}$.
$C_1=C_2=C_3=C_4=C_5=\{\{p_1, p_2, p_4\}, \{p_1, p_2, p_5\}, \{p_1, p_3, p_4\}, \{p_1, p_3, p_5\}, \{p_1, p_4, p_5\}, \{p_2, p_3, p_4, p_5\}\}$.

Since resources r_1 and r_2 are accessed by the same set and only the same set of processes, we can regard them as a virtual resource r_3. For the virtual resource r_3, we can derive Cr_3 by letting $Cr_3=Cr_1$ or $Cr_3=Cr_2$. Thus, we have $C_1=C_2=C_3=C_4=C_5=Cr_3=Cr_1$ or $C_1=C_2=C_3=C_4=C_5=Cr_3=Cr_2$. We can check that in either case, the corresponding local coterie is better than the original one. Thus, we can conclude that if some resources are accessed by the same set and only the same set of processes, then we should regard those resources as one virtual resource. Note that we will refer to the concept just mentioned the "*virtual resource*" concept.

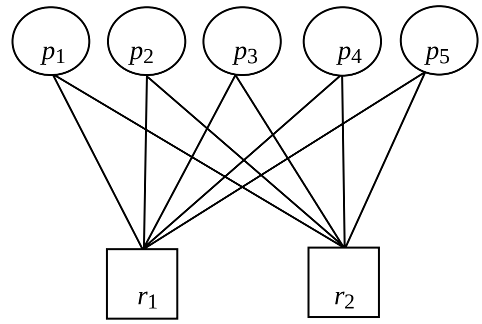

Figure 2. The RAG for a special case of the system in which some resources are accessed by the same set and only the same set of processes.

In addition to the improvement by the virtual resource concept, we also find that there may be another improvement for the Cheng et al.'s algorithm. The improvement is based on the concept of nondominated (ND) coteries. A coterie is always better than the coterie it dominates in the sense that if a quorum can be formed in the dominated one then a quorum can be formed in the dominating one. Thus, we should always concentrate on the nondominated (ND) coteries that no coterie can dominate.

In the next section, we introduce the coterie domination concept. We define a new type of coteries, called *p-coteries*, to aid the construction of local coteries. We then develop theorems about the nondomination of p-coteries, and propose an operation, called *pairwise-union* (*p-union*), to help generate ND p-coteries from known ND coteries. ND p-coteries can then be used to generate ND local coteries for solving the distributed resource allocation problem.

3. Theorems for Local Coterie Domination

In this section, we give definition of the nondominated local coteries and develop theorems about the nondomination of local coteries. We first introduce the concept of coterie domination.

Definition 1. (coterie domination) [10]
Let C and D be two distinct coteries. C is said to *dominate* D iff $\forall q, \exists q'$: $q\in D$, $q'\in C$: $q'\subseteq q$. (We say that q' is the set that dominates q.)

For example, coterie $C=\{\{p_1, p_2\}, \{p_1, p_3\}, \{p_1, p_4\}, \{p_2, p_3, p_4\}\}$ dominates coterie $D=\{\{p_1, p_2, p_3\}, \{p_1, p_2,$

p_4}, {p_1, p_3, p_4}, {p_2, p_3, p_4}} because for every quorum q in D we can find a quorum q' in C such that q is a super set of q'. A dominating coterie, such as C, is always better than a dominated coterie, such as D, since if a quorum can be formed in the dominated one then a quorum can be formed in the dominating one. A coterie is *nondominated* (ND) if no other coterie can dominate it. ND coteries are candidates to achieve the highest availability, which is the probability that a quorum can be formed in an error-prone environment. Thus, we should always concentrate on ND coteries if fault-tolerance is one of the main concerns. Some classes of coteries, such as the majority coteries [27], the tree coteries [1], the hierarchical coteries [19], the Lovasz coteries [21], the crumbling walls coteries [24] and the cohorts coteries [15] have been shown to be ND.

Theorem 1 in the following is developed by Garcia-Molina and Barbara in [10]. This theorem is useful to check if a coterie is dominated or not.

Theorem 1. Let C be a coterie under P. Then, C is dominated iff there exists a set $x \subseteq P$ such that

L1. $\forall q : q \in C: q \not\subseteq x$.

L2. $\forall q : q \in C: q \cap x \neq \emptyset$.

Following the definition of coterie domination, we give the definition of local coterie domination below.

Definition 2. (local coterie domination)

Let $C=(C_1,\ldots,C_n)$ and $D=(C_1',\ldots,C_n')$ be two distinct local coteries. C is said to *dominate D* iff $C_i=C_i'$ or C_i dominates C_i', for $1\leq i\leq n$ (i.e., every coterie in C equals or dominates its corresponding coterie in D).

For example, let local coterie C be ({{p_1}}, {{p_1, p_3}}, {{p_3}}) and local coterie D be ({{p_1, p_2}}, {{p_1, p_2, p_3}}, {{p_3}}). We can see that C dominates D since {{p_1}} dominates {{p_1, p_2}}, {{p_1, p_3}} dominates {{p_1, p_2, p_3}}, and {{p_3}} equals {{p_3}}.

By Definition 2, the domination of two distinct local coteries is based on the domination (or equality) of each pair of corresponding coteries. When a process accesses only one resource, we can apply Theorem 1 to check the domination of the coterie associated with the process since the coterie is exactly the same as defined in [10]. However, when a process accesses more than one resource, the coterie associated with the process has inter-coterie quorum intersection relation with other coteries. Below, we define a new type of coteries, called *p-coteries*, to capture the inter-coterie quorum intersection relation.

Definition 3. (p-coterie)

Given m, m>1, coteries $Cr_1,\ldots,Cr_m$, a *p-coterie C* from $Cr_1,\ldots,Cr_m$ is defined to be a coterie satisfying $\forall q \forall q' \forall j$: $q\in C$, $q'\in Cr_j$, $1\leq j\leq m$: $q\cap q'\neq\emptyset$.

In the Cheng et al.'s algorithm, if coteries $Cr_1,\ldots,Cr_m$ are selected respectively to be the coteries for resources $r_1,\ldots,r_m$, then we can easily check that a p-coterie from $Cr_1,\ldots,Cr_m$ can be used as a coterie associated with the process accessing resources $r_1,\ldots,r_m$.

With the inference similar to that in [10] for Theorem 1, we have the following theorem for checking the domination of a p-coterie. Note that by definition a p-coterie is also a coterie, which is a fact used in the proof of Theorem 2.

Theorem 2. Let C be a p-coterie from coteries $Cr_1,\ldots,Cr_m$, m>1. C is dominated if and only if there exists a set x such that

L1. $\forall q$: $q\in C$: $q\not\subseteq x$

L2. $\forall q$: $q\in C$: $q\cap x\neq\emptyset$

L3. $\forall q \forall j$: $q\in Cr_j$, $1\leq j\leq m$: $q\cap x\neq\emptyset$

Proof :

(if part)

We first show that L1, L2 and L3 imply C is dominated. There are two cases to consider. Case 1: If there are one or more $q_1,\ldots,q_l\in C$ such that $x\subset q_1,\ldots,x\subset q_l$, then construct set $S=(C-q_1-\ldots-q_l)\cup\{x\}$. It is easy to see that S is a p-coterie from $Cr_1,\ldots,Cr_m$ and S dominates C. Case 2: If there are no supersets of x in C, then $S=C\cup\{x\}$ is a p-coteries from $Cr_1,\ldots,Cr_m$ and S dominates C.

(only if part)

Now, assume that C is dominated by D, we show that conditions L1, L2 and L3 hold by considering two cases. Case 1: $C\subset D$. Let x be one of the elements in $D-C$. Set x must satisfy conditions L1, L2 and L3 or else D would not be a valid p-coterie from $Cr_1,\ldots,Cr_m$. Case 2: $C\not\subset D$. In such a case, there must be a set $q\in C$ and a set $x\in D$ such that $x\subset q$ (see Definition 1). If condition L1 is false for x, then $q'\subseteq x$ for some $q'\in C$ and C is not a coterie because $q' \subseteq x \subset q$. Similarly, if condition L2 doesn't hold for x, then D would not be a coterie because $\neg$L2 implies $\exists q'$: $q'\in C$: $q'\cap x=\varnothing$, which in turn implies $x\cap x'=\varnothing$, where x' is the set in D that dominates q'. If condition L3 doesn't hold for x, then D would not be a p-coterie from $Cr_1,\ldots,Cr_m$ because $\neg$L3 implies $\exists q'$: $q'\in Cr_j$: $q'\cap x=\varnothing$ for some Cr_j, $1\leq j\leq m$, which in turn implies $x\cap x'=\varnothing$, where x' is the set in D that dominates q'. We can see that either in case 1 or in case 2, the conditions L1, L2 and L3 should hold. ■

Inspired by the Cheng et al.'s algorithm, we propose an operation, denoted by ⊗ and called *pairwise-union* (*p-union*, for short), to generate p-coteries from coteries. As will be shown later, we can apply p-union operation on ND coteries to generate ND p-coteries for the construction of ND local coteries.

Definition 4. (pairwise-union operation)

Let P_1 and P_2 be two non-empty sets of processes. Also let G be a coterie under P_1, and H be a coterie under P_2. The *pairwise-union* (*p-union*) operation ⊗ of G and H is defined to be

$G\otimes H=\{g\cup h \mid g\in G, h\in H\}$.

For example, let $G=\{\{p_1, p_2\},\{p_2, p_3\},\{p_1, p_3\}\}$ be a coterie under $P_1=\{p_1, p_2, p_3\}$ and $H=\{\{p_2, p_3\},\{p_3, p_4\},\{p_2, p_4\}\}$ be a coterie under $P_2=\{p_2, p_3, p_4\}$. Then $G\otimes H=\{ \{p_1, p_2, p_3\}, \{p_1, p_2, p_3, p_4\}, \{p_1, p_2, p_4\}, \{p_2, p_3\}, \{p_2, p_3, p_4\}, \{p_2, p_3, p_4\}, \{p_1, p_2, p_3\}, \{p_1, p_3, p_4\}, \{p_1, p_2, p_3, p_4\} \}$.

Let $F=Min(G\otimes H)$, where $Min(Q)$ is a function to eliminate non-minimal quorums from a collection Q of quorums. The following Theorem 3, Theorem 4 and Theorem 5 are about properties of F.

Theorem 3. Let P_1 and P_2 be two non-empty sets of processes. If G is a coterie under P_1 and H is a coterie under P_2, then $F=Min(G\otimes H)$ is a p-coterie from G and H under $P_1\cup P_2$.

Proof:

The minimality property is satisfied after $Min()$ function is applied. Thus, to prove the theorem, we only have to show (F1) $\forall f, \forall f'$: $f, f'\in F$: $f\cap f'\neq\varnothing$ (F2) $\forall f, \forall g$: $f\in F, g\in G$: $f\cap g\neq\varnothing$ (F3) $\forall f, \forall h$: $f\in F, h\in H$: $f\cap h\neq\varnothing$.

Let f and f' be two sets in F. We have $f=(g\cup h)$ for some $g\in G$ and some $h\in H$, and $f'=(g'\cup h')$ for some $g'\in G$ and some $h'\in H$. Assume $f\cap f'=\varnothing$. It follows that $(g\cup h)\cap(g'\cup h')=\varnothing$. And hence, we have $g\cap g'=\varnothing$ and $h\cap h'=\varnothing$, which contradicts the fact that G and H are coteries. So, the condition (F1) holds.

Let f be a set in F. We have $f=(g\cup h)$ for some $g\in G$ and some $h\in H$. Assume $f\cap g'=\varnothing$ for some $g'\in G$. It follows that $(g\cup h)\cap g'=\varnothing$. We have $g\cap g'=\varnothing$, which contradicts the fact that G is a coterie. Thus, the condition (F2) holds.

Let f be a set in F. We have $f=(g\cup h)$ for some $g\in G$ and some $h\in H$. Assume $f\cap h'=\varnothing$ for some $h'\in H$. It follows that $(g\cup h)\cap h'=\varnothing$. We have $h\cap h'=\varnothing$, which contradicts the fact that H is a coterie. Thus, the condition (F3) holds. ■

In Theorem 3, G and H are taken to be coteries. However, G and H in Theorem 3 can also be taken to be p-coteries because a p-coterie is also a coterie. Below, we apply Theorem 3 with G being a p-coterie to prove the following Theorem 4.

Theorem 4. Let $P_1,\ldots,P_m$, $m>1$, be non-empty sets of processes. If Cr_j is a coterie under P_j, $1\leq j\leq m$, then $Min(Cr_1\otimes\ldots\otimes Cr_m)$ is a p-coterie from $Cr_1,\ldots,Cr_m$ under $P_1\cup\ldots\cup P_m$.

Proof : (by induction on the value of m)

(1) Basis: (m=2)

By Theorem 3, the basis case holds.

(2) Induction hypothesis:

Assume that if Cr_j is a coterie under P_j for $1\leq j\leq m$, then $G=Min(C_1\otimes\ldots\otimes C_m)$ is a p-coterie from $Cr_1,\ldots,Cr_m$ under $P_1\cup\ldots\cup P_m$.

(3) Induction step:

On the basis of the induction hypothesis, below we show that if Cr_j is a coterie under P_j for $1\leq j\leq m+1$, then $F=Min(Cr_1\otimes\ldots\otimes Cr_{m+1})$ is a p-coterie from $Cr_1,\ldots,Cr_{m+1}$ under $P_1\cup\ldots\cup P_{m+1}$.

Let G be $Min(Cr_1\otimes\ldots\otimes Cr_m)$. Then $F=Min(G\otimes Cr_{m+1})$. Since G is a p-coterie from $Cr_1,\ldots,Cr_m$ under $P_1\cup\ldots\cup P_m$ (by the induction hypothesis) and Cr_{m+1} is a coterie under P_{m+1}, we have F is a p-coterie from G and Cr_{m+1} under $P_1\cup\ldots\cup P_{m+1}$ by Theorem 3. Because G is a p-coterie from $Cr_1,\ldots,Cr_m$, each quorum in G intersects every quorum in $Cr_1,\ldots,Cr_m$.

And because $F=Min(G\otimes Cr_{m+1})$, any quorum f in F must be of the form $f=g\cup q$, where $g\in G$ and $q\in Cr_{m+1}$. It follows that each quorum in F intersects every quorum in $Cr_1,\ldots,Cr_{m+1}$. Hence, we have that F is a p-coterie from $Cr_1,\ldots,Cr_{m+1}$ under $P_1\cup\ldots\cup P_{m+1}$.

Therefore, by the induction principle, we have $Min(C_1\otimes\ldots\otimes C_m)$ is a p-coterie from $Cr_1,\ldots,Cr_m$ under $P_1\cup\ldots\cup P_m$ for $m>1$. ■

The following Theorem 5 is about the nondomination of the p-coteries generated by the p-union operation.

Theorem 5. Let $P_1,\ldots,P_m$, $m>1$, be non-empty sets of processes. Also let Cr_j be a coterie under P_j for $1\leq j\leq m$, and $F=Min(Cr_1\otimes\ldots\otimes Cr_m)$ be a p-coterie from $Cr_1,\ldots,Cr_m$ under $P_1\cup\ldots\cup P_m$. Then, F is ND if $Cr_1,\ldots,Cr_m$ are all ND.

Proof:

Assume F is dominated, then by Theorem 2, there must exist a set $x\subseteq(P_1\cup\ldots\cup P_m)$ such that (L1) $\forall f : f\in F : f\not\subseteq x$, (L2) $\forall f : f\in F : f\cap x\neq\varnothing$, (L3) $\forall q\forall j : q\in Cr_j, 1\leq j\leq m: q\cap x\neq\varnothing$.

Let $x_1=x\cap P_1$, $x_2=x\cap P_2$, …, and $x_m=x\cap P_m$. Then, we have $\forall q$: $q\in Cr_1$: $q\cap x_1\neq\varnothing$ because $q\cap x_1=q\cap x\cap P_1\neq\varnothing$ by (L3) and $(q\cap x)\subseteq P_1$. Similarly, we have $\forall q$: $q\in Cr_2$: $q\cap x_2\neq\varnothing$ because $q\cap x_2=q\cap x\cap P_2\neq\varnothing$ by (L3) and $(q\cap x)\subseteq P_2$. … And we have $\forall q$: $q\in Cr_m$: $q\cap x_m\neq\varnothing$ because $q\cap x_m=q\cap x\cap P_m\neq\varnothing$ by (L3) and $(q\cap x)\subseteq P_m$. To sum up, we have $\forall q\forall j$: $q\in Cr_j$, $1\leq j\leq m$: $q\cap x_j\neq\varnothing$.

Suppose $\forall q$: $q\in Cr_1$: $q\not\subseteq x_1$. Then, we have $Cr_1\cup\{x_1\}$ is a coterie dominating Cr_1, which contradicts the fact that Cr_1 is ND. It follows that $\exists q_1$: $q_1\in Cr_1$: $q_1\subseteq x_1$. We can proceed with the same inference to have $\exists q_2$: $q_2\in Cr_2$: $q_2\subseteq x_2$, …, and $\exists q_m$: $q_m\in Cr_m$: $q_m\subseteq x_m$. It follows that $(q_1\cup\ldots\cup q_m)\subseteq x$ since $(q_1\cup\ldots\cup q_m)\subseteq(x_1\cup\ldots\cup x_m)=(x\cap P_1)\cup\ldots\cup(x\cap P_m)\subseteq x$. Because $F=Min(Cr_1\otimes\ldots\otimes Cr_m)$, we have $\exists f$: $f\in F$: $f\subseteq(q_1\cup\ldots\cup q_m)$ by $\otimes$ operation definition. We then have $\exists f$: $f\in F$: $f\subseteq(q_1\cup\ldots\cup q_m)\subseteq x$, which contradicts (L1).

The assumption that F is dominated cannot stand. Hence, the theorem holds. ■

Note that we do not know whether the "only if" part of Theorem 5 (i.e., F is ND only if $Cr_1,\ldots,Cr_m$ are all ND) is true or not; we leave it as an open problem. Fortunately, Theorem 5 itself is sufficient to guide us to derive ND p-coteries for the construction of ND local coteries.

4. Concluding Remarks

In this paper, we have defined a new type of coteries, called *p-coteries*, to aid the construction of local coteries. We have developed theorems about the nondomination of p-coteries, and proposed an operation, called *pairwise-union* (*p-union*), to help generate ND p-coteries from known ND coteries. By the virtual resource concept discussed in Section 2 and all the theorems developed in Section 3, we now have the following 3 steps to construct an ND local coterie $LC=(C_1,\ldots,C_{|P|})$ to solve the distributed resource allocation problem for the system with process set P and resource set R:

Step 1. Treat a set S of resources that are accessed by the same set and only the same set of processes as a virtual resource v and let $R=(R\text{-}S)\cup\{v\}$. This step should be repeated until no S exists.

Step 2. For each resource r_j in R, construct an ND coterie Cr_j. Note that Cr_j may be a majority coterie [27], a tree coterie [1], a hierarchical coterie [19], a Lovasz coterie [21], a crumbling walls coterie [24] or a cohorts coterie [15].

Step 3. For each process p_i in P, construct an ND p-coterie C_i as follows. If p_i accesses only one resource, say r_j, then $C_i=Cr_j$. Otherwise, p_i accesses two or more resources, say $r_1,\ldots,r_m$, $m>1$. In such a case, $C_i=Min(Cr_1\otimes\ldots\otimes Cr_m)$.

In the future, we plan to study the availability of local coteries constructed with the aid of p-union operation, where the availability means the probability that a quorum can be successfully formed in an error-prone environment. We also plan to apply the p-union operation to ND *k*-coteries to solve the distributed multiple instance resource allocation problem, which is similar to the distributed resource allocation problem except that there are multiple instances for each shared resource.

References

[1] D. Agrawala and A. El Abbadi, "An efficient and fault-tolerant solution for distributed mutual exclusion," *ACM Transactions on Computing Systems*, 9(1):1-20, 1991.

[2] G. Agrawal and P. Jalote, "An efficient protocol for voting in distributed systems," in *Proc. of the 12th IEEE International Conference on Distributed Computing Systems*, pp. 640-647, 1992.

[3] J. Bar-Ilan and D. Peleg, "Distributed resource allocation algorithms," *Lecture Notes in Computer Science 647(WDAG '92)*, pp. 276-291, 1992.

[4] K. M. Chandy and J. Misra, "The drinking philosophers problem," *ACM Transactions on Programming Languages and Systems*, 6(4):632-646, 1984.

[5] M. Choy and A. K. Singh, "Efficient fault-tolerant algorithms for distributed resource allocation," *ACM Transactions on Programming Languages and Systems*, 17(3):535-559, 1995.

[6] Z. Cheng, Y. Wada, S. Hashimoto, A. He and T. Huang, "A new method for constructing efficient local coteries," in *Proc. of the 15th International Conference on Information Networking*, pp. 512 –517, 2001.

[7] E. W. Dijkstra, "Solution to a problem in concurrent programming control," *CACM*, 8(9):569, 1965.

[8] E. W. Dijkstra, "Hierarchical ordering of sequential processes," *Acta Informatica*, 1:115-138, 1971.

[9] M. Fisher, N. Lynch, J. Burns and A. Borondin, "Resource allocation with immunity to limited process failure," in *Proc. of the 20th IEEE annual symposium on foundations of Computer Science*, pp. 234–254, 1979.

[10] H. Garcia-Molina and D. Barbara, "How to assign votes in a distributed system," *JACM*., 32(4):841-860, 1985.

[11] S.-T. Huang, J.-R. Jiang and Y.-C. Kuo, "*k*-Coteries for fault-tolerant *k* entries to a critical section," in *Proc. of the 13th IEEE International Conference on Distributed Computing Systems*, pp.74-81, 1993.

[12] T. Harada and M. Yamashita, "Coterie join operation and tree structured *k*-coteries," *IEEE Transactions on Parallel and Distributed Systems*, 12(9):865-874, 2001.

[13] J.-R. Jiang and S.-T. Huang, "Obtaining nondominated *k*-coteries for fault-tolerant distributed *k*-mutual exclusion," in *Proc. of 1994 IEEE International Conference on Parallel and Distributed Systems*, pp. 582–587, 1994.

[14] J.-R. Jiang, S.-T. Huang and Y.-C. Kuo, "Cohorts structures for fault-tolerant *k* entries to a critical section," *IEEE Transactions on Computers*, 48(2):222-228, 1997.

[15] J.-R. Jiang, "Fault-tolerant distributed mutual exclusion with O(1) message overhead," in *Proc. of the 13th International Conference on Applied Informatics*, pp.228-231, 1995.

[16] J.-R. Jiang, "Distributed *h*-out of-*k* mutual exclusion using *k*-coteries," in *Proc. of the 3rd International Conference on Parallel and Distributed Computing, Application and Technologies* (*PDCAT'02*), pp. 218-226, 2002.

[17] H. Kakugawa, S. Fujita, M. Yamashita and T. Ae, "A distributed *k*-mutual exclusion algorithm using *k*-coterie," *Information Processing Letters*, 49:213-238, 1994.

[18] Y.-C. Kuo and S.-T. Huang, "Recognizing nondominated coteries and *wr*-coteries by availability," *IEEE Transactions on Parallel and Distributed Systems*, 9(8):721-728, 1998.

[19] A. Kumar, "Hierarchical quorum consensus: a new algorithm for managing replicated data," *IEEE Transactions on Computers*, 40(9):996-1004, 1991.

[20] H. Kakugawa and M. Yamashita, "Local coteries and a distributed resource allocation algorithm," *Transactions of Information Processing Society of Japan*, 37(8):1487-1496, 1996.

[21] M. L. Neilsen, "Quorum structures in distributed systems," *Ph. D. Dissertation*, Kansas State University, 1992.

[22] M. L. Neilsen and M. Mizuno, "Nondominated *k*-coteries for multiple mutual exclusion," *Information Processing Letters*, 50(5):247-252, 1994.

[23] M. L. Neilsen and M. Mizuno, "Erratum to nondominated *k*-coteries for multiple mutual exclusion," *Information Processing Letters*, 60(6):319-23, 1996.

[24] D. Peleg and A. Wool, "Crumbling walls: a class of practical and efficient quorum systems," *Distributed Computing*, 10:87-98, 1997.

[25] M. Raynal, "A distributed solution for the *k*-out of-*m* resources allocation problem," *Lecture Notes in Computer Sciences*, Springer Verlag, 497:599-609, 1991.

[26] I. Rhee, "A fast distributed modular algorithm for resource allocation," in *Proc. of the 15th IEEE International Conference on Distributed Computing Systems*, pp. 161-168, 1995.

[27] R. H. Thomas, "A majority consensus approach to concurrency control," *ACM Transactions on Database Systems*, 4(2):180-209, 1979.