

Real-time Optimal Selection of Multirobot Coalition Formation Algorithms using Conceptual Clustering

Sayan D. Sen

Department of Electrical Engineering
and Computer Science, Vanderbilt University
Nashville, Tennessee
sayan.d.sen@vanderbilt.edu

Julie A. Adams

Department of Electrical Engineering
and Computer Science, Vanderbilt University
Nashville, Tennessee
julie.a.adams@vanderbilt.edu

Abstract

Intelligent multirobot coalition formation systems equipped with mission driven algorithm selection strategies are crucial for a wide spectrum of real-world situations. *i-CiFHaR* is a unique coalition formation framework that comprises a library of diverse algorithms and uses a taxonomy-based real-time probabilistic reasoning to select the most appropriate algorithm(s) to apply. However, *i-CiFHaR* suffers from high computational time with an increase in the number of algorithms. This paper is the first to demonstrate the use of conceptual clustering in order to mine crucial patterns and relationships among existing coalition formation algorithms. *i-CiFHaR* leverages the derived optimal hierarchical classification tree to analyze only the most appropriate algorithms' cluster for application to a real-world mission scenario. The results show that the conceptual clustering technique reduces computation time by 67%.

The multirobot coalition formation problem seeks to intelligently partition a team of heterogeneous robots into coalitions for a set of real-world tasks. Besides being \mathcal{NP} -complete (Sandholm et al. 1999), the problem is also hard to approximate (Service and Adams 2011a). Traditional approaches to solving the problem include a number of greedy algorithms (Shehory and Kraus 1998; Vig and Adams 2006b), approximation algorithms (Sandholm et al. 1999), and auction-based approaches (Vig and Adams 2006a; Gerkey and Mataric 2002; Shiroma and Campos 2009). A single greedy algorithm-based system generates coalitions quickly, but fails to guarantee solution quality. Despite guaranteeing solution quality, approximation algorithm-based systems are inappropriate for real-time applications with large teams of robots, because of their high worst case run-time complexities. Auction-based algorithms provide the desired scalability and decentralization, but are inadequate for low communication environments due to their high communication overhead. Therefore, conventional single algorithm-based systems are inadequate and brittle for a wide spectrum of complex, uncertain real-world missions.

These limitations led to the development of the intelligent-Coalition Formation framework for Humans and Robots (*i-CiFHaR*) that incorporates a library of algorithms,

each applicable to different categories of real-world problems (Sen and Adams 2014). The framework exhibits probabilistic reasoning in order to make online decisions regarding the most suitable algorithm(s) to apply based on multiple mission criteria. The intelligent, real-time selection of appropriate algorithms from the library via a decision network renders *i-CiFHaR* flexible and adaptive, when computing robust coalitions for a wide spectrum of real-world missions. Despite the successful demonstration of *i-CiFHaR*'s algorithm selection capability for a wide variety of mission scenarios, it suffers from a high computational time as the number of algorithms and taxonomy attributes increase.

The presented work mines patterns in *i-CiFHaR*'s algorithms using conceptual clustering, an unsupervised machine learning approach in order to extract the most suitable cluster of algorithms for application analysis in a specific mission; thereby, accomplishing better scalability and reduced computational time. The presented framework is the first to leverage a conceptual clustering technique to partition any set of coalition formation algorithms in order to derive an optimal hierarchy classification tree, given any classification taxonomy. The results contribute to the state-of-the-art in multiagent systems by demonstrating the existence of crucial patterns and intricate relationships among existing coalition algorithms.

The presented *i-CiFHaR* framework leverages COBWEB, a conceptual clustering algorithm (Fisher 1987) for identifying clusters of similar algorithms in the library. Rather than employing all the algorithms, many of which may not be applicable during a particular mission situation, the improved framework uses the probabilistic metric, *category utility* (Gluck 1985) to identify the most suitable cluster of algorithms. Based on the selected cluster, the decision network optimizes the ranking of the algorithms in the chosen cluster by maximizing the expected utility scores. The experimental results show that algorithm rankings match those found for each of the twenty four mission scenarios as the original *i-CiFHaR*, but requiring approximately 67% less computation time.

The *Background* section provides a comprehensive overview of the related work. The incorporation of the conceptual clustering technique is described in the *System Design*. The experimental design, results, and the conclusions are provided in the subsequent sections.

Background

i-CiFHaR incorporates an expandable library of coalition formation algorithms and performs probabilistic online algorithm selection in accordance with multiple mission criteria (Sen and Adams 2014). The broad set of coalition formation algorithms that are implemented in *i-CiFHaR*'s library are categorized into three classes:

- Greedy (Shehory and Kraus 1998; Vig and Adams 2006b; Abdallah and Lesser 2004; Tošić and Agha 2005; Weerd, Zhang, and Klos 2007; Campbell, Wu, and Shumaker 2008; Sujit, George, and Beard 2008; Service and Adams 2011a; Gaston and desJardins 2005; Koes, Nourbakhsh, and Sycara 2005; Ramchurn et al. 2010; Zhang et al. 2010),
- Market/Auction-based (Vig and Adams 2006a; Gerkey and Mataric 2002; Service, Sen, and Adams 2014; Shirota and Campos 2009), and
- Approximation (Service and Adams 2011b; 2011a).

This paper aims to identify patterns, or clusters in the algorithms based on multiple taxonomy attributes, with the goal to extract only the most suitable cluster of algorithms for application analysis. This section highlights relevant clustering approaches and justifies the incorporation of COBWEB.

Clustering techniques are broadly categorized into partitioning and hierarchical classes. *k*-Means clustering (MacQueen 1967) falls into the former category and leverages a distance metric to identify a planar partitioning of the given data observations. The algorithm is applicable to large data sets, given its linear computational complexity, but its performance depends on the pre-specified *k* output clusters.

Hierarchical clustering generates hierarchical cluster trees. *Agglomerative* clustering starts with singleton clusters containing a single data observation and merges cluster pairs in a “bottom-up” fashion. Conversely, *divisive* clustering starts with a composite cluster containing all the data observations and recursively splits each cluster, until singleton clusters are reached. A distance metric (e.g., Euclidean distance, cosine similarity, Hamming distance) computes the degree of similarity between a pair of observations, while the linkage criteria computes the degree of similarity between two clusters and includes *single-linkage* (Sneath and Sokal 1973) and *complete-linkage* (Defays 1977). The high computational complexity, $\mathcal{O}(n^3)$ for a naive implementation and $\mathcal{O}(n^2)$ for a more efficient implementation (Defays 1977) is too inefficient for large data sets. Traditional clustering approaches leveraging numerical distance similarity metrics are only applicable to numerical data observations. However, data is often expressed in terms of some description (e.g., events, facts), rather than numerical values, wherein traditional techniques are rendered inapplicable due to their inability to acknowledge the context or concepts.

Conceptual clustering is a model-based approach, where the clustering is performed based on nominal, or categorical data descriptions and each cluster describes a concept, or data class (Michalski and Stepp 1983). Fisher (1987) introduced COBWEB, a conceptual clustering technique that learns a hierarchical classification tree from a set of observations. COBWEB's greedy, bi-directional search is driven by

a probabilistic function, the *category utility* (Gluck 1985). COBWEB's performance was improved by *pruning* singleton classes containing a single data observation (Fisher 1995). Talavera and Béjar (2001) presented the *Generality-based Concept Formation* (GCF) algorithm that incorporates a user-driven generality degree to compute the hierarchy of concepts based on a similarity histogram index. COBWEB was shown to perform better than GCF on three of six data sets. A number of extensions address COBWEB's inability to handle numerical data. The *Similarity-Based Agglomerative Clustering* algorithm leverages the Goodall similarity metric (Goodall 1966) to work with data comprising both numeric and nominal features (Li and Biswas 2002). COBWEB/3 incorporates probability distributions over the domain values of the numerical attributes (McKusick and Thompson 1990). However, these clustering algorithms do not handle missing and uncertain data (both nominal and/or numerical). Xia and Xi (2007) introduced the Extended-COBWEB that is designed to handle uncertain data.

The application of conventional clustering approaches that employ distance-based objective functions in order to discern the partitioning of *i-CiFHaR*'s coalition formation algorithms is inappropriate, because none of the taxonomy attributes that *i-CiFHaR* leverages have numerical domain sets. The *Similarity-Based Agglomerative Clustering* and COBWEB/3 algorithms are also irrelevant, because they are designed to accommodate data observations containing both numerical and nominal attributes. Although GCF performed similarly to COBWEB, its user dependency for hierarchy levels and generality degree necessitates a human in the loop, which is undesirable for real-world missions. Thus, the original COBWEB conceptual clustering algorithm is incorporated into *i-CiFHaR* for mining patterns among the library's algorithms that are described by nominal attributes.

System Design

i-CiFHaR leverages an existing coalition formation algorithm taxonomy that provides a comprehensive set of eighteen attributes (dimensions/features) for the multirobot task allocation problem (Sen and Adams 2014). Each of the attributes, F_i has an associated non-empty domain set, D_i of nominal values, such that F_i can be assigned a value, $V_j \in D_i$ to generate an attribute-value pair, (F_i, V_{ij}) . Table 1 provides an example containing four taxonomy attributes and their nominal domain sets.

Taxonomy Attributes (F)	Feature Domain Set (D)
Agent Capability Model	{Resource, Service}
Agent Structure	{Social Network, None, Organization Hierarchy}
Inter-Task Constraints	{Yes, Prerequisite, No}
Task Preemption	{Yes, No}

Table 1: Taxonomy attributes and nominal domain values.

Each of *i-CiFHaR*'s algorithms, A_x representing a data point in the context of clustering, is characterized by its re-

Algorithms	Taxonomy Attributes														
	F_3	F_5	F_6	F_7	F_8	F_9	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	
Shehory and Kraus (1998)	Res	None	PReq	No	Res	No	MC	High	IA	No	Yes	Greedy	DC	k	
Abdallah and Lesser (2004)	Res	Org	PReq	No	Res	No	MU	Low	TE	No	No	Greedy	DC	None	
Shiroma and Campos (2009)	Ser	None	PReq	No	Ser	No	MU	Low	IA	No	Yes	Auction	DC	None	

Taxonomy Attribute Key

F_3 : Agent Capability Model

F_5 : Agent Structure

F_6 : Inter-Task Constraints

F_7 : Task Preemption

F_8 : Task Requirement Model

F_9 : Intra-Task Constraints

F_{11} : Performance Criterion

F_{12} : Communication Overhead

F_{13} : Task Allocation

F_{14} : Spatial Constraints

F_{15} : Overlapping Coalitions

F_{16} : Algorithm Technique

F_{17} : Algorithm Implementation

F_{18} : Coalition Size Constraint

Attribute Domain Value Key

TE:Time-Extended

PReq:Prerequisite

Res:Resource-Model

k :Bounded Size

DC:Decentralized

MC:Minimize Cost

MU:Maximize Utility

Ser:Service-Model

Org:Organization Hierarchy

IA:Instantaneous

Table 2: Three coalition formation algorithms with respective taxonomy attribute-value pairs

spective vector, $AV_x = \{(F_i, V_{ij})\}$ of attribute-value pairs. Table 2 provides the taxonomy attributes and domain values for three of the nineteen coalition formation algorithms in *i-CiFHaR*'s library. *i-CiFHaR* employs a *principal component analysis* in order to identify the fourteen most important taxonomy attributes that contribute significantly in the algorithm classification. These fourteen attributes are presented in Table 2. Although the algorithms are associated with certain attribute-value pairs, a real-world mission scenario is highly dynamic with uncertain or missing information. Such uncertain missions are represented using a general model for nominal or categorical data with uncertainty. Under this uncertainty nominal model, each mission situation is represented by a vector of *uncertain categorical attributes* (UCAs), each of which is assigned to one of the attribute's nominal domain values with some probability that signifies the event's likelihood. Each UCA is represented by a probability distribution over the attribute's domain set. Let an attribute, $F_i \in F$ be assigned a particular value, V_{ij} from F_i 's domain set, D_i with some probability p_{ij} . Assuming the cardinality of F_i 's domain set, $D_i = |D_i|$, the attribute's probability distribution over all the domain values is governed by $\sum_{j=1}^{|D_i|} p_{ij} = 1$. Therefore, each mission situation, MS_y is represented by a vector, MV_y of uncertain nominal attribute-value pairs, $\{(F_i, V_{ij}, p_{ij})\}$.

i-CiFHaR without clustering employs a decision network to handle uncertain real-world missions. *i-CiFHaR* leverages *principal component analysis* in order to identify only the prominent taxonomy attributes for classification, thereby performing dimensionality reduction. The extracted prominent attributes become the random *chance nodes* for the framework's decision network. *i-CiFHaR*'s objective is to optimize the selection of the coalition formation algorithms; therefore, the domain of the decision network's *decision node* contains all algorithms in the library. All the *chance nodes* and the decision node become the immediate parents of the network's *utility node*. The *utility node*'s utility table

is exponential in size, in terms of all possible configurations of the parent nodes. During decision making, *i-CiFHaR* acts as a rational agent and uses the exponentially large utility table to compute the expected utility score for each of its available action choices (i.e., coalition formation algorithms). The most suitable algorithm is the one that maximizes the framework's expected utility score. Additionally, in order to provide decision support, *i-CiFHaR* can select a subset of algorithms with expected utility scores exceeding a certain threshold, $EU^* = \gamma \times \max_{A_x \in CFL} EU(A_x)$, where $EU(\bullet)$ denotes the expected utility score, γ signifies a pre-specified deviation percentage, and A_x represents a particular coalition formation algorithm in *i-CiFHaR*'s library, *CFL*. When analyzing each mission, many algorithms may not be applicable; therefore, conceptual clustering is used to identify the most appropriate cluster of algorithms, given the mission criteria. *i-CiFHaR*'s decision network performs the expected utility optimization by analyzing only the algorithms in the selected cluster for application to a given mission scenario. The decision network uses only the most suitable cluster of algorithms, instead of the entire library of algorithms; thus, limiting the utility table size considerably and resulting in an order of magnitude improvement in the computation time.

The improved *i-CiFHaR* incorporating COBWEB's conceptual clustering algorithm (Fisher 1987) partitions the library's coalition formation algorithms into clusters. COBWEB incrementally builds a hierarchical classification tree of concept nodes without a predefined number of clusters, a crucial drawback in some partitioning clustering approaches. COBWEB starts with an empty root node and each algorithm, expressed as a vector of nominal attribute-value pairs is added to an incremental classification tree, one at a time. COBWEB performs a hill-climbing search through the classification space, which is governed by the *category utility* heuristic (Gluck 1985; Fisher 1987). The *category utility* metric is a tradeoff between intra-class similarity and inter-

class dissimilarity. Given a cluster, C_k , the *category utility* metric is defined as:

$$CU(C_k) = P(C_k) \left[\sum_i \sum_j P(F_i = V_{ij} | C_k)^2 - \sum_i \sum_j P(F_i = V_{ij})^2 \right], \quad (1)$$

where $P(\bullet)$ defines the probability, $F_i \in F$ denotes the taxonomy attribute, and $F_i = V_{ij}$ represents an attribute-value pair, when F_i is assigned to the j^{th} domain value, $V_{ij} \in D_i$. $P(F_i = V_{ij} | C_k)^2$ defines the intra-class similarity and represents the expected number of attribute-value pairs correctly guessed, given a particular class (Fisher 1987). $P(F_i = V_{ij})^2$ represents the expected number of attribute-value pairs guessed when no classification is provided.

COBWEB (Fisher 1987) creates a classification tree, where the root node represents the concept containing all data observations (*i-CiFHaR*'s coalition formation algorithms), while the leaf nodes represent singleton concepts, each containing an individual observation. Each node either contains singleton concepts, or subsumes other sub-concepts. Additionally, each node holds the attribute-value counts of all the objects that it contains; therefore, representing a probability concept label. COBWEB incrementally absorbs a new object into the existing hierarchy, while employing four operators recursively in order to classify the object into the best matching concept. Given a node, the *addition* operator adds the new object to one of the node's children and computes the *CU* score for each case, with the objective of identifying the best two concept clusters that can house the new object. The *create a new class* operator generates a new singleton concept containing only the new observation and adds this concept to the given node. COBWEB attempts to counter the ill-effects of initially skewed data by introducing two operators. The *merge* operator combines the two best hosts into a new combined concept, which is accepted as a better partition, if and only if the *CU* score is higher than the previously generated clusters. The *split* operator decomposes the best concept into multiple concept clusters.

The COBWEB's search is heuristic; thus, the generated classification tree varies across multiple trials of the algorithm depending on the ordering of the data set. *i-CiFHaR* mitigates the influence of an initially skewed data set by randomly selecting an initial algorithm seed, followed by an iterative selection of a different algorithm data point that maximizes the Manhattan distance between it and the previous n seeds. *i-CiFHaR* incrementally derives a classification tree for each of thirty trials, and on each instance, calculates the partition utility score, $PU = \frac{1}{m} \sum_{k=0}^{m-1} CU(C_k)$ of the partition structure containing m clusters at first level of the tree (Level-0 denotes the root node). The tree with the maximum *PU* score is deemed the best partitioning of the coalition algorithms, given the objective function. Once *i-CiFHaR* identifies the optimal classification tree with the maximum *PU* score, uncertain real-world missions, described in terms of a vector of uncertain attribute-value pairs are classified according to this best hierarchical tree.

The original COBWEB conceptual clustering technique is used to pre-compute the optimal clustering hierarchy of the coalition formation algorithms in *i-CiFHaR*'s library, because each of the algorithms is described in terms of the taxonomy attributes containing nominal domain values with complete certainty. This offline processing of the algorithms' hierarchical partitioning is justified, because the system library will not change during the real-world applications. However, during the online classification of the uncertain mission scenarios, all the mission attribute-value pairs are assigned likelihood probabilities to simulate uncertainties in the real-world. The original COBWEB is not designed to handle uncertain data sets; therefore, *i-CiFHaR* adopts the modified $CU(C_k)$ calculation methodology from the Extended-COBWEB (Xia and Xi 2007) in order to classify the uncertain mission scenarios according to the pre-computed optimal algorithm hierarchy with the intent of identifying the best matching algorithm cluster.

For example, let F_1 and F_2 represent two attributes and a particular concept cluster, C_1 in the identified tree has three algorithm objects. Let the domain sets of the attributes, F_1 and F_2 be $\{V_{11}, V_{12}\}$ and $\{V_{21}, V_{22}\}$, respectively. Let the objects in the cluster be described in terms of the attribute-value pairs: $[A_1 : \{F_1 = V_{11}, F_2 = V_{21}\}]$, $[A_2 : \{F_1 = V_{11}, F_2 = V_{22}\}]$, and $[A_3 : \{F_1 = V_{11}, F_2 = V_{22}\}]$. Therefore, the concept attribute-value counts are: $[\{F_1 = V_{11} : 3, F_2 = V_{21} : 1, F_2 = V_{22} : 2\}]$. Addressing mission uncertainty, the concept counts include all possible attribute-value pair counts, even when some of the pairs are zero. For example, the concept count is represented as: $[\{F_1 = V_{11} : 3, F_1 = V_{12} : 0, F_2 = V_{21} : 1, F_2 = V_{22} : 2\}]$. A sample mission, described as a vector of *UCAs* is represented as: $[\{F_1 = V_{11} : 0.8, F_1 = V_{12} : 0.2, F_2 = V_{21} : 0.7, F_2 = V_{22} : 0.3\}]$. During the *category utility* computation with the mission added concept C_1 , the probability counts take the format: $[\{F_1 = V_{11} : 3.8, F_1 = V_{12} : 0.2, F_2 = V_{21} : 1.7, F_2 = V_{22} : 2.3\}]$. Once the mission scenario is categorized to a particular algorithm cluster, then the identified cluster contains the most appropriate subset of algorithms for the mission. The cluster provides the action choices for *i-CiFHaR*'s decision network, which optimizes and ranks only the algorithms within the identified cluster.

Experiments and Results

The *i-CiFHaR* library contains nineteen algorithms programmed on a Linux platform (Ubuntu-12.04) with an Intel Core i5, 2.30GHz processor using C++ and the Qt framework (version 4.8) (Nokia 2012). *i-CiFHaR*'s influence diagram implementation uses the Netica-C API (NORSYS 2012), a Bayesian network development software tool that leverages a junction tree for calculations. An open-source Python implementation of COBWEB (McLellan 2014) was used to generate the coalition formation algorithm partitions.

A total of 124,416 mission scenarios are possible with 14 prominent taxonomy features, many of which are unrealistic. Twenty four carefully selected mission scenarios are reused from prior experiments that exploit crucial taxonomy attribute-value pair combinations. Each mission scenario is associated with a vector of attribute-value pairs, along with a

Key	
A_1 :(Shehory and Kraus 1998)	A_{10} :(Gerkey and Matarić 2002)
A_2 :(Vig and Adams 2006b)	A_{11} :(Service and Adams 2011a)
A_3 :(Vig and Adams 2006a)	A_{12} :(Service and Adams 2011b)
A_4 :(Abdallah and Lesser 2004)	A_{13} :(Service and Adams 2011a)
A_5 :(Tošić and Agha 2005)	A_{14} :(Service, Sen, and Adams 2014)
A_6 :(Weerd, Zhang, and Klos 2007)	A_{15} :(Gaston and desJardins 2005)
A_7 :(Campbell, Wu, and Shumaker 2008)	A_{16} :(Koes, Nourbakhsh, and Sycara 2005)
A_8 :(Sujit, George, and Beard 2008)	A_{17} :(Ramchurn et al. 2010)
A_9 :(Service and Adams 2011a)	A_{18} :(Shiroma and Campos 2009)
	A_{19} :(Zhang et al. 2010)

Level	Clusters	Algorithms
Root	C_0	$\{A_1 \text{ through } A_{19}\}$
Level-1	C_1	$\{A_3, A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}, A_{17}, A_{18}\}$
	C_2	$\{A_1, A_2, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{19}\}$
Level-2	C_3	$\{A_1, A_2, A_9, A_{19}\}$
	C_4	$\{A_4, A_5, A_6, A_7, A_8, A_{10}\}$
	C_5	$\{A_3, A_{14}\}$
Level-3	C_6	$\{A_{11}, A_{12}, A_{13}, A_{15}, A_{16}, A_{17}, A_{18}\}$
	C_7	$\{A_2, A_9\}$
	C_8	$\{A_4, A_5, A_6\}$
	C_9	$\{A_{12}, A_{13}\}$
	C_{10}	$\{A_{16}, A_{17}\}$

Table 3: Concept clusters (non-leaf nodes) and the respective coalition formation algorithms. Levels 1-3 contain some singleton clusters that include the algorithms that are not listed at the specific level.

probability value. Consider two mission scenarios, MS_y and MS_z that simulate very different real-world situations. The missions follow the *service model*, where robots' capabilities and task requirements are described in terms of services (e.g., sentry-duty, surveillance) with high likelihood. The resulting uncertain attribute-value pairs are $\{Agent\ Capability = service: 0.8\}$ and $\{Task\ Requirement = service: 0.8\}$. However, the missions differ in many aspects. MS_y 's objective is to maximize utility, thus $\{Performance = Maximize\ Utility\ (MU): 0.6\}$. Conversely, MS_z seeks to maximize the number of tasks completed, thus $\{Performance = Maximize\ Task\ (MT): 0.7\}$. MS_y does not require overlapping coalitions, $\{Overlapping = No: 0.7\}$, but MS_z aims to control resource losses, $\{Overlapping = Yes: 0.9\}$. The remaining attribute-value pairs are assigned probability values similarly in order to simulate uncertainty within the mission scenario.

i-CiFHaR's nineteen algorithms were processed in thirty trials of the COBWEB algorithm in order to generate different classification trees. The *partition utility* score was computed for each trial and the hierarchical cluster with the highest *PU* score was identified as the best partitioning, given the evaluation function. Once the most suitable cluster of algorithms was identified, it was provided to *i-CiFHaR*'s decision network in order to generate the algorithm rankings.

Results

The hierarchical classification tree, with the maximum *PU* score as identified by COBWEB is shown in Table 3. The hierarchy is represented by the level numbers with the root containing all the coalition formation algorithms at Level-0. As one moves down the hierarchy, children C_1 and C_2 at Level-1 partition the entire library into *Service-model* and *Resource-model* based algorithms. More concept clusters are realized that group similar algorithms based on their attribute-value pairs lower in the hierarchy levels. The leaf nodes (nineteen singleton concepts), one for each algorithm

are not depicted.

i-CiFHaR acts as a decision support system; therefore, it selects either a single coalition formation algorithm, or a subset of algorithm(s) that satisfy all or most of a given mission's criteria. *i-CiFHaR* analyzes the most suitable cluster, as identified by COBWEB and optimizes the algorithm rankings by maximizing the expected utility score. The algorithm with the maximum expected utility score satisfies all or most of the mission criteria and is ranked first. The remaining algorithms have expected utility scores higher than 90% of the maximum expected score derived from all the algorithms in the selected cluster. The coalition formation algorithm rankings for each of the twenty four missions are provided in Table 4. The table also provides the pertinent cluster and the cluster size by mission.

Figure 1 presents *i-CiFHaR*'s computational time with and without the COBWEB clustering. It is to be noted that *i-CiFHaR*'s computational time with clustering is lower than *i-CiFHaR* without clustering across all the mission scenarios. The mean time of the latter is 16.16 seconds, with a standard deviation is 0.32 seconds. Conversely, *i-CiFHaR* with clustering computed solutions in a mean time of 5.35 seconds, with a standard deviation of 2.76 seconds, a 67% improvement. The computational time standard deviation differences stems from the fact that *i-CiFHaR* without clustering always considered all nineteen algorithms, irrespective of the mission scenario. The conceptual clustering based *i-CiFHaR* selects a single best cluster of algorithms to apply to a given mission, but the cluster sizes differ, as seen in Table 4. The computational time of certain missions (e.g., MS_6, MS_{19}, MS_{20}) is much higher than that of other missions (e.g., MS_1, MS_3, MS_{11}), because the cluster sizes in the former scenarios are much larger than those of the latter.

i-CiFHaR with clustering produced identical algorithm rankings as *i-CiFHaR* without clustering (see Table 4) for twenty-two mission scenarios. *i-CiFHaR* without cluster-

ing selected three algorithms, A_{18} , A_{19} , and A_{11} for MS_5 ; however, i -CiFHaR with clustering chose two *Service-model* based algorithms, A_{18} and A_{11} from cluster C_6 and excluded A_{19} , because A_{19} leverages a *Resource-model* and belongs to cluster C_3 . Similarly, for MS_8 , i -CiFHaR without clustering selected algorithms A_{15} and A_6 , but i -CiFHaR with clustering ranked A_{15} from cluster C_6 as the most suitable choice and dropped A_6 , which being a *Resource – model* based algorithm belongs to cluster C_8 .

Cluster # (Size)	Mission Scenarios	Rankings of Algorithms				
		1 st	2 nd	3 rd	4 th	5 th
$C_3(4)$	MS_1	A_9	A_1	A_2	-	-
$C_3(4)$	MS_2	A_1	A_2	A_9	-	-
$C_3(4)$	MS_3	A_1	A_2	A_{19}	A_9	-
$C_3(4)$	MS_4	A_{19}	A_9	-	-	-
$C_6(7)$	MS_5	A_{18}	A_{11}	-	-	-
$C_2(10)$	MS_6	A_5	A_6	A_9	A_4	-
$C_8(3)$	MS_7	A_6	A_5	A_4	-	-
$C_6(7)$	MS_8	A_{15}	-	-	-	-
$C_2(10)$	MS_9	A_4	A_6	A_9	A_5	-
$C_5(2)$	MS_{10}	A_3	A_{14}	-	-	-
$C_5(2)$	MS_{11}	A_{14}	A_3	-	-	-
$C_9(2)$	MS_{12}	A_{12}	A_{13}	-	-	-
$C_6(7)$	MS_{13}	A_{11}	A_{12}	A_{13}	-	-
$C_4(6)$	MS_{14}	A_8	A_7	-	-	-
$C_{10}(2)$	MS_{15}	A_{16}	A_{17}	-	-	-
$C_{10}(2)$	MS_{16}	A_{17}	A_{16}	-	-	-
$C_2(10)$	MS_{17}	A_{10}	A_9	-	-	-
$C_1(9)$	MS_{18}	A_{16}	A_{14}	A_{11}	A_3	-
$C_1(9)$	MS_{19}	A_{17}	A_{16}	A_{14}	A_{11}	A_3
$C_1(9)$	MS_{20}	A_{18}	A_3	A_{14}	A_{11}	-
$C_1(9)$	MS_{21}	A_{15}	A_{18}	A_3	A_{14}	A_{11}
$C_1(9)$	MS_{22}	A_{18}	A_{11}	A_{12}	A_{13}	-
$C_2(10)$	MS_{23}	A_8	A_{19}	A_1	A_2	A_9
$C_4(6)$	MS_{24}	A_6	A_8	A_5	A_4	-

Table 4: Algorithm rankings of each mission scenario by decreasing expected utility scores. The cluster sizes for each mission are also provided.

i -CiFHaR’s computational time, by cluster size is provided in Table 5. i -CiFHaR’s computational time with COBWEB increases linearly with the cluster size. However, as i -CiFHaR scales to include more algorithms in the library, COBWEB can generate a different classification hierarchy tree comprised of clusters with different sizes. An increased cluster size will result in increased computational time. The worst case leverages the root cluster containing all algorithms in the library, as is utilized by i -CiFHaR without clustering; thereby, considering $\mathcal{O}(n)$ algorithms for decision making. However, with the hierarchical clustering approach, i -CiFHaR potentially will leverage $\mathcal{O}(\log_b n)$ algorithms, where n is the number of algorithms and b is the branching factor of the hierarchical tree.

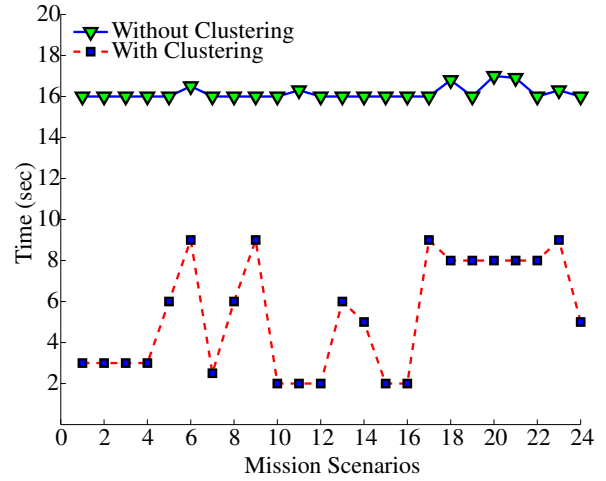


Figure 1: i -CiFHaR’s computational time in seconds with and without conceptual clustering.

Cluster Size	2	3	4	6	9	10	19
Computation Time (sec)	2	2.5	3.0	5	8	9	16.2

Table 5: Cluster Size vs Average Computation Time (sec).

Conclusions

The i -CiFHaR framework with conceptual clustering of coalition formation algorithms has been presented. The COBWEB conceptual clustering partitions the algorithms based on attribute-value pairs, and i -CiFHaR analyzes only the most suitable cluster of algorithms for application to a given mission. Experimental results show that the rankings of the algorithms for each mission scenario remained the same when i -CiFHaR with clustering was compared to i -CiFHaR without clustering. However, the computational time is 67% faster when the conceptual clustering version is used with no detrimental effects on performance. Real-time applications may require faster processing; however, the presented framework exhibits “optimal” algorithm selection strategies with a mean of 5.35 seconds, which is quite good when human mission planners are planning operations. The incorporation of COBWEB’s conceptual clustering renders the framework more scalable as the library of algorithms grows; thereby, making i -CiFHaR more flexible and applicable to a wider range of real-world mission situations.

Acknowledgments

The authors thank Dr. Douglas Fisher for valuable discussions on conceptual clustering and Christopher MacLellan for providing guidance on incorporating his Python version of COBWEB. This research was partially supported by ONR grant # N000140911161.

References

Abdallah, S., and Lesser, V. 2004. Organization-based cooperative coalition formation. In *Proceedings of the*

- IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 162–168.
- Campbell, A.; Wu, A. S.; and Shumaker, R. 2008. Multi-agent task allocation: Learning when to say no. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, 201–208.
- Defays, D. 1977. An efficient algorithm for a complete link method. *The Computer Journal* 20(4):364–366.
- Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2):139–172.
- Fisher, D. 1995. Optimization and simplification of hierarchical clusterings. In *KDD*, 118–123.
- Gaston, M. E., and desJardins, M. 2005. Agent organized networks for dynamic team formation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, 230–237.
- Gerkey, B., and Matarić, M. J. 2002. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation* 18(5):758–768.
- Gluck, M. 1985. Information, uncertainty and the utility of categories. In *Proc. of the Seventh Annual Conf. on Cognitive Science Society*, 283–287.
- Goodall, D. W. 1966. A new similarity index based on probability. *Biometrics* 882–907.
- Koes, M.; Nourbakhsh, I.; and Sycara, K. 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 1292–1297.
- Li, C., and Biswas, G. 2002. Unsupervised learning with mixed numeric and nominal data. *Knowledge and Data Engineering, IEEE Transactions on* 14(4):673–690.
- MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 281–297.
- McKusick, K., and Thompson, K. 1990. Cobweb/3: A portable implementation.
- McLellan, C. 2014. Concept formation. https://github.com/cmaclell/concept_formation.git.
- Michalski, R. S., and Stepp, R. E. 1983. Learning from observation: Conceptual clustering. In *Machine learning*. 331–363.
- Nokia. 2012. Qt. <http://www.qt.nokia.com>.
- NORSYS. 2012. Netica application: A complete software package to solve problems using bayesian belief networks and influence diagrams. <http://www.norsys.com/netica.html>.
- Ramchurn, S. D.; Polukarov, M.; Farinelli, A.; Truong, C.; and Jennings, N. R. 2010. Coalition formation with spatial and temporal constraints. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, 1181–1188.
- Sandholm, T.; Larson, K.; Andersson, M.; Shehory, O.; and Tohmé, F. 1999. Coalition structure generation with worst case guarantees. *Artificial Intelligence* 111(1-2):209–238.
- Sen, S. D., and Adams, J. A. 2014. An influence diagram based multi-criteria decision making framework for multi-robot coalition formation. *Autonomous Agents and Multi-Agent Systems* 1–30.
- Service, T. C., and Adams, J. A. 2011a. Coalition formation for task allocation: Theory and algorithms. *Journal of Autonomous Agents and Multi-agent Systems* 22(2):225–248.
- Service, T. C., and Adams, J. A. 2011b. Constant factor approximation algorithms for coalition structure generation. *Journal of Autonomous Agents and Multi-Agent Systems* 23(1):1–17.
- Service, T. C.; Sen, S. D.; and Adams, J. A. 2014. A simultaneous descending auction for multirobot task allocation. In *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics*.
- Shehory, O., and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence* 101:165–200.
- Shiroma, P. M., and Campos, M. F. M. 2009. CoMutaR: A framework for multi-robot coordination and task allocation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 4817–4824.
- Sneath, P. H., and Sokal, R. R. 1973. *Numerical taxonomy. The principles and practice of numerical classification*.
- Sujit, P. B.; George, J. M.; and Beard, R. W. 2008. Multiple UAV coalition formation. In *Proceedings of the American Control Conference*, 2010–2015.
- Talavera, L., and Béjar, J. 2001. Generality-based conceptual clustering with probabilistic concepts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23(2):196–206.
- Tošić, P. T., and Agha, G. A. 2005. Maximal clique based distributed coalition formation for task allocation in large-scale multi-agent systems. In *Proceedings of the Massively Multi-Agent Systems I*, 104–120.
- Vig, L., and Adams, J. A. 2006a. Market-based multi-robot coalition formation. In *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems*, 227–236.
- Vig, L., and Adams, J. A. 2006b. Multi-robot coalition formation. *IEEE Transactions on Robotics* 22(4):637–649.
- Weerd, M.; Zhang, Y.; and Klos, T. 2007. Distributed task allocation in social networks. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 17–24.
- Xia, Y., and Xi, B. 2007. Conceptual clustering categorical data with uncertainty. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 1, 329–336.
- Zhang, G.; Jiang, J.; Su, Z.; Qi, M.; and Fang, H. 2010. Searching for overlapping coalitions in multiple virtual organizations. *Information Sciences* 180(17):3140–3156.