

An influence diagram based multi-criteria decision making framework for multirobot coalition formation

Sayan D. Sen · Julie A. Adams

Published online: 15 October 2014
© The Author(s) 2014

Abstract Novel systems allocating coalitions of humans and unmanned heterogeneous vehicles will act as force multipliers for future real-world missions. Conventional coalition formation architectures seek to compute efficient robot coalitions by leveraging either a single greedy, approximation, or market-based algorithm, which renders such architectures inapplicable to a variety of real-world mission scenarios. A novel, intelligent multi-criteria decision making framework is presented that reasons over a library of coalition formation algorithms for selecting the most appropriate subset of algorithm(s) to apply to a wide spectrum of complex missions. The framework is based on influence diagrams in order to handle uncertainties in dynamic real-world environments. An existing taxonomy comprised of multiple mission and domain dependent features is leveraged to classify the coalition formation algorithms. Dimensionality reduction is achieved via principal component analysis, which extracts the most significant taxonomy features crucial for decision making. A link analysis technique provides the mission specific utility values of each feature-value pair and algorithm in the library. Experimental results demonstrate that the presented framework accurately selects the most appropriate subset of coalition formation algorithm(s) based on multiple mission criteria, when applied to a number of simulated real-world mission scenarios.

Keywords Coalition formation · Influence diagrams · Link analysis · Multi-criteria decision making

1 Introduction

Complex mission requirements are often greater than the capabilities of a single robot. Coalition formation can intelligently group heterogeneous unmanned vehicles (robots) to perform

S. D. Sen (✉) · J. A. Adams
Department of Electrical Engineering and Computer Science,
Vanderbilt University, Nashville, TN 37240, USA
e-mail: sayan.d.sen@vanderbilt.edu

J. A. Adams
e-mail: julie.a.adams@vanderbilt.edu

tasks collectively. Coalition formation is an NP-complete problem [1], which is also hard to approximate within a factor of $O(m^{1-\epsilon})$ for m tasks ($\epsilon > 0$) [2]. This computational intractability has led to the development of a number of greedy [3–6], anytime [1,7,8], design-to-time [9], and market-based [10–12] algorithms; however, all of these algorithms have drawbacks. Greedy algorithms generate solutions quickly, but do not guarantee solution quality and often result in suboptimal solutions. Anytime algorithms guarantee solution quality and permit premature termination, but require $O(n^n)$ worst case run-time to obtain quality solutions. Conversely, design-to-time algorithms must run to completion in order to generate near-optimal solutions, but have a better worst case run-time ($O(3^n)$) when compared to anytime algorithms. Market-based approaches suffer from high communication overhead, longer convergence time, and are highly sensitive to bidding parameters. Against this background, real-world critical missions requiring real-time coalitions may have a greater propensity towards using heuristic-based algorithms that can generate solutions of acceptable quality within the stipulated mission time.

Conventional coalition formation systems typically incorporate a single algorithm tailored for specific mission scenarios, which make such systems fragile and inapplicable to a wide spectrum of highly dynamic and complex real-world missions. Heuristic algorithms have been shown experimentally to be effective only in compatible real-world missions [13]. However, the dynamics and uncertainties of real-world environments do not guarantee the availability of such information in every mission scenario; thus, a single heuristic algorithm based system will be rendered less applicable. Conversely, despite guaranteeing solution quality, approximation approaches do not scale well to large robot teams due to their exorbitant worst case run-time complexities. Therefore, time constrained, critical real-world missions cannot be accomplished using a single approximation algorithm based coalition formation system.

These limitations motivate the primary contribution, which is the development of the intelligent-Coalition Formation framework for Humans and Robots (*i-CiFHaR*) that incorporates a library of coalition formation algorithms, each employing a different problem solving approach. Equipped with probabilistic reasoning based on influence diagrams, *i-CiFHaR* reasons over the library and selects the most appropriate algorithm(s) to apply in accordance with multiple mission criteria and environmental constraints when generating robot coalitions. The unified library coupled with the intelligent algorithm selection process allows *i-CiFHaR* to generate robust solutions for a wide variety of real-world missions and provide effective decision support to human mission supervisors. Experimental results demonstrate that *i-CiFHaR* successfully chooses the most suitable subset of algorithm(s) by optimizing the expected utility score, when applied to a number of simulated mission scenarios.

Consider a first response mission after a bomb blast, where a single algorithm requiring high communication bandwidth may be rendered inapplicable. Reports suggest that multiple victims have been injured and there is a high likelihood of unknown bombs in the area. Coalitions of robots assess the situation and report back the victim locations and any potential threats. The wireless connectivity can be negatively impacted due to the increased electromagnetic activities. During such a mission, *i-CiFHaR* can dynamically re-evaluate the environmental constraints and refine the algorithm selection procedure, thereby selecting an alternative algorithm applicable to the mission's new low communication criterion.

i-CiFHaR leverages a coalition formation algorithm taxonomy [14] that classifies coalition algorithms along multiple dimensions. Complex real-world missions can have high uncertainty; therefore, *i-CiFHaR* employs an influence diagram to address uncertainties during decision making. Influence diagrams are an extension of Bayesian networks that model decision making under multiple, uncertain criteria [15]. *i-CiFHaR* generates an influence diagram online based on a subset of prominent taxonomy features extracted via *Principal Component*

Analysis. The smaller subset of features addresses the combinatorial explosion issue associated with an increase in the number of taxonomy features. This problem dimensionality reduction, along with a built-in mathematical model for computing the utility table, renders *i-CiFHaR* scalable and flexible.

i-CiFHaR is most similar to the autonomous squadron formation framework for unmanned aerial vehicles [13], which leverages three coalition formation algorithms to compute the necessary teams. The brute force algorithms search through all possible $(2^n - 1)$ combinations of n vehicles, rendering the framework impractical for large values of n . Coalitions are computed by all three algorithms and the best coalition is selected based on a utility metric [13]. *i-CiFHaR* differs in two ways: (1) a library of nineteen diverse and intelligent algorithms is incorporated, which renders *i-CiFHaR* more adaptable to a wide range of domains; and (2) an intelligent algorithm selection process chooses the most appropriate algorithm(s) to apply, instead of applying all available algorithms, many of which may not be applicable.

Section 2 provides an overview of the related work. *i-CiFHaR*'s design and implementation are detailed in Section 3. Experimental design and results are provided in Section 4. Section 5 presents the conclusion.

2 Related work

A number of coalition formation algorithms have been proposed to address the combinatorial optimization problem, each tailored for specific mission conditions. The presented framework, *i-CiFHaR* is the first system to leverage a library of diverse coalition formation algorithms as one of its many built-in components. This section summarizes a subset of existing coalition formation algorithms from the literature, a majority of which have been incorporated into *i-CiFHaR*'s library.

Shehory and Kraus' heuristic-based coalition formation algorithm uses group-rational agents to generate both overlapping and disjoint coalitions by minimizing the overall system cost [4]. The algorithm runs in polynomial time by leveraging a heuristic that permits coalitions up to a maximum size, k . Vig and Adams [6] extended Shehory and Kraus' algorithm for multirobot domains by modeling resources as non-transferable entities and each task as a constraint graph. This algorithm utilizes arc-consistency to validate potential coalitions and constrains the coalition sizes up to a maximum limit, k ; however, it does not permit overlapping coalitions.

Overlapping coalitions in real-world situations can lead to better efficiency and reduced resource usage. In addition to Shehory and Kraus' approach, Zhang et al.'s particle swarm optimization based coalition formation algorithm also generates overlapping coalitions [16]. Particle swarm-based heuristic techniques mimic behaviors, such as bird-flocking, where a collection of potential problem solutions (*particles*) move through the problem search space according to domain specific mathematical models, ultimately aiming to converge towards the optimal solution.

Abdallah and Lesser's coalition formation algorithm utilizes an underlying organization hierarchy as the heuristic for computing coalitions in polynomial time [3]. The leaf nodes represent the robots and the non-leaf nodes represent computational units (managers) that control respective sub-trees of the organization, representing subsets of robots. The managers use Q-learning with neural networks to optimize and improve local decisions over time. Tošić and Agha [5] demonstrated a clique-based, distributed coalition formation algorithm that bounds the sizes of robot coalitions to a maximum upper limit, k . This algorithm uses a topology network representing inter-agent communications and generates modest sized robot

coalitions that form maximal cliques. This algorithm has low communication bandwidth requirements, given the sparse topology network.

Weerd et al.'s greedy, distributed coalition formation algorithm leverages an underlying social network derived from the inter-agent communication in order to compute coalitions [17]. The greedy algorithm finds task coalitions in $O(nm)$ run-time, requiring low $O(n^2m)$ communication messages for n tasks and m agents. However, despite the heuristic, the task allocation problem with node degree (agent connections) greater than three remains NP-complete.

Gaston and desJardins [18] proposed a similar greedy coalition formation algorithm using an agent-oriented network, where each agent in the network possesses a single skill set. Globally advertised tasks seeking agent coalitions are introduced at fixed time intervals for a fixed time duration. This greedy algorithm's performance is determined by two factors: (1) *global performance*, defined as a ratio of the number of successful task coalitions formed to the total number of tasks introduced, and (2) *local performance* of each agent. This algorithm generates time-extended task allocations.

Real-world tasks involve inter-task constraints (e.g., precedence dependencies), intra-task constraints (e.g., task deadlines), and spatial constraints that require robots to plan and schedule tasks. A two-stage, distributed coalition formation algorithm specific to unmanned aerial vehicles (UAVs) concentrates on: (1) minimizing coalition sizes, and (2) minimizing task completion time [19]. The UAV that discovers a given task becomes the task leader and broadcasts the task's resource requirements to all other UAVs. Addressing the spatial constraints in real-time, each UAV calculates its traveling cost by using Dubin's curve based on its location and the task location. The UAV leader computes small sized coalition of aerial vehicles to accomplish each task.

A communication less multi-agent task allocation algorithm was presented that allows agents to leverage their past experience, while making task assignments [20]. Campbell et al. adopt the multiprocessor scheduling problem for task allocations in order to address the intra-task constraints. The shortcoming of this approach is that only a single sized coalition is allocated for each task.

Koes et al.'s centralized anytime algorithm is based on Mixed Integer Linear Programming and addresses the real-world scheduling issues for search and rescue missions [21]. This algorithm finds near optimal solutions for problems involving path planning, online scheduling, and tasks with temporal and spatial constraints; however, task deadlines are not considered. Addressing this drawback, Ramchurn et al. [22] proposed coalition formation that considers spatial and temporal constraints, but determined the problem is NP-hard. Ramchurn et al. provided an optimal solution to the problem for small sized teams using Mixed Integer Programming. New anytime heuristics were devised to approximate solutions faster for larger problem instances.

The aforementioned greedy algorithms sacrifice coalition quality in order to reduce computational time. Approximation algorithms; however, guarantee solution quality and fall into two categories: (1) Anytime and (2) Design-to-time. Service and Adams [2] presented a dynamic programming based anytime algorithm that generates optimal coalitions for agents that are drawn from given fixed types. This algorithm runs in $O(nm)$ time for a team of homogeneous agents, and in $O(n^j m)$ time for heterogeneous agents of fixed j types (n robots and m tasks). Given the fixed categories of agents, this algorithm is applicable in multi-robot domains, where there is a high likelihood of similar robots in a team. Service and Adams extended Shehory and Kraus' heuristic approach by proposing two greedy approximate algorithms for multi-robot domains with bounded coalitions of size k [2]. The *Resource Model*-based algorithm is applicable to robots

that possess resources, such as camera, sonar, laser, etc., while the *Service Model*-based extension is presented for robots that perform services (e.g., sentry-duty, mapping, surveillance). Both algorithm generate coalitions within a constant factor from the optimal, in terms of total utility. Additionally, the *Service Model*-based algorithm offers excellent scalability improvement over Vig and Adams' algorithm [6] with a time complexity of $O(n^{3/2}m)$.

Unlike anytime algorithms, design-to-time algorithms must run to completion in order to generate optimal solutions, but have better worst case run-times. Service and Adams presented a hybrid r ($r > 2$) factor approximation algorithm that exploits the advantages of both the design-to-time and anytime algorithms to generate high quality coalition structures with better worst case run-time than $O(3^n)$ [23].

Market-based approaches offer effective and decentralized coalition formation. One of the earliest market-based distributed, fault-tolerant architectures is MURDOCH [10], an auction-based task allocation system that incorporates a resource centric publish/subscribe messaging protocol, where messages are addressed by *subject*. Task allocations are calculated via auction with the robot submitting the lowest cost bids for a task being awarded the task. A major limitation is MURDOCH's assumption that real-world tasks can be sub-divided into multiple single-robot, independent tasks. CoMutaR, a distributed fault-tolerant framework handles overlapping coalitions with low communication requirements [11]. The robots' capabilities are expressed as actions and each task is defined as a set of robot actions. The framework allows virtual sensor sharing by incorporating share-restricted resources. Single-round auctions are leveraged to compute the robot coalitions. Conventional auction-based algorithms require robots to bid for tasks; however, during complex missions, the global task information is often unavailable. Therefore, RACHNA [12] reverses this traditional bidding process and the tasks bid for robots using tasks' utilities. RACHNA exploits the resource redundancies in real robots to generate more computationally tractable coalitions and allows task preemption by rerunning the bidding process; however, RACHNA requires high communication bandwidth. RACHNA's ascending auction methodology introduces a drawback, when an on-going task can be reassigned to a different robot coalition, despite the current coalition's ability to complete the task. Service et al.'s recent simultaneous descending auction based task allocation procedure offers task preemption and addresses RACHNA's unnecessary task reassignment shortcoming [24].

Owing to the strengths and weaknesses of individual coalition formation algorithm, no single algorithm will be suitable for application to a wide spectrum of critical real-world mission situations. Therefore, *i-CiFHaR* incorporates a broad and expandable set of coalition formation algorithms in its library, such that the most appropriate subset of algorithm(s) can be selected dynamically in accordance with multiple mission constraints. The breadth of the algorithms considered for *i-CiFHaR*'s library include the reviewed algorithms that are broken into three clusters: (1) Greedy [2–6, 16–22], (2) Market/Auction-based [10–12, 24], and (3) Approximation [2, 23] approaches.

3 System design

i-CiFHaR is a three-tiered framework (see Fig. 1) that selects appropriate algorithm(s) to apply to given missions and provides decision support to human mission supervisors. *i-CiFHaR* integrates three tiers: (1) a User Interface, (2) the Middle Level Logic Tier, and (3) a Library of coalition formation algorithms.

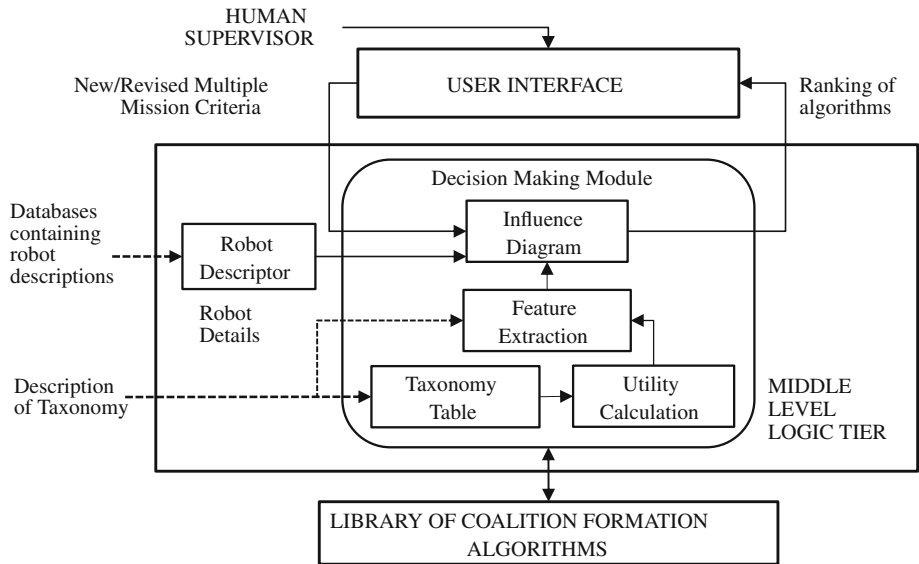


Fig. 1 The *i-CiFHaR* architecture

3.1 User interface

The Graphical User Interface (see *User Interface* in Fig. 1) will allow a human supervisor to provide task descriptions, environment constraints, and multiple mission criteria. The interface will provide: ongoing task progress (e.g., executing, waiting, finished), robot coalition and task allocations, individual robot status (e.g., engaged, idle, faulty), and newly discovered tasks.

3.2 Middle level logic tier

The *Middle Level Logic Tier* reasons over the mission requirements and constraints, the robotic assets' descriptions, and the library of coalition formation algorithms in order to probabilistically select the most appropriate algorithm(s) to apply. The *Middle Level Logic Tier* is the focus of this paper.

3.2.1 Decision making module

The primary component of the *Middle Level Logic Tier* is the *Decision Making Module* that chooses the most appropriate coalition formation algorithm(s) to apply to a given situation. This module classifies the coalition formation algorithms in *i-CiFHaR*'s library along multiple dimensions, or features based on an existing taxonomy [14]. The *Decision Making Module* is comprised of the: (1) *Taxonomy*, (2) *Utility Calculation*, (3) *Feature Extraction*, and (4) *Influence Diagram*. The *taxonomy table* stores the taxonomy features and the respective domain values that facilitate classification of the coalition algorithms. The *Utility Calculation* determines the feature-value pair utility scores that are essential for creating the influence diagram's utility table (see Sect. 3.2.1.2). *Feature Extraction* determines the most important features that discriminate the algorithms, thus reducing the problem dimensionality (see

Sect. 3.2.1.3). The *Influence Diagram* builds the system's influence diagram/decision network dynamically at run-time based on the extracted prominent features (see Sect. 3.2.1.4).

3.2.1.1 Taxonomy table *i*-CiFHaR employs Service and Adams' existing coalition formation algorithm taxonomy that defines multiple dimensions, or features for algorithm classification [14]. A number of multirobot taxonomies exist [25–28], each defining multiple dimensions/features along which algorithms can be classified. Gerkey and Mataric proposed one of the earliest and most widely used taxonomies that was designed for task allocation in multi-robot systems [28]; however, the taxonomy dimensions are highly abstracted and not broad enough to classify coalition algorithms effectively. Dudek et al.'s taxonomy presents a number of dimensions, some of which include the communication topology, number of robots, team composition, etc [26]. Cao et al.'s taxonomy provides the theoretical bases for cooperative multiple robots [25]. The dimensions include: (1) group composition, capturing the characteristics of the agents (homogeneous/ heterogeneous), (2) group architecture (centralized/decentralized), (3) communication structure of the agents, (4) modeling of other agents (aware/ unaware), (5) learning, which renders systems adaptive to environment dynamics, etc. Farinelli et al. proposed a hierarchical taxonomy comprised of two major dimensions (coordination and system) [27]. The coordination dimension, which captures the degree of agent cooperation is divided into three sub-classes: (1) knowledge, capturing agent awareness, (2) coordination, required in tightly coupled tasks, and (3) organization, denoting the centralized or distributed robot architecture. The system dimension, on the other hand is divided into four sub-classes: (1) communication topology, (2) team composition, (3) system architecture, representing the system's level of responsiveness towards environmental changes, and (4) team size.

Service and Adams' taxonomy encapsulates a broad set of features/dimensions for the multi-robot task allocation problem and borrows many of the dimensions from the aforementioned taxonomies. The taxonomy dimensions are partitioned into four relation-based categories: (1) agent, (2) task, (3) domain, and (4) algorithm. Service and Adams classified a number of coalition formation algorithms according to the taxonomy dimensions. Table 1 categorizes the taxonomy features into the four categories and highlights the respective domain values.

Let F be a set that contains the N taxonomy features ($N = 18$), where each feature has its respective non-empty domain set (see Table 1). Let Dom be a collective set containing all the respective domain sets of N taxonomy features. All this information is captured by Eq. 1, which is defined as:

$$\forall F_i \in F, \exists D_i \in Dom \mid 1 \leq i \leq N, D_i \neq \emptyset, \quad (1)$$

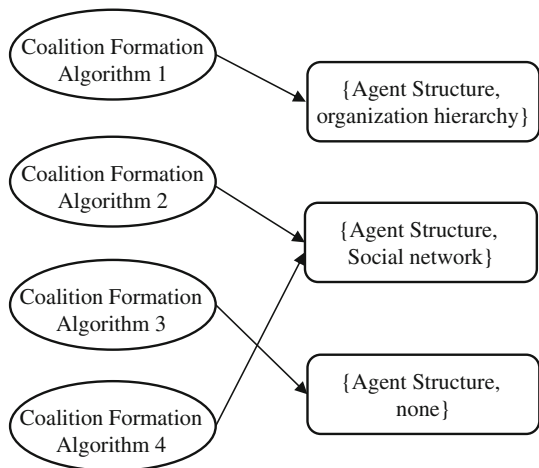
where D_i is the domain value set of feature F_i . A feature $F_i \in F$ can be instantiated with any particular value of its domain value set, D_i .

3.2.1.2 Utility calculation Influence diagrams contain *chance nodes* representing random variables, *decision nodes*, and a single *utility node*. The *utility node* has a utility value table (degree of preference) for all possible parent node configurations. The parents of the *utility node* in *i*-CiFHaR's influence diagram are: (1) a set of *chance nodes* representing the subset of important taxonomy features and (2) a *decision node* with its domain containing all the coalition formation algorithms. Since *i*-CiFHaR's built-in mathematical model computes the utility table entries, the utility scores of the feature-value pairs and the algorithms need to be determined. The utility calculation is based on *link analysis*, which has previously been used to capture the connections or associations in social networks among friends, computers in computer networks, webpages on the internet, etc. The exploration of link analysis in world wide web led to the notable applications: HITS [29] and PageRank [30]

Table 1 Taxonomy features and respective domain values

| Category | Taxonomy features (F) | Feature domain values (Dom) |
|-----------|--|--|
| Agent | Agent Orientation (F_1) | {Group Rational, Self-Interested} |
| | Agent Type (F_2) | {Homogeneous, Heterogeneous} |
| | Agent Capability Model (F_3) | {Resource, Service} |
| | Agent Awareness (F_4) | {Aware, Partially, Unaware} |
| | Agent Structure (F_5) | {Social Network, Organization Hierarchy, None} |
| Task | Inter-Task Constraints (F_6) | {Yes, Prerequisite, No} |
| | Task Preemption (F_7) | {Yes, No} |
| | Task Requirement Model (F_8) | {Resource, Service} |
| | Intra-Task Constraints (F_9) | {Yes, No} |
| | Task Coupling (F_{10}) | {Tightly, Loosely, Intermediate} |
| Domain | Performance Criterion (F_{11}) | {Maximize Utility, Minimize Cost, Maximize Task} |
| | Communication Overhead (F_{12}) | {High, Low} |
| | Task Allocation (F_{13}) | {Instantaneous, Time-Extended} |
| | Spatial Constraints (F_{14}) | {Yes, No} |
| | Overlapping Coalitions (F_{15}) | {Yes, No} |
| Algorithm | Algorithm Technique (F_{16}) | {Greedy, Auction-based, Approximation} |
| | Implementation (F_{17}) | {Centralized, De-Centralized} |
| | Coalition Size Constraint (F_{18}) | {Single, None, Fixed Upper Limit} |

Fig. 2 Link graph connecting four coalition formation algorithms to three feature-value pairs of the taxonomy feature, *Agent Structure*



that compute composite numerical scores for web pages with the intent of measuring their relative importance. Query web pages (called *hubs*) are linked to multiple query relevant web pages (called *authorities*) in a hyperlinked environment [29].

Each coalition formation algorithm can be linked to a subset of related feature-value pairs that govern the algorithm’s applicability (Fig. 2); therefore, the algorithms and the feature-value pairs can be visualized as *hubs* and *authorities*, respectively. Let V be a set of size d containing all possible feature-value pairs derived from the set, F of taxonomy features and their corresponding domain sets Dom , such that

$$V = \{(F_x, d_i) \mid F_x \in F, d_i \in D_x, D_x \in Dom\}. \tag{2}$$

The size of the feature-value pair set V is defined as $d = \sum_{x=1}^N |D_x|$, where $|D_x|$ represents the domain size of $F_x \in F$. A feature-value pair, $FVP_\phi \in V$ when associated with a particular taxonomy feature, $F_x \in F$ is represented as FVP_ϕ^x . Based on the associations between *i-CiFHaR*'s coalition formation algorithms and feature-value pairs, a link structure can be derived. For example, the taxonomy feature *Agent Structure* (F_5), with domain $D_5 = \{organization\ hierarchy, social\ network, none\}$, and $|D_5| = 3$ results in three possible feature-value pairs: (1) $\{Agent\ Structure, organization\ hierarchy\}$, (2) $\{Agent\ Structure, social\ network\}$, and (3) $\{Agent\ Structure, none\}$ (see Fig. 2). For instance, let *i-CiFHaR* incorporate four random algorithms (Coalition Formation Algorithms 1 through 4 in Fig. 2) and leverage only the single taxonomy feature *Agent Structure* for algorithm classification. The algorithms are connected manually to the respective feature-value pairs, thereby forming a link structure, as shown in Fig. 2. Building on this idea, *i-CiFHaR* generates a complete link graph between the coalition formation algorithms in the library and all possible feature-value pairs in set, V in accordance with Service and Adams' taxonomy.

Let C be the set containing p coalition formation algorithms in *i-CiFHaR*'s library. The complete link graph is represented by a bipartite directed graph $G(C, V, E)$, which is constructed using C and V as the two disjoint node sets of G . A directed edge $e_{lo} \in E$ from a coalition formation algorithm, $C_l \in C$ to a feature-value pair, $FVP_o \in V$ associates C_l with feature-value pair FVP_o .

Algorithm 1: Utility Calculation algorithm

```

Input:  $p \times d$  matrix,  $AMat$ ; constant,  $const$ ; Number of iterations,  $itr$ 
Output:  $FVPBaseScore$  of size  $1 \times d$ 
1:  $algoVector \leftarrow 1 \times p$  vector of 1s
2:  $FVPVector \leftarrow 1 \times d$  vector of 1s
3: for  $i = 1$  to  $itr$  do
4:  $FVPVector \leftarrow algoVector \times AMat$  // Vector-Matrix Multiplication
5:  $FVPVector \leftarrow \frac{FVPVector}{2-Norm(FVPVector)}$ 
6:  $algoVector \leftarrow FVPVector \times AMat^T$  // Vector-Matrix Multiplication
7:  $algoVector \leftarrow \frac{algoVector}{2-Norm(algoVector)}$ 
8: end for
9:  $FVPBaseScore \leftarrow const \times FVPVector$ 
10: Return  $FVPBaseScore$ 
    
```

Motivated by the HITS algorithm [29], the Utility Calculation algorithm (Algorithm 1) computes the base utility score of each feature-value pair. A $p \times d$ matrix, $AMat = \{a_{ij}\}$ is computed based on the complete link structure. The rows of $AMat$ represent the coalition formation algorithms, while the columns represent all possible feature-value pairs. An element, $a_{ij} \in AMat$ ($1 \leq i \leq p, 1 \leq j \leq d$) is defined as,

$$a_{ij} = \begin{cases} 1 & \text{if } C_i \in C \text{ is associated with } FVP_j \in V \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The link structure node weights are initialized to 1 and are updated iteratively until they converge to steady-state utility values. The convergence is guaranteed because the feature-value pair utility scores constitute the principal *Eigen vector* of $AMat^T \times AMat$ [29]. During each iteration of Algorithm 1, the vectors, $algoVector$ and $FVPvector$ are normalized using an Euclidean norm (2-Norm), such that $\sum_{i=1}^p algoVector_i^2 = 1$, and

$\sum_{j=1}^d FVPVector_j^2 = 1$. The constant, *const* scales the normalized utilities. The time complexity of Algorithm 1 is $O(itr \times p \times d)$ and the utilities converge to steady-state values very quickly, with $itr \approx 20$. The generated feature-value pair base utility scores are purely a function of the link structure. Given the feature-value pair set V containing all possible feature-value pairs (FVPs), the base utility scores ($FVPBaseScore$) are calculated using the Utility Calculation algorithm (Algorithm 1) and

$$\forall FVP_\phi \in V, \exists FVPBaseScore_\phi \in FVPBaseScore, | 1 \leq \phi \leq d, \tag{4}$$

where FVP_ϕ is the ϕ th feature-value pair. These base utility scores are weighted dynamically in accordance with the mission requirements (see Sect. 3.2.1.4).

3.2.1.3 Feature extraction i-CiFHaR leverages eighteen features; however, many features do not contribute to classifying the algorithms. For example, the feature *Agent Orientation* is assigned the same value (*Group Rational*) for all algorithms currently in the library. Feature extraction removes redundant features and reduces the problem dimensionality. *Principal Component Analysis* has been demonstrated for feature selection [31, 32]. *i-CiFHaR* extracts prominent features that discriminate between algorithms using a Feature Extraction algorithm similar to Song et al.’s approach for image processing [32]. *i-CiFHaR*’s feature selection algorithm differs in that it rejects all taxonomy features that result in zero coefficient factors across all major principal components. The procedure is formally proven in Lemma 1.

The selection algorithm leverages a $p \times N$ matrix U , where p is the number of algorithms in *i-CiFHaR*’s library and N is the number of taxonomy features. An element, $u_{ij} \in U$ is the base utility score (computed by Algorithm 1) for the specific feature-value pair, with feature j associated with algorithm i . *i-CiFHaR*’s feature selection algorithm requires $O(p \times N^2)$ time to generate the covariance matrix, $covC$ of U and uses the *Singular Value Decomposition* technique to compute the *Eigen vectors*; therefore, the total time complexity of *i-CiFHaR*’s feature selection algorithm is $O((p \times N^2) + N^3)$.

Each eigenvector accounts for some variance in the original data set and is expressed as a linear combination of the N taxonomy features. The κ th eigenvector, pc_κ is defined by

$$pc_\kappa = z_{\kappa 1}F_1 + z_{\kappa 2}F_2 + \dots + z_{\kappa N}F_N = \sum_{i=1}^N z_{\kappa i}F_i = FZ, \tag{5}$$

where F is the row feature vector of size N with $F_i \in F$ representing the i th taxonomy feature. Z is a matrix of size $N \times N$ containing the weight coefficients of all the N eigenvectors. The κ th column of Z consists of all the weight coefficients of the κ th eigenvector ($\kappa \in [1, N]$).

The primary statistics resulting from the κ th eigenvector constitute the associated variance (λ_κ) and the weight vector ($z_{\kappa 1}, z_{\kappa 2}, \dots, z_{\kappa N}$). The relative sizes of the coefficients ($z_{\kappa i}$) in the weight vector indicate the relative contributions of the corresponding feature, $F_i \in F$ in the original feature data set to the variance (λ_κ) of the eigenvector, pc_κ [33].

Lemma 1 *If a feature, $F_i \in F$ produces zero coefficient factors consistently for all the major principal components, then F_i contributes nothing towards the variance of the entire data set, σ_{data} .*

Proof Let N be the total number of taxonomy features, then $\sigma_{data} = \sum_{i=1}^N \sigma_i^2$, where σ_i^2 is the variance of the i th feature, F_i . Let λ_κ represents the variance of the eigenvector, pc_κ . The eigendecomposition generates eigenvector pc_κ , such that the variance of $pc_\kappa = \sum_{i=1}^N z_{\kappa i}F_i$ is maximized under the constraint, $\sum_{i=1}^N z_{\kappa i}^2 = 1$. The weight coefficient, $z_{\kappa i}$ also represents the correlation coefficient between feature F_i and the principal component

Table 2 Weight coefficients of the first six principal components or Eigen Vectors (EigVecx)

| | | Six Principal Components/Eigen Vectors | | | | | |
|------------------------------|------------------------|--|---------|---------|---------|---------|---------|
| | | EigVec1 | EigVec2 | EigVec3 | EigVec4 | EigVec5 | EigVec6 |
| Eigen Values | | 4.7 | 3.3 | 2.1 | 1.5 | 1.3 | 1 |
| Weight Coefficients | z₁ | 0 | 0 | 0 | 0 | 0 | 0 |
| | z₂ | 0 | 0 | 0 | 0 | 0 | 0 |
| | z₃ | 0.14 | 0.09 | 0.24 | 0.03 | 0.05 | 0.16 |
| | z₄ | 0 | 0 | 0 | 0 | 0 | 0 |
| | z₅ | 0.14 | 0.57 | 0.45 | 0.07 | 0.23 | 0.16 |
| | z₆ | 0.21 | 0.16 | 0.16 | 0.14 | 0.03 | 0.19 |
| | z₇ | 0.16 | 0.05 | 0.2 | 0.03 | 0.72 | 0.16 |
| | z₈ | 0.14 | 0.09 | 0.24 | 0.03 | 0.05 | 0.16 |
| | z₉ | 0.38 | 0.28 | 0 | 0.18 | 0.11 | 0.02 |
| | z_{10*} | 0 | 0 | 0 | 0 | 0 | 0 |
| | z₁₁ | 0.42 | 0.07 | 0.03 | 0.33 | 0 | 0.52 |
| | z₁₂ | 0.03 | 0.39 | 0.24 | 0.21 | 0.22 | 0.23 |
| | z₁₃ | 0.47 | 0.02 | 0.3 | 0.19 | 0.01 | 0.43 |
| | z₁₄ | 0.4 | 0.28 | 0.07 | 0.32 | 0.26 | 0.44 |
| | z₁₅ | 0.17 | 0.04 | 0.33 | 0.73 | 0.18 | 0.37 |
| | z₁₆ | 0.33 | 0.31 | 0.37 | 0.01 | 0.1 | 0.11 |
| | z₁₇ | 0 | 0.47 | 0.41 | 0.03 | 0.51 | 0.09 |
| | z₁₈ | 0.17 | 0.07 | 0.25 | 0.37 | 0 | 0.09 |

*Weight factors are very small (in order of 10^{-19}), and thus rounded to 0

pc_κ ; therefore, $z_{\kappa i} = 0$ means the angle between pc_κ and a unit vector along F_i is 90 degrees ($\because \cosine(90^\circ) = 0$); therefore, the vectors are orthogonal. $z_{\kappa i} = 0$ means no linear dependencies exist between F_i and pc_κ , and F_i does not contribute to the variance of pc_κ , because $\lambda_\kappa = \sum_{i=1}^N \sum_{j=1}^N z_{\kappa i} z_{\kappa j} \sigma_{ij}$. Since, $\sum_{i=1}^N \lambda_\kappa = \sum_{i=1}^N \sigma_i^2 = \sigma_{data}$; therefore, F_i does not contribute to the variance of the entire data set. \square

Eighteen *Eigen Vectors* or principal components are computed by *i-CiFHaR*'s feature extraction algorithm. Six components (the EigVecx in Table 2, where $1 \leq x \leq 6$) account for approximately 94% of the total variance in the original data set. These six principal components and their associated variances (Eigen Values) are enumerated in Table 2. Each principal component comprises a weight vector containing eighteen weight coefficients corresponding to the taxonomy features. The weight coefficients of the taxonomy features $F_1, F_2, F_4,$ and F_{10} are zero for all of the major principal components and do not contribute significantly to the classification (shaded gray in Table 2). The remaining features become *chance nodes* in the influence diagram.

3.2.1.4 Influence diagram construction Once the Feature Extraction algorithm identifies the most prominent features, the influence diagram is built dynamically at run-time. An influence diagram augments a Bayesian network by introducing decision variables and a utility function that characterizes the decision maker's (here *i-CiFHaR*) preferences. *i-CiFHaR* solves the decision problem by determining the optimal strategy that maximizes the expected utility score for the framework.

The prominent taxonomy features are the most influential and uncertain criteria that can be leveraged to discriminate between the coalition formation algorithms. The influence diagram's *decision node* contains the decision alternatives that are mutually exclusive, finite, and exhaustive. Since *i-CiFHaR* seeks to select the most optimal algorithm(s) to apply at a specific time, the *decision node*'s domain consists of all the coalition formation algorithms in its library. The random *chance nodes* and the *decision node* become the parents of the

single *utility node*, which holds a utility table for all possible configurations of the parent nodes. During a real-world mission scenario, the incomplete information regarding the situation is captured in terms of probability values for each of the *chance nodes*. Given that all the *chance nodes* and *decision node* are parents of the *utility node*; the utility function represents all the taxonomy features and the algorithm scores. The utility table values are usually obtained by consulting domain experts or through intuition and preferences of the system designer [34]; however, *i-CiFHaR* calculates the utility table entries automatically. The number of entries is exponential in size to the number of parents to the *utility node*. The influence diagram’s utility table size (U_{size}) is given by:

$$U_{size} = p \times \prod_{x=1}^{N_{extr}} |D_x|, D_x \in Dom, F_x \in F_{Prom}, \tag{6}$$

where N_{extr} is the number of extracted taxonomy features; D_x is the domain set of feature F_x ; and F_{Prom} is the Prominent Feature Set containing all prominent taxonomy features, with $|F_{Prom}| = N_{extr}$.

The exponential utility table size prohibits calculating the entries based on designer preferences or intuitions. Two approaches are implemented. First, the most prominent features are used to construct the influence diagram, reducing the problem dimension. Second, a mathematical model automatically generates the utility table entries by leveraging the base utility scores of the feature-value pairs, as computed by the *Utility Calculation* module. However, a direct use of the base scores for the utility table entry calculation, as demonstrated in a previous implementation [35] has a major drawback. The link-analysis algorithm computes the feature-value pair utility scores based on the link structure; thus, the more in-links to a feature-value pair, the higher its score. The feature-value pairs with low endorsements, i.e., a feature-value pair not associated with many algorithms receives a very low utility score. When feature-value pairs with low utility scores are required for a mission, the corresponding coalition algorithm is often not selected because *i-CiFHaR* seeks to maximize the system’s expected utility score. Moreover, the previous implementation [35] of *i-CiFHaR* incorporates an influence diagram with a persistent utility function that makes it susceptible to inconsistencies in decision making. Addressing the stated drawbacks, *i-CiFHaR*’s presented implementation employs an adaptive utility function. Under the neoclassical approach, only the probabilities of the *chance nodes* vary with new information; however, no such provision exists for modifying the decision maker’s preferences. Therefore, adaptive utility functions have been advocated [36–38] and it has been shown that the expected utility hypothesis still holds [39].

An adaptive utility function through *dynamic* scoring of the feature-value pairs addresses the aforementioned drawbacks, where the base utility scores (computed by Eq. 4) are dynamically weighted in accordance with the mission requirements. This approach renders *i-CiFHaR* more responsive to real-world mission scenarios. Each mission is described in terms of the feature-value pair assignments (*FVP*) for the prominent features. The mission uncertainties are captured using probability values (*Pr*) for each feature-value pair assignment. The mission dependent utility score ($FVPMissionScore_{\zeta}^i$) of the ζ th feature-value pair ($FVP_{\zeta} \in V$) that corresponds to the feature $F_i \in F_{Prom}$ is defined by:

$$\forall FVP_{\zeta} \in V, FVPMissionScore_{\zeta}^i = \exp^{\{\alpha \times \beta \sqrt{|Pr_{\zeta} - avg_i|}\}} \times FVPBaseScore_{\zeta}^i, \tag{7}$$

where $\alpha = sgn(Pr_{\zeta} - avg_i)$ is the signum function, $\beta = |D_i|$ is the domain size of the feature $F_i \in F_{Prom}$, and $avg_i = 1/\beta$ is the equal likelihood of F_i being assigned to any one of its domain values. The probability assigned to the feature-value pair is denoted by

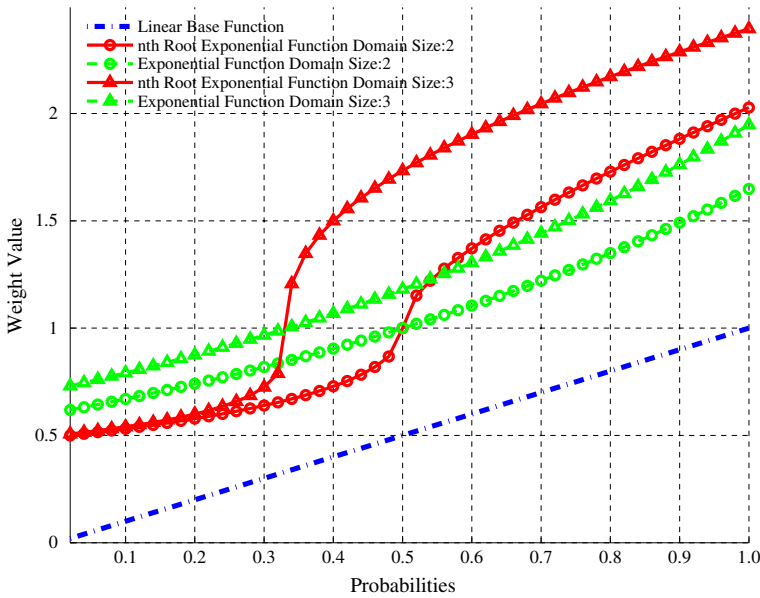


Fig. 3 *i-CiFHaR*'s Dynamic Weighting Function. For illustration purposes, two pairs of curves are highlighted: one for a feature with domain size = 2 (shown with circle markers), and the other for a feature with domain size = 3 (shown with triangle markers). *i-CiFHaR*'s weighting function (in red) is compared to an exponential function (in green) and a base line linear function (in blue) for each of the features. It is noted that for the random features with domain sizes 2 and 3, the corresponding curves intersect at weight value = 1 for probabilities 0.5 and 0.33, respectively. Under such circumstances, the base utility scores of the feature-value pairs are not scaled, because each of the possible values in the features' domains has equal likelihood of being selected (Color figure online)

Pr_{ζ} . The motivation for leveraging the aforementioned scaling approach stems from the fact that the mutual exclusion property of each taxonomy feature's domain values permits the calculation of the deviation for a particular feature-value pair $FVP_{\zeta} \in V$ from avg_i of the corresponding feature, given the mission criteria. Based on the *n*th-root exponential function (Eq. 7), the weighting factor is greater than 1 when the deviation is positive and when the deviation is negative, the base feature-value pair score is weighted by a factor < 1 . Figure 3 illustrates the effectiveness of *i-CiFHaR*'s *n*th-root exponential weighting function, when compared to a conventional exponential function ($exp\{Pr_{\zeta} - avg_i\}$) and a base linear function. The figure demonstrates that for the two exemplary features with domain sizes two and three, *i-CiFHaR*'s weighting function generates a larger variation in the weight factors of the domain values, given their probabilities.

Consider a single taxonomy feature, $F_i = Task\ Preemption$ with the domain set $D_i = \{Yes, No\}$. The mean assignment value (avg_i), assuming equal probability for assigning F_i to one of its two domain values, is $avg_i = 0.5$. For example, let a mission require the feature-value pair $\{Task\ Preemption, Yes\}$. Assume a high confidence in this assignment, then the assignment probability value, $Pr_{\zeta} = 0.8$. The base utility score of $\{Task\ Preemption, Yes\}$ is very low, as computed by the Utility Calculation algorithm (Algorithm 1), because only two coalition formation algorithms permit task preemption. The low utility score increases the likelihood of false positives; however, if the mission requirements are considered, the low utility score is weighted by the factor $exp\{1 \times \sqrt[2]{0.8 - 0.5}\} = 1.73$. Consider another mission

scenario in which task preemption is not required. Let the probability value Pr_{ζ} be 0.2. The mission dependent utility score is weighted by $\exp\{-1 \times \sqrt[3]{0.2-0.5}\} = 0.57$.

Once the mission dependent feature-value pair utility scores are generated, each coalition formation algorithm is assigned a utility score. The intermediate utility score is derived using only the associated feature-value pairs that belong to the prominent taxonomy features, as calculated by:

$$\forall C_l \in C, algoScore_l = \sum_{\zeta} a_{l\zeta} \times FVPMissionScore_{\zeta}^i \quad | \quad 1 \leq l \leq p, F_i \in F_{Prom}, FVP_{\zeta}^i \in V, \tag{8}$$

where $a_{l\zeta} \in AMat$ and p is the number of coalition formation algorithms in *i-CiFHaR*'s library. Equation 8 leverages the adjacency matrix, *AMat* that captures the associations between the algorithm and the feature-value pairs (Eq. 3). The intermediate utility scores depend on the complete link structure used by Algorithm 1 to calculate the base utility scores. Once the algorithms' intermediate utility scores are calculated, the final mission dependent utility scores are obtained by normalizing the intermediate scores:

$$\forall C_l \in C, algoMissionScore_l = const \times \frac{algoScore_l}{2 - Norm(algoScore)} \quad | \quad 1 \leq l \leq p, \tag{9}$$

where $2 - Norm(algoScore) = \sqrt{\sum_{i=1}^p algoScore_i^2}$ is the Euclidean norm leveraged for the normalization (similar to that in Algorithm 1). The constant, *const* scales the normalized mission dependent utility scores and is set to the same value as that in Algorithm 1. The dynamic utility scores are mission specific, which result in *i-CiFHaR* being more adaptable to a wide-range of real-world missions.

Based on the mission dependent utility scores of the coalition algorithms and the feature-value pairs, *i-CiFHaR*'s adaptive mathematical model generates the utility table entries automatically for the influence diagram, as defined by:

$$\forall St_v \in W, U(St_v|Act_i) = algoMissionScore_i \times \sum_{j=1}^{N_{extr}} a_{i\zeta} \times FVPMissionScore_{\zeta}^j, \tag{10}$$

where $algoMissionScore_i$ is the mission dependent utility score of the *i*th coalition formation algorithm (defined by Eq. 9) and $FVPMissionScore_{\zeta}^j$ is the ζ th feature-value pair associated with the *j*th prominent feature, $F_j \in F_{Prom}$ (defined by Eq. 7). Each state of the world, $St_v \in W$ is defined in terms of feature-value pairs of the prominent taxonomy features. The mathematical model (Eq. 10) requires a $p \times d$ adjacency matrix, $AMat = a_{i\zeta}$, where $a_{i\zeta} \in AMat$ is defined by Eq. 3. The number of prominent taxonomy features extracted by *i-CiFHaR*'s *Feature Extraction* module is N_{extr} . *i-CiFHaR*'s utility function, $U(St_v|Act_i)$ maps from every state St_v of the hypothetical world W to a value, when an action Act_i is taken. Act_i indicates that the *decision node* chooses the *i*th coalition formation algorithm, $C_i \in C$.

The *i-CiFHaR* framework behaves as a self-interested decision making agent and considers all the possible hypothetical states of the world as outcomes (χ) of a lottery. According to microeconomic utility theory, such a rational agent models its interest by quantifying each possible outcome using a utility/reward value that captures the agent's preference. A utility function, $U : \chi \rightarrow \mathbb{R}$ maps from the world states to real numbers. Given a particular coalition formation algorithm, *i-CiFHaR* has a preference for each of the possible world states; therefore, $\forall (St_i, St_j) \in W, \exists St_i > St_j$, or $St_j > St_i$ or $St_i \sim St_j$, where $>$

denotes strict preference, while \sim denotes indifference. This completeness of *i-CiFHaR*'s preferences over all possible world states represents that *i-CiFHaR* either strictly prefers one state to the other, or is indifferent between the two, given a particular coalition formation algorithm. Moreover, the preference is transitive, i.e., if $St_i > St_j$ and $St_j > St_k$, then $St_i > St_k$. Since all the possible world states represent outcomes of a lottery, there exists a probability distribution over the states expressed as $[p_1 : St_1, p_2 : St_2, \dots, p_k : St_k]$, where $\sum_k p_k = 1$. The von Neumann–Morgenstern expected utility model [40] states that if an agent's preference relation satisfies completeness, monotonicity, and transitivity, then there exists a utility function that satisfies: (1) $U(St_1) > U(St_2)$, iff $St_1 > St_2$, and (2) $EU([p_1 : St_1, p_2 : St_2, \dots, p_k : St_k]) = \sum_k p_k \times U(St_k)$, where $EU(\bullet)$ denotes the *Expected Utility* of a given lottery/gamble.

i-CiFHaR's utility function (Eq. 10) maps every possible world state to a preference utility score $\in \mathbb{R}_+$, given the algorithms. Seeking to solve the multicriteria decision problem at hand, each state $St_v \in W$ is expressed in terms of feature-value pairs of the taxonomy features; therefore, St_v is a conjunction of $F_i \in F_{Prom}$. Additionally, the probabilities associated with each state, St_v are translated into the joint probability distribution over all the prominent taxonomy features.

Theorem 1 *i-CiFHaR's utility function represents its preferences over all possible choices.*

Proof Equation 10 defines $U(St_v|Act_i)$, which calculates the utility value of every possible world state $St_v \in W$, given an algorithm choice, Act_i . The base utility score ($FVPBaseScore_{\zeta}^j$ as computed by Algorithm 1) of each feature-value pair is *const* (See Algorithm 1) times the weight coefficient of the corresponding pair in the principal *Eigenvector* of $AMat^T \times AMat$, where $AMat$ is the adjacency matrix representing the connections in the link graph. Each algorithm's utility score is derived from Eqs. 8 and 9 and is given by $algoMissionScore_i = k \times \sum_{\zeta} a_{i\zeta} \times FVPMissionScore_{\zeta}^j$, where $k = \frac{const}{2-Norm(algoScore)}$ is a constant, and j corresponds to $F_j \in F_{Prom}$. $FVPBaseScore_{\zeta}^j$ is probabilistically scaled to achieve $FVPMissionScore_{\zeta}^j$; therefore, the algorithm utility score, $algoMissionScore_i$ is constant (\mathcal{K}_i) for a given mission instance. *i-CiFHaR*'s utility function can be re-written as $U(St_v|Act_i) = \mathcal{K}_i \times \sum_{j=1}^{N_{extr}} a_{i\zeta} \times FVPMissionScore_{\zeta}^j$, where $a_{i\zeta}$ is 1 if algorithm Act_i is connected to ζ th feature-value pair in the link graph, 0 otherwise. The world states in W are expressed by every possible configuration of the prominent taxonomy features. A state, St_{α} receives a higher utility than that of St_{β} , only if the former contains the conjunction of feature-value pairs that are connected to Act_i in the link graph. A valid utility function depends on the ordinality, rather than cardinality of the states; therefore, *i-CiFHaR*'s preferences over possible world states are correctly modeled by $U(St_v|Act_i)$, which generates $U(St_{\alpha}) > U(St_{\beta})$, if and only if $St_{\alpha} > St_{\beta}$ for algorithm Act_i . □

Once the utility table entries are generated (Eq. 10), *i-CiFHaR* leverages the influence diagram to optimize the algorithm selection process by maximizing its expected utility score. The expected utility score ($EU(Act_i)$) for each algorithm decision (Act_i) is:

$$EU(Act_i) = \sum_{v=1}^{|states|} Pr(St_v|Act_i) \times U(St_v|Act_i), \tag{11}$$

where *states* is the subset of all possible hypothetical world states in the hypothetical world W , where the action Act_i can be selected; $U(St_v|Act_i)$ is the utility of the particular world state $St_v \in states$ is derived from the network's utility table (Eq. 10). *i-CiFHaR* selects the

most appropriate algorithm (Act^*) to apply to a given mission scenario that maximizes the expected utility score i.e.,

$$Act^* = \operatorname{argmax}_{Act_i \in C} EU(Act_i), \quad (12)$$

where C is the set containing p coalition formation algorithms.

The objective is to provide decision support to a human mission supervisor; therefore, *i-CiFHaR* may select a subset of algorithm(s) most applicable to the mission scenario when a single algorithm does not meet all mission requirements. This set of algorithm(s) includes Act^* and all other algorithms with expected utility scores greater or equal to the threshold (EU^*), defined by:

$$EU^* = \gamma \times \max_{Act_i \in C} EU(Act_i), \quad (13)$$

where γ is the desired fraction of the maximum expected utility score that is required by the mission supervisor. A supervisor requiring 100 % performance will obtain the best algorithm that has the maximum expected utility score, while a 90 % performance will select all algorithms that have their expected utility scores greater than $EU^* = 0.9 \times \max_{Act_i \in C} EU(Act_i)$.

3.3 Library of Algorithms

Nineteen coalition formation algorithms have been selected to be incorporated into *i-CiFHaR*'s library that are associated with all possible feature-value pairs generated from the taxonomy features. This broad collection of algorithms increases the likelihood of *i-CiFHaR* applying to a wide spectrum of missions based on the taxonomy features. The algorithms have been classified into three major categories (greedy, approximation, and auction-based) and are associated with their respective feature-value pairs corresponding to the fourteen prominent features extracted by Algorithm 2 (see Table 3).

4 Experiments

An evaluation assessed the efficiency of selecting appropriate coalition formation algorithm(s) based on multiple mission criteria and constraints. The current *i-CiFHaR* framework has been implemented on a Linux platform (Ubuntu-12.04, 64-bit) with an Intel Core i5, 2.30 GHz processor using C++ and Qt framework (version 4.8) [41]. Five algorithms (A_{15} through A_{19} in Table 3) are not yet implemented. The purpose of this experiment is to evaluate the *Decision Making Module*'s ability to choose appropriate coalition formation algorithms, not actually forming the coalitions, thus it is only necessary to characterize the algorithm's capabilities and parameters for purposes of this evaluation. The influence diagram implementation leverages the Netica-C API [42], a Bayesian network development software tool that uses a junction tree algorithm to evaluate influence diagrams. Twenty-four missions were created and the *hypothesis* is that *i-CiFHaR* will select a set of most suitable algorithm(s) to apply for each mission scenario by maximizing the system's expected utility score.

4.1 Experimental design

The total number of possible mission scenarios is 124,416. Many of these mission scenarios include feature-value pairs that are not realizable for real-world scenarios. The twenty-four mission scenarios delineated in Table 4 represent the subset of realistic situations that were used to evaluate *i-CiFHaR*'s algorithm selection process.

Each mission scenario has a set of feature-value pair assignments for each prominent feature and a probability value. The twenty-four scenarios were simulated in two ways. First, the domain value assigned to each prominent feature was altered, denoted by the Feature-Value pair assignments (*FVP*) in Table 4. Second, the uncertainty related to each mission

Table 3 Taxonomy Features versus Coalition Formation Algorithms (A_1 – A_{19}). See Table 1 for feature domains. Please refer to the key after the last subtable

| Taxonomy features | Coalition formation algorithms | | | | | | |
|-------------------|--------------------------------|--------|---------|--------|--------|--------|--------|
| | A_1 | A_2 | A_3 | A_4 | A_5 | A_6 | A_7 |
| | Algorithm category | | | | | | |
| | Greedy | Greedy | Auction | Greedy | Greedy | Greedy | Greedy |
| F_3 | Res | Res | Ser | Res | Res | Res | Res |
| F_5 | None | None | None | Org | SNet | SNet | None |
| F_6 | PReq | PReq | PReq | PReq | No | No | PReq |
| F_7 | No | No | Yes | No | No | No | No |
| F_8 | Res | Res | Ser | Res | Res | Res | Res |
| F_9 | No | No | No | No | No | No | Yes |
| F_{11} | MC | MC | MU | MU | MU | MU | MT |
| F_{12} | H | H | H | L | L | L | L |
| F_{13} | IA | IA | IA | TE | IA | TE | IA |
| F_{14} | No | No | No | No | No | No | No |
| F_{15} | Yes | No | No | No | No | No | No |
| F_{16} | Gr | Gr | Auc | Gr | Gr | Gr | Gr |
| F_{17} | DC | DC | DC | DC | DC | DC | DC |
| F_{18} | k | k | None | None | k | None | Sngl |

| Taxonomy features | Coalition formation algorithms | | | | | |
|-------------------|--------------------------------|--------|----------|----------|----------|----------|
| | A_8 | A_9 | A_{10} | A_{11} | A_{12} | A_{13} |
| | Algorithm category | | | | | |
| | Greedy | Greedy | Auction | Greedy | Approx | Approx |
| F_3 | Res | Res | Res | Ser | Ser | Ser |
| F_5 | None | None | None | None | None | None |
| F_6 | No | PReq | PReq | PReq | No | No |
| F_7 | No | No | No | No | No | No |
| F_8 | Res | Res | Res | Ser | Ser | Ser |
| F_9 | No | No | No | No | No | No |
| F_{11} | MT | MU | MU | MU | MU | MU |
| F_{12} | L | H | L | H | H | H |
| F_{13} | TE | IA | IA | IA | IA | IA |
| F_{14} | Yes | No | No | No | No | No |
| F_{15} | No | No | No | No | No | No |
| F_{16} | Gr | Gr | Auc | Gr | Approx | Approx |
| F_{17} | DC | DC | DC | DC | C | C |
| F_{18} | None | k | Sngl | k | None | None |

Table 3 continued

| Taxonomy features | Coalition formation algorithms | | | | | |
|-------------------|--------------------------------|----------|----------|----------|----------|----------|
| | A_{14} | A_{15} | A_{16} | A_{17} | A_{18} | A_{19} |
| | Algorithm category | | | | | |
| | Auction | Greedy | Greedy | Greedy | Auction | Greedy |
| F_3 | Ser | Ser | Ser | Ser | Ser | Res |
| F_5 | None | SNet | None | None | None | None |
| F_6 | PReq | PReq | Yes | Yes | PReq | PReq |
| F_7 | Yes | No | No | No | No | No |
| F_8 | Ser | Ser | Ser | Ser | Ser | Res |
| F_9 | No | No | Yes | Yes | No | No |
| F_{11} | MU | MT | MU | MT | MU | MU |
| F_{12} | H | L | H | H | L | H |
| F_{13} | TE | TE | TE | TE | IA | IA |
| F_{14} | No | No | Yes | Yes | No | No |
| F_{15} | No | No | No | No | Yes | Yes |
| F_{16} | Auc | Gr | Gr | Gr | Auc | Gr |
| F_{17} | DC | DC | C | C | DC | DC |
| F_{18} | None | None | None | None | None | None |

Feature Domain Key

| | | |
|----------------------|-----------------------------|-------------------------------|
| Approx:Approximation | k :Bounded Size | Res:Resource-Model |
| Auc:Auction-based | L:Low Communication | Ser:Service-Model |
| C:Centralized | MC:Minimize Cost | SNet:Social Network |
| DC:Decentralized | MT:Maximize Tasks Completed | Sngl: Single Sized Coalitions |
| Gr:Greedy | MU:Maximize Utility | TE:Time-Extended |
| H:High Communication | Org:Organization Hierarchy | |
| IA:Instantaneous | PReq:Prerequisite | |

Algorithm Key

| | |
|---|---|
| A_1 :Shehory and Kraus [4] | A_2 :Vig and Adams [6] |
| A_3 : Vig and Adams [12] | A_4 :Abdallah and Lesser [3] |
| A_5 : Tošić and Agha [5] | A_6 :Weerd et al. [17] |
| A_7 :Campbell et al. [20] | A_8 :Sujit et al. [19] |
| A_9 :Service and Adams [2]-Resource Model | A_{10} :Gerkey and Matarić [10] |
| A_{11} :Service and Adams [2]-Service Model | A_{12} :Service and Adams [23]-Approximation |
| A_{13} :Service and Adams [2]-Dynamic Programming Agent-Types | A_{14} :Service et al. [24]-Simultaneous Descending |
| A_{15} :Gaston and desJardins [18] | A_{16} :Koes et al. [21] |
| A_{17} :Ramchurn et al. [22] | A_{18} :Shiroma and Campos [11] |
| A_{19} :Zhang et al. [16] | |

Table 3 continued**Taxonomy Feature Key***F*₃: Agent Capability Model*F*₆: Inter-Task Constraints*F*₈: Task Requirement Model*F*₁₁: Performance Criterion*F*₁₃: Task Allocation*F*₁₅: Overlapping Coalitions*F*₁₇: Algorithm Implementation*F*₅: Agent Structure*F*₇: Task Preemption*F*₉: Intra-Task Constraints*F*₁₂: Communication Overhead*F*₁₄: Spatial Constraints*F*₁₆: Algorithm Technique*F*₁₈: Coalition Size Constraint

was varied by varying the probability value associated with each feature-value pair, denoted by Pr in Table 4. System users can establish domain and mission appropriate probabilities based on domain knowledge, prior mission deployments, intelligence, etc. Feature domain values are mutually exclusive; therefore, the sum of the feature-value probability assignments is 1. Additionally, the mission scenarios were partitioned into clusters (see Table 5), such that within each cluster, certain taxonomy feature(s) were instantiated to constant domain values, while the remaining features were altered in order to distinguish between each of the grouped missions.

Let us consider two mission scenarios, MS_1 and MS_3 in order to illustrate the simulation of two different real-world situations. Both missions are based on the *resource model*, where the robots' capabilities and tasks' requirements are described in terms of resources (e.g., camera, sonar, laser). Therefore, the features *Agent Capability* and *Task Requirements* were assigned to *Resource* with a high probability value, 0.8 for both missions. However, the missions differ in many aspects. MS_1 's objective is to maximize utility, thus *Performance* is assigned *Maximize Utility (MU)* with a probability of 0.6. Conversely, MS_3 seeks to minimize cost, thus *Performance* is assigned to *Minimize Cost (MC)* with a high probability of 0.9. Additionally, MS_1 does not require overlapping coalitions, thus *Overlapping* is set to *No* with probability 0.8. MS_3 seeks to reduce resource losses; thus, requiring overlapping coalitions, so *Overlapping* is set to *Yes* with a high likelihood of 0.8.

Each mission scenario differs in terms of mission criteria (defined by feature-value assignments and probability values). An exhaustive set of missions cannot be evaluated, thus the twenty-four missions represent a good subset of potential real-world scenarios focused on the prominent taxonomy features and their respective domain sets. The impact of *const* was assessed by varying its value from 25 to 250, in increments of 25; however, this change in value resulted in no variance in the algorithm rankings (Table 7). Therefore, the Utility Calculation algorithm (Algorithm 1) and Eq. 9 *const* value was set to 100 as a designer choice. The variable, γ in Eq. 13 was set to 90%, based on designer selection.

4.2 Experimental results

i-CiFHaR selects a subset of the most appropriate algorithms for each mission scenario by optimizing the expected utility score. Table 6 presents all the coalition formation algorithms and the missions for which each algorithm was chosen. Table 7 ranks the most appropriate algorithms for each mission scenario, where algorithms are ordered by decreasing expected utility scores and the algorithm with the highest score is deemed the most appropriate. A high expected utility score indicates the corresponding algorithm's ability to satisfy the mission's criteria. The subset of the most applicable algorithm(s) is determined by the cutoff threshold, which is a 90% lower bound of the maximum expected utility score. For instance, *i-CiFHaR* selects Service and Adams' heuristic algorithm (A_9) for MS_1 with the maximum

expected utility score of 9467.4 as the most suitable algorithm (see Table 7). Additionally, the framework selects two additional algorithms with expected utility scores higher than the threshold, $EU^* = 0.9 \times 9467.4 = 8520.67$.

The experimental results show that *i-CiFHaR* selects appropriate algorithms for each mission scenario. This section discusses the mission scenario results and justifies the selection of the particular algorithms.

Mission scenario 1 (MS_1 in Table 4) simulated a mission focused on maximizing total utility that consisted of independent tasks requiring small sized coalitions. All the mission’s

Table 4 Mission Scenarios (MS_1 – MS_{24}) characterized by Feature-Value Pairs. Each mission is defined in the format (Feature-value assignments (*FVP*), Assignment Probability (*Pr*))

| Taxonomy features | Mission scenarios | | | | | |
|-------------------|-------------------|----------|----------|----------|----------|----------|
| | MS_1 | MS_2 | MS_3 | MS_4 | MS_5 | MS_6 |
| F_3 | Res,0.8 | Res,0.8 | Res,0.8 | Res,0.8 | Ser,0.6 | Res,0.8 |
| F_5 | None,0.8 | None,0.8 | None,0.8 | None,0.8 | None,0.8 | SNet,0.6 |
| F_6 | PReq,0.7 | PReq,0.7 | PReq,0.7 | PReq,0.7 | PReq,0.7 | No,0.6 |
| F_7 | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 |
| F_8 | Res,0.8 | Res,0.8 | Res,0.8 | Res,0.8 | Ser,0.6 | Res,0.7 |
| F_9 | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 |
| F_{11} | MU,0.6 | MC,0.9 | MC,0.9 | MU,0.7 | MU,0.7 | MU,0.7 |
| F_{12} | H,0.8 | H,0.8 | H,0.8 | H,0.8 | L,0.7 | L,0.9 |
| F_{13} | IA,0.7 | IA,0.7 | IA,0.7 | IA,0.7 | IA,0.7 | IA,0.6 |
| F_{14} | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 | No,0.8 |
| F_{15} | No,0.8 | No,0.8 | Yes,0.8 | Yes,0.9 | Yes,0.9 | No,0.8 |
| F_{16} | Gr,0.7 | Gr,0.7 | Gr,0.7 | Gr,0.7 | Auc,0.6 | Gr,0.7 |
| F_{17} | DC,0.8 | DC,0.8 | DC,0.8 | DC,0.8 | DC,0.8 | DC,0.8 |
| F_{18} | k ,0.6 | k ,0.6 | k ,0.6 | None,0.7 | None,0.7 | k ,0.5 |

| Taxonomy features | Mission scenarios | | | | | |
|-------------------|-------------------|-----------|-----------|-----------|-----------|-------------|
| | MS_7 | MS_8 | MS_9 | MS_{10} | MS_{11} | MS_{12} |
| F_3 | Res, 0.8 | Ser, 0.7 | Res, 0.8 | Ser, 0.7 | Ser, 0.7 | Ser, 0.7 |
| F_5 | SNet, 0.6 | SNet, 0.8 | Org, 0.7 | None, 0.7 | None, 0.7 | None, 0.7 |
| F_6 | No, 0.7 | No, 0.5 | PReq, 0.6 | PReq, 0.7 | PReq, 0.7 | No, 0.6 |
| F_7 | No, 0.8 | No, 0.8 | No, 0.8 | Yes, 0.9 | Yes, 0.9 | No, 0.8 |
| F_8 | Res, 0.7 | Ser, 0.7 | Res, 0.7 | Ser, 0.8 | Ser, 0.8 | Ser, 0.8 |
| F_9 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 |
| F_{11} | MU, 0.7 | MT, 0.6 | MU, 0.8 | MU, 0.8 | MU, 0.8 | MU, 0.8 |
| F_{12} | L, 0.9 | L, 0.9 | L, 0.8 | H, 0.8 | H, 0.8 | H, 0.8 |
| F_{13} | TE, 0.7 | TE, 0.8 | TE, 0.8 | IA, 0.6 | TE, 0.8 | IA, 0.7 |
| F_{14} | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 |
| F_{15} | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 |
| F_{16} | Gr, 0.7 | Gr, 0.7 | Gr, 0.7 | Auc, 0.8 | Auc, 0.8 | Approx, 0.8 |
| F_{17} | DC, 0.8 | DC, 0.8 | DC, 0.8 | DC, 0.8 | DC, 0.8 | C, 0.6 |
| F_{18} | None, 0.5 | None, 0.6 | None, 0.5 | None, 0.8 | None, 0.8 | None, 0.8 |

Table 4 continued

| Taxonomy features | Mission scenarios | | | | | |
|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| | <i>MS</i> ₁₃ | <i>MS</i> ₁₄ | <i>MS</i> ₁₅ | <i>MS</i> ₁₆ | <i>MS</i> ₁₇ | <i>MS</i> ₁₈ |
| <i>F</i> ₃ | Ser, 0.9 | Res, 0.8 | Ser, 0.7 | Ser, 0.7 | Res, 0.8 | Ser, 0.9 |
| <i>F</i> ₅ | None, 0.7 | None, 0.7 | None, 0.7 | None, 0.7 | None, 0.7 | None, 0.7 |
| <i>F</i> ₆ | PReq, 0.6 | No, 0.7 | Yes, 0.9 | Yes, 0.9 | PReq, 0.6 | Yes, 0.7 |
| <i>F</i> ₇ | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | Yes, 0.7 |
| <i>F</i> ₈ | Ser, 0.9 | Res, 0.8 | Ser, 0.7 | Ser, 0.7 | Res, 0.8 | Ser, 0.9 |
| <i>F</i> ₉ | No, 0.8 | No, 0.5 | Yes, 0.8 | Yes, 0.8 | No, 0.7 | Yes, 0.8 |
| <i>F</i> ₁₁ | MU, 0.8 | MT, 0.9 | MU, 0.6 | MT, 0.7 | MU, 0.6 | MU, 0.7 |
| <i>F</i> ₁₂ | H, 0.8 | L, 0.9 | H, 0.6 | H, 0.6 | L, 0.8 | H, 0.8 |
| <i>F</i> ₁₃ | IA, 0.7 | TE, 0.6 | TE, 0.8 | TE, 0.8 | IA, 0.7 | TE, 0.8 |
| <i>F</i> ₁₄ | No, 0.8 | Yes, 0.6 | Yes, 0.8 | Yes, 0.8 | No, 0.8 | Yes, 0.8 |
| <i>F</i> ₁₅ | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 | No, 0.8 |
| <i>F</i> ₁₆ | Approx, 0.8 | Gr, 0.6 | Gr, 0.8 | Gr, 0.8 | Auc, 0.6 | Gr, 0.6 |
| <i>F</i> ₁₇ | C, 0.6 | DC, 0.7 | C, 0.6 | C, 0.6 | DC, 0.7 | DC, 0.5 |
| <i>F</i> ₁₈ | <i>k</i> , 0.6 | None, 0.5 | None, 0.6 | None, 0.6 | Sngl, 0.8 | None, 0.7 |
| Taxonomy features | Mission scenarios | | | | | |
| | <i>MS</i> ₁₉ | <i>MS</i> ₂₀ | <i>MS</i> ₂₁ | <i>MS</i> ₂₂ | <i>MS</i> ₂₃ | <i>MS</i> ₂₄ |
| <i>F</i> ₃ | Ser, 0.9 | Ser, 0.9 | Ser, 0.9 | Ser, 0.9 | Res, 0.8 | Res, 0.8 |
| <i>F</i> ₅ | None, 0.7 | None, 0.7 | SNet, 0.7 | None, 0.7 | None, 0.7 | SNet, 0.8 |
| <i>F</i> ₆ | Yes, 0.7 | PReq, 0.7 | PReq, 0.7 | No, 0.5 | No, 0.5 | No, 0.6 |
| <i>F</i> ₇ | Yes, 0.7 | Yes, 0.8 | Yes, 0.8 | No, 0.7 | No, 0.7 | No, 0.7 |
| <i>F</i> ₈ | Ser, 0.9 | Ser, 0.9 | Ser, 0.9 | Ser, 0.9 | Res, 0.8 | Res, 0.8 |
| <i>F</i> ₉ | Yes, 0.8 | Yes, 0.8 | No, 0.7 | No, 0.7 | No, 0.7 | No, 0.7 |
| <i>F</i> ₁₁ | MT, 0.7 | MU, 0.7 | MT, 0.7 | MU, 0.7 | MT, 0.6 | MT, 0.6 |
| <i>F</i> ₁₂ | H, 0.8 | L, 0.6 | L, 0.7 | L, 0.7 | L, 0.7 | L, 0.9 |
| <i>F</i> ₁₃ | TE, 0.8 | TE, 0.5 | TE, 0.7 | IA, 0.7 | TE, 0.7 | TE, 0.7 |
| <i>F</i> ₁₄ | Yes, 0.8 | No, 0.7 | No, 0.7 | No, 0.7 | Yes, 0.7 | Yes, 0.7 |
| <i>F</i> ₁₅ | No, 0.8 | Yes, 0.9 | Yes, 0.9 | Yes, 0.9 | Yes, 0.9 | No, 0.8 |
| <i>F</i> ₁₆ | Gr, 0.6 | Auc, 0.7 | Auc, 0.6 | Approx, 0.6 | Gr, 0.7 | Gr, 0.7 |
| <i>F</i> ₁₇ | DC, 0.5 | DC, 0.5 | DC, 0.5 | DC, 0.5 | DC, 0.8 | DC, 0.8 |
| <i>F</i> ₁₈ | None, 0.7 | None, 0.5 | None, 0.5 | None, 0.5 | None, 0.5 | None, 0.6 |

Feature Domain Key

Approx:Approximation

Auc:Auction-based

C:Centralized

DC:Decentralized

Gr:Greedy

H:High Communication

IA:Instantaneous

k:Bounded Size

L:Low Communication

MC:Minimize Cost

MT:Maximize Tasks Completed

MU:Maximize Utility

Org:Organization Hierarchy

PReq:Prerequisite

Res:Resource-Model

Ser:Service-Model

SNet:Social Network

Sngl: Single Sized Coalitions

TE:Time-Extended

Table 4 continued**Taxonomy Feature Key**

| | |
|---|--|
| <i>F</i> ₃ : Agent Capability Model | <i>F</i> ₅ : Agent Structure |
| <i>F</i> ₆ : Inter-Task Constraints | <i>F</i> ₇ : Task Preemption |
| <i>F</i> ₈ : Task Requirement Model | <i>F</i> ₉ : Intra-Task Constraints |
| <i>F</i> ₁₁ : Performance Criterion | <i>F</i> ₁₂ : Communication Overhead |
| <i>F</i> ₁₃ : Task Allocation | <i>F</i> ₁₄ : Spatial Constraints |
| <i>F</i> ₁₅ : Overlapping Coalitions | <i>F</i> ₁₆ : Algorithm Technique |
| <i>F</i> ₁₇ : Algorithm Implementation | <i>F</i> ₁₈ : Coalition Size Constraint |

criteria were satisfied by Service and Adams' heuristic algorithm (A_9) that generates bounded robot coalitions, while maximizing the utility. Shehory and Kraus' heuristic algorithm (A_1) and Vig and Adams' algorithm (A_2) were also appropriate (Table 7). Both algorithms A_2 and A_9 extend A_1 , but A_9 maximizes utility, while A_2 minimizes system cost. The expected utility scores of A_1 and A_2 were almost identical because neither can satisfy the mission criterion of maximizing utility.

Mission Scenario 2 (MS_2) was the same as MS_1 except that MS_2 minimized system cost as the performance objective. *i-CiFHaR* selected Shehory and Kraus' heuristic algorithm (A_1) as the best algorithm and Vig and Adams' algorithm (A_2) as the second most suitable algorithm. Since both algorithms satisfied all the mission criteria and are associated with the the same feature-value pairs, except overlapping coalitions, they have similar expected utility scores. A_2 extends A_1 for real-robot domains and both seek to minimize cost. Service and Adams' algorithm (A_9) ranked third because it satisfied all mission requirements, but the minimize cost requirement.

Mission Scenario 3 (MS_3) required overlapping and bounded coalitions, while minimizing the overall system cost. Four algorithms were selected with Shehory and Kraus' algorithm (A_1) ranked as the most appropriate. A_1 is the only algorithm that allows overlapping and bounded coalitions, while also minimizing system cost. Vig and Adams' algorithm (A_2), the second best caters to almost all the mission criteria, but does not permit overlapping coalitions, despite being an extension of A_1 . Despite generating overlapping coalitions, Zhang et al.'s algorithm (A_{19}) was selected as the third choice, because it maximizes total utility. Service and Adams' algorithm (A_9) ranked fourth, because it does not permit overlapping coalitions and maximizes the total utility.

Mission Scenario 4 (MS_4) required overlapping coalitions and the maximization of the system utility, but did not restrict the coalition sizes. *i-CiFHaR* selected Zhang et al.'s algorithm (A_{19}) as the most suitable algorithm. The algorithm leverages particle swarm based optimization technique in order to generate overlapping coalitions, while maximizing system utility without incorporating the bounded coalition size heuristic. Service and Adams' algorithm (A_9) ranked second, because it restricts the maximum coalition size and does not permit overlapping coalitions; however, this algorithm satisfies the rest of the mission criteria.

The communication bandwidth availability was high for both MS_3 and MS_4 ; however, Mission Scenario 5 (MS_5) required a low communication footprint due to constrained bandwidth. Moreover, MS_5 required overlapping coalitions serves (e.g., box-pushing, foraging, sentry-duty) were used to represent the agents' capabilities and the tasks' requirements. Shiroma and Campos' CoMuTaR (A_{18}) is a service-based coalition formation algorithm that permits overlapping coalitions and requires low communication bandwidth, as a result it was ranked as the most appropriate algorithm. Zhang et al.'s algorithm (A_{19}) was selected as

Table 5 Clustering mission scenarios based on features

| Clustering of mission scenarios | | |
|--|----------------------------|--|
| Constant features | Impacted mission scenarios | Features altered |
| $F_3, F_5, F_6, F_7, F_8, F_9, F_{13}, F_{14}, F_{15}, F_{17}$ | MS_1 | $F_{11}, F_{12}, F_{16}, F_{18}$ |
| | MS_2 | |
| | MS_{17} | |
| $F_5, F_6, F_7, F_9, F_{13}, F_{14}, F_{15}, F_{17}$ | MS_3 | $F_3, F_8, F_{11}, F_{12}, F_{16}, F_{18}$ |
| | MS_4 | |
| | MS_5 | |
| $F_7, F_9, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}$ | MS_6 | $F_3, F_5, F_6, F_8, F_{11}, F_{13}, F_{18}$ |
| | MS_7 | |
| | MS_8 | |
| | MS_9 | |
| | MS_{10} | |
| $F_3, F_5, F_6, F_7, F_8, F_9, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}$ | MS_{11} | F_{13} |
| | MS_{12} | |
| $F_3, F_5, F_7, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{17}$ | MS_{13} | F_6, F_{18} |
| | MS_{14} | |
| | MS_{15} | |
| $F_5, F_7, F_{13}, F_{14}, F_{15}, F_{16}, F_{18}$ | MS_{16} | $F_3, F_6, F_8, F_9, F_{11}, F_{12}, F_{17}$ |
| | MS_{17} | |
| | MS_{18} | |
| | MS_{19} | |
| $F_3, F_8, F_{13}, F_{17}, F_{18}$ | MS_{20} | $F_5, F_6, F_7, F_9, F_{11}, F_{12}, F_{14}, F_{15}, F_{16}$ |
| | MS_{21} | |
| | MS_{22} | |
| | MS_{23} | |
| | MS_{24} | |
| $F_3, F_6, F_7, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{16}, F_{17}, F_{18}$ | MS_{23} | F_5, F_{15} |
| | MS_{24} | |

Taxonomy Feature Key F_3 : Agent Capability Model F_6 : Inter-Task Constraints F_8 : Task Requirement Model F_{11} : Performance Criterion F_{13} : Task Allocation F_{15} : Overlapping Coalitions F_{17} : Algorithm Implementation F_5 : Agent Structure F_7 : Task Preemption F_9 : Intra-Task Constraints F_{12} : Communication Overhead F_{14} : Spatial Constraints F_{16} : Algorithm Technique F_{18} : Coalition Size Constraint

Table 6 Coalition formation algorithm selections for mission scenarios

| | Algorithms | Mission scenarios |
|-----------------|---|--|
| A ₁ | Shehory and Kraus [4] | $MS_1, MS_2, MS_3, MS_{23}$ |
| A ₂ | Vig and Adams [6] | $MS_1, MS_2, MS_3, MS_{23}$ |
| A ₃ | Vig and Adams [12] | $MS_{10}, MS_{11}, MS_{18}, MS_{19}, MS_{20}, MS_{21}$ |
| A ₄ | Abdallah and Lesser [3] | $MS_6, MS_7, MS_9, MS_{24}$ |
| A ₅ | Tošić and Agha [5] | $MS_6, MS_7, MS_9, MS_{24}$ |
| A ₆ | Weerd et al. [17] | $MS_6, MS_7, MS_8, MS_9, MS_{24}$ |
| A ₇ | Campbell et al. [20] | MS_{14} |
| A ₈ | Sujit et al. [19] | $MS_{14}, MS_{16}, MS_{23}, MS_{24}$ |
| A ₉ | Service and Adams [2]-Resource Model | $MS_1, MS_2, MS_3, MS_4, MS_6, MS_9, MS_{17}, MS_{23}$ |
| A ₁₀ | Gerkey and Mataric [10] | MS_{17} |
| A ₁₁ | Service and Adams [2]-Service Model | $MS_5, MS_{13}, MS_{18}, MS_{19}, MS_{20}, MS_{21}, MS_{22}$ |
| A ₁₂ | Service and Adams [23]-Approximation | $MS_{12}, MS_{13}, MS_{22}$ |
| A ₁₃ | Service and Adams [2]-Dynamic Programming Agent Types | $MS_{12}, MS_{13}, MS_{22}$ |
| A ₁₄ | Service et al. [24]-Simultaneous Descending | $MS_{10}, MS_{11}, MS_{18}, MS_{19}, MS_{20}, MS_{21}$ |
| A ₁₅ | Gaston and desJardins [18] | MS_8, MS_{21} |
| A ₁₆ | Koes et al. [21] | $MS_{15}, MS_{16}, MS_{18}, MS_{19}$ |
| A ₁₇ | Ramchurn et al. [22] | $MS_{15}, MS_{16}, MS_{19}$ |
| A ₁₈ | Shiroma and Campos [11] | $MS_5, MS_{20}, MS_{21}, MS_{22}$ |
| A ₁₉ | Zhang et al. [16] | $MS_3, MS_4, MS_5, MS_{23}$ |

the second most appropriate alternative, but it does not satisfy two mission criteria, namely the low communication and the service model-based requirements. Service and Adams' algorithm (A₁₁) ranked third, because it does not permit overlapping coalitions, generates bounded coalitions, and requires a high communication bandwidth for its operation.

Consider the first response example from Sect. 1, where coalitions of robots assess the situation. Such real-world environments often require low communications. Mission Scenario 6 (MS_6) depicts such a situation, where the communication bandwidth is restricted due to environmental constraints and small-sized coalitions, bounded within a maximum limit of k are preferred. Moreover, the critical situation demanded instantaneous task allocations with robots forming a social network topology based on the limited inter-robot communication, resulting in a sparse network. Given the mission criteria, *i-CiFHaR* ranked Tošić and Agha's algorithm (A₅) as the most appropriate, because it leverages a social network and requires low communication bandwidth, when the underlying topology graph is sparse. Weerd et al.'s algorithm (A₆) was ranked second, because it too leverages a team of social networked robots to compute task coalitions under constrained communication requirements. Service and Adams' algorithm (A₉) ranked third, because it computes bounded coalitions, and requires a high communication bandwidth for its operation. Abdallah and Lesser's heuristic algorithm (A₄) satisfies almost all the mission criteria and it was ranked fourth, because it considers robots to be part of an organization hierarchy, rather than a social network.

Mission Scenario 7 (MS_7) was very similar to MS_6 , except that MS_7 required time-extended allocations that stem from the need of task scheduling. Additionally, there was no restrictions on the coalition sizes. Given the new conditions, *i-CiFHaR* ranked Weerd et al.'s

Table 7 Ranking of coalition formation algorithms by decreasing expected utility scores for each mission scenario. Each mission describes the most appropriate subset of algorithms in the format, *Algorithm (Expected Utility Score)*

| Mission scenarios | Rankings of algorithms | | | | |
|-------------------------|---------------------------------|---------------------------------|----------------------------------|---------------------------------|---------------------------------|
| | 1st | 2nd | 3rd | 4th | 5th |
| <i>MS</i> ₁ | <i>A</i> ₉ (9467.4) | <i>A</i> ₁ (8914.5) | <i>A</i> ₂ (8867.5) | – | – |
| <i>MS</i> ₂ | <i>A</i> ₁ (9297.1) | <i>A</i> ₂ (9248.4) | <i>A</i> ₉ (9046.5) | – | – |
| <i>MS</i> ₃ | <i>A</i> ₁ (8628.9) | <i>A</i> ₂ (8373.3) | <i>A</i> ₁₉ (8193.9) | <i>A</i> ₉ (8173.8) | – |
| <i>MS</i> ₄ | <i>A</i> ₁₉ (9321.2) | <i>A</i> ₉ (8421.3) | – | – | – |
| <i>MS</i> ₅ | <i>A</i> ₁₈ (7019.5) | <i>A</i> ₁₉ (7013.9) | <i>A</i> ₁₁ (6358.61) | – | – |
| <i>MS</i> ₆ | <i>A</i> ₅ (7611.2) | <i>A</i> ₆ (7122.5) | <i>A</i> ₉ (7109.6) | <i>A</i> ₄ (6937.9) | – |
| <i>MS</i> ₇ | <i>A</i> ₆ (7600.5) | <i>A</i> ₅ (7404.9) | <i>A</i> ₄ (6987.6) | – | – |
| <i>MS</i> ₈ | <i>A</i> ₁₅ (6379.5) | <i>A</i> ₆ (5845.6) | – | – | – |
| <i>MS</i> ₉ | <i>A</i> ₄ (7583.8) | <i>A</i> ₆ (7124.1) | <i>A</i> ₉ (7094.7) | <i>A</i> ₅ (6835.6) | – |
| <i>MS</i> ₁₀ | <i>A</i> ₃ (8062.9) | <i>A</i> ₁₄ (7326.3) | – | – | – |
| <i>MS</i> ₁₁ | <i>A</i> ₁₄ (7667.7) | <i>A</i> ₃ (7598.2) | – | – | – |
| <i>MS</i> ₁₂ | <i>A</i> ₁₂ (7528.4) | <i>A</i> ₁₃ (7528.4) | – | – | – |
| <i>MS</i> ₁₃ | <i>A</i> ₁₁ (7583.9) | <i>A</i> ₁₂ (7008.3) | <i>A</i> ₁₃ (7008.3) | – | – |
| <i>MS</i> ₁₄ | <i>A</i> ₈ (5970.1) | <i>A</i> ₇ (5469.9) | – | – | – |
| <i>MS</i> ₁₅ | <i>A</i> ₁₆ (5009.7) | <i>A</i> ₁₇ (4716.5) | – | – | – |
| <i>MS</i> ₁₆ | <i>A</i> ₁₇ (4892.8) | <i>A</i> ₁₆ (4817.2) | <i>A</i> ₈ (4536.7) | – | – |
| <i>MS</i> ₁₇ | <i>A</i> ₁₀ (7662.2) | <i>A</i> ₉ (7492.2) | – | – | – |
| <i>MS</i> ₁₈ | <i>A</i> ₁₆ (4948.6) | <i>A</i> ₁₄ (4709.4) | <i>A</i> ₁₁ (4683.5) | <i>A</i> ₃ (4634.7) | – |
| <i>MS</i> ₁₉ | <i>A</i> ₁₇ (4472.1) | <i>A</i> ₁₆ (4396.7) | <i>A</i> ₁₄ (4158.8) | <i>A</i> ₁₁ (4125.2) | <i>A</i> ₃ (4083.33) |
| <i>MS</i> ₂₀ | <i>A</i> ₁₈ (4727.8) | <i>A</i> ₃ (4689.8) | <i>A</i> ₁₄ (4342.6) | <i>A</i> ₁₁ (4328.8) | – |
| <i>MS</i> ₂₁ | <i>A</i> ₁₅ (4002.3) | <i>A</i> ₁₈ (3819.2) | <i>A</i> ₃ (3660) | <i>A</i> ₁₄ (3649) | <i>A</i> ₁₁ (3635.7) |
| <i>MS</i> ₂₂ | <i>A</i> ₁₈ (5472.8) | <i>A</i> ₁₁ (5219) | <i>A</i> ₁₂ (5150.5) | <i>A</i> ₁₃ (5150.5) | – |
| <i>MS</i> ₂₃ | <i>A</i> ₈ (5547.2) | <i>A</i> ₁₉ (5341.7) | <i>A</i> ₁ (5334.9) | <i>A</i> ₂ (5076.1) | <i>A</i> ₉ (5037.3) |
| <i>MS</i> ₂₄ | <i>A</i> ₆ (6072.1) | <i>A</i> ₈ (5690.5) | <i>A</i> ₅ (5687.8) | <i>A</i> ₄ (5524.0) | – |

algorithm (*A*₆) as the most appropriate algorithm, because it provides time-extended task allocations with a low communication overhead and leverages a social network. Tošić and Agha's algorithm (*A*₅) was ranked second, because it did not meet the time-extended mission requirement, while it computes only bounded coalitions. Abdallah and Lesser's algorithm (*A*₄) ranked third, because it permits time-extended allocations with limited communication bandwidth requirements and unbounded coalitions.

Mission Scenario 8 (*MS*₈) differed from *MS*₆ and *MS*₇, but the robots' communication structure remained the same (Social network). This mission was different in that it required a service-based model and the objective was to maximize the number of tasks to be completed within a stipulated time frame, unlike the utility maximization objective of *MS*₆ and *MS*₇. Time-extended allocation was also a criterion. Gaston and desJardins' coalition formation algorithm (*A*₁₅) was selected as the most appropriate algorithm that satisfied all the mission requirements. Weerd et al.'s algorithm (*A*₆) was ranked second, because it satisfies most of the mission requirements, but fails in that it maximizes the total utility and leverages a resource-based model.

Mission Scenario 9 (MS_9) was similar to MS_6 , but simulated a scenario where robots are connected in an organizational hierarchy. Low communication was a mission constraint and there was no bound on the coalition size. Time-extended allocations were necessary. *i-CiFHaR* selected Abdallah and Lesser's algorithm (A_4) as the most appropriate algorithm, because this is the only algorithm that utilizes an organizational hierarchy to compute time-extended coalitions with no size restrictions. Weerd et al.'s algorithm (A_6) ranked second. Service and Adams' algorithm (A_9 -Resource Model) ranked third, while Tošić and Agha's algorithm (A_5) was selected as the fourth alternative.

Mission scenario 10 (MS_{10}) included a number of high priority tasks requiring frequent task preemption. MS_{10} required a *service model* and an auction-based coalition formation algorithm. Vig and Adams' RACHNA (A_3) ranked as the most appropriate algorithm, because it satisfied all mission requirements, including task preemption. Service et al.' simultaneous descending auction-based algorithm (A_{14}), an extension of RACHNA, was ranked second.

Mission Scenario 11 (MS_{11}) required task preemption and a service-model similar to MS_{10} ; however, MS_{11} required time-extended allocations. *i-CiFHaR* selected the simultaneous descending auction algorithm (A_{14}) as the most appropriate algorithm, since it generates both instantaneous and time-extended coalitions, while permitting online task preemption. Vig and Adams' RACHNA (A_3) ranked second, because it allows only instantaneous allocations.

Mission Scenario 12 (MS_{12}) simulated critical tasks requiring high utility coalitions with guaranteed solution quality, thereby requiring approximation algorithms. Agent capabilities and task resource requirements were expressed in terms of services (e.g., patrolling, sentry-duty, foraging), while robot coalition sizes were not constrained. *i-CiFHaR* selected the two approximation algorithms, A_{12} and A_{13} with equal expected utility scores. The two algorithms are equally applicable to MS_{12} , since they are associated with the same feature-value pairs.

Mission Scenario 13 (MS_{13}) differed from MS_{12} in that coalition sizes were restricted by an upper bound, k and solution quality was not critical; therefore, there was an equal likelihood that either a greedy or an approximation algorithm is applicable. The remaining mission requirements were identical to MS_{12} . Service and Adams' service-model based heuristic algorithm (A_{11}) was selected the most appropriate algorithm, since it met all mission requirements. *i-CiFHaR* selected the two approximation algorithms, A_{12} and A_{13} as the remaining alternatives.

Real robots must travel to the assigned task's location; therefore, satisfying spatial constraints in real-world conditions is essential. Mission Scenario 14 (MS_{14}) required that robot coalitions met spatial constraints and had a low communication footprint. Time-extended task allocations were necessary and the performance objective was to maximize the number of tasks completed in a given time. *i-CiFHaR* selected Sujit et al.'s algorithm (A_8) as the most suitable fit, because this algorithm satisfies spatial constraints using Dubin's curves to estimate the travel time to task locations. Moreover, A_8 uses low inter-robot communication bandwidth ($O(n \times m)$) with n robots and m tasks, and maximizes the number of tasks completed. Campbell et al.'s heuristic algorithm (A_7) ranked second, since it creates single robot coalitions with no inter-agent communication at all, and models the coalition formation problem as a multi-processor scheduling problem to maximize the number of completed tasks.

Mission Scenario 15 (MS_{15}) simulated tasks with hard task completion deadlines. Additionally, MS_{15} involved tasks with dependencies, invoking the need to satisfy inter-task precedence constraints. Consider two mission tasks: the first is to triage a victim and the second is to clear debris covering an injured victim. The second task must be performed first. The performance objective of MS_{15} was to maximize system utility, while time-extended allo-

cations were preferred. *i-CiFHaR* selected Koes et al.'s algorithm (A_{16}) as the best, because it met all mission requirements. Ramchurn et al.'s algorithm (A_{17}) was second, because it maximizes the number of completed tasks.

Mission Scenario 16 (MS_{16}) differed from MS_{15} in that it the performance objective was to maximize the number of completed tasks, while the remaining mission requirements were kept the same. *i-CiFHaR* selected Ramchurn et al.'s algorithm (A_{17}) as the most appropriate algorithm, as it satisfied all mission criteria. This time Koes et al.'s algorithm (A_{16}) was selected as the second most suitable alternative. Sujit et al.'s algorithm (A_8) was chosen as the third alternative.

Mission scenario 17 (MS_{17}) depicted a situation where single robot coalitions were necessary under low communication requirements, and auction-based algorithms were preferred. Gerkey and Mataric's MURDOCH (A_{10}) ranked as the best fit, followed by Service and Adams' resource-model based algorithm (A_9). The latter was ranked lower, because it requires high inter-agent communication messages for computing coalitions.

Mission situation MS_{18} was a conglomeration of several critical requirements, including task preemption and addressing of inter- and intra-task constraints along with spatial constraints. Additionally, the mission required scheduling, thus time-extended allocation was crucial for the team of service-based robots. *i-CiFHaR* considered all the critical criteria and selected Koes et al.'s centralized coalition formation algorithm (A_{16}), because this algorithm addresses both inter- and intra-task constraints, alongside spatial constraints. Moreover, Koes et al.'s algorithm permits time-extended allocations, but fails to allow task preemption. Service et al.'s simultaneous descending auction approach (A_{14}) was ranked second, on the grounds that this algorithm allows task preemption and time-extended allocations; however, fails to address the task and spatial constraints. Service and Adams' service-model algorithm (A_{11}) was the third alternative, while RACHNA (A_3) being a task preemptive approach was ranked fourth.

The mission requirements of the next scenario (MS_{19}) were almost identical to that of MS_{18} , except that the former sought to maximize the number of completed tasks that were spatially distributed. *i-CiFHaR* re-evaluated the requirements and recommended Ramchurn et al.'s coalition formation algorithm (A_{17}), because it satisfies most of the mission criteria, except task preemption. Koes et al.'s algorithm (A_{16}) was ranked second, because it aims to maximize the utility and does not permit preemption; however, it does satisfy the remaining criteria. Service et al.'s approach (A_{14}) was ranked higher than RACHNA (A_3), because the former allows time-extended allocations along with task preemption, whereas the latter permits instantaneous allocations and task preemption. Service and Adams' service-model algorithm (A_{11}) was the third alternative.

Mission Scenario 20 (MS_{20}) demanded two additional criteria. Aiming to reduce resource usage, the mission preferred overlapping coalitions. Low communication bandwidth was required alongside the aforementioned criteria of MS_{18} and MS_{19} . CoMutaR (A_{18}) was ranked first, owing to the fact that this service-model based algorithm computes overlapping coalitions with low communication bandwidth requirements. RACHNA (A_3) and Service et al.'s algorithm (A_{14}) were chosen as the next suitable alternatives, because these algorithms allow task preemption and seek to maximize the utility. Service and Adams' service-model algorithm (A_{11}) was ranked fourth.

The robots in mission scenario MS_{21} were connected in a communication social network and task preemption was necessary. The objective was to maximize the number of completed tasks and reduce the communication footprint. Time-extended allocations were also crucial. *i-CiFHaR* selected Gaston and desJardins' service-model algorithm (A_{15}) that employs a social network as a heuristic to perform the time-extended task allocations with reduced

communication overhead. CoMutaR (A_{18}) ranked second and falls short of maximizing the completed task objective criterion, while catering to most of the mission requirements. RACHNA (A_3), along with its improved and extended successor (A_{14}) ranked third and fourth respectively, given the fact that both the algorithms permit task preemption. Service and Adams' service-model algorithm (A_{11}) was the remaining alternative.

Mission scenario MS_{22} aimed for approximate overlapping coalitions along with low communication overhead and the objective to maximize team utility. *i-CiFHaR* selected CoMutaR (A_{18}), because this algorithm allows overlapping coalitions with constraints on communication bandwidth and maximizes total team utility. However, the approach is an auction-based technique and fails to provide guarantees on the solution quality. *i-CiFHaR* recommended Service and Adams' two approximation algorithms, A_{12} and A_{13} as the next two highest ranked algorithms, because they provide solutions within a fixed bound from the optimal solutions. However, the approximation algorithms fail to allow overlapping coalitions with a low communication footprint.

Mission scenario MS_{23} was designed for robots that follow the *Resource-Model*. The mission requirements involved overlapping coalitions with time-extended allocations for spatially distributed tasks. *i-CiFHaR* selected Sujit et al.'s coalition formation algorithm (A_8) that uses Dubin's curves to address spatially distributed tasks and computes time-extended allocations with low communication messages. However, this algorithm does not permit overlapping coalitions. Zhang et al.'s particle-swarm based approach (A_{19}) and Shehory and Kraus' algorithm (A_1) were selected as the second and third choices respectively, since both algorithms allow overlapping coalitions, but fail to provide time-extended allocations. Vig and Adams' algorithm (A_2), which is an extension of A_1 was ranked fourth, while Service and Adams' algorithm (A_9) was the last alternative.

The last mission, MS_{24} was similar to MS_{23} , except that the robots used a communication topology as a social network. *i-CiFHaR* selected Weerd et al.'s algorithm (A_6) as the most appropriate algorithm, based on the fact that this algorithm offers time-extended allocations with low communication footprints and leverages a social network to compute coalitions. Sujit et al.'s approach (A_8) was ranked second, because it seeks to maximize the number of completed tasks and provides time-extended allocations with low communication overhead, few of the mission criteria. A_5 and A_4 were the remaining alternatives.

The results show that for all the simulated mission scenarios, *i-CiFHaR* successfully selected the most appropriate algorithms to apply based on multiple mission criteria. When a single best fit algorithm is not found, *i-CiFHaR* determines a subset of algorithms that are most suitable. The highlighted missions represent a subset of all possible scenarios; however, the number of all possible missions is exorbitantly large. The selected mission scenarios exploited all taxonomy features and provided a good subset of the possible missions.

5 Conclusion

An intelligent framework is presented that reasons online over a library of coalition formation algorithms to select the most appropriate coalition formation algorithm(s) to apply to a given mission scenario. *i-CiFHaR* is the first coalition framework to use a library of algorithms, rather than a single heuristic based algorithm. The framework leverages an influence diagram to make decisions online under multiple, uncertain mission criteria. A link analysis based algorithm calculates the utility values of the feature-value pairs. The framework uses a number of features to select the most suitable coalition formation algorithm(s). The curse of dimensionality is addressed by extracting prominent features that discriminate the coali-

tion algorithms. These prominent features are utilized to dynamically create the influence diagram at run-time. The experimental results show that *i-CiFHaR* selects the appropriate algorithm(s), given multiple mission criteria. When a single best fit algorithm is unavailable, *i-CiFHaR* selects a subset of suitable algorithms. *i-CiFHaR* is applicable to missions with frequent contingency occurrences that introduce changing mission requirements (e.g., overlapping coalitions resulting from robot failures, task preemption). The likelihood of handling diverse situations increases with the inclusion of a broad set of algorithms. *i-CiFHaR* provides a more robust approach to allocate task coalitions for dynamic, real-world scenarios.

Acknowledgments This research has been supported by an ONR DEPSCOR Award # N000140911161.

References

- Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2), 209–238.
- Service, T.C., Adams, J.A. (2011). Coalition formation for task allocation: Theory and algorithms. *Journal of Autonomous Agents and Multi-agent Systems*, 22(2), 225–248.
- Abdallah, S., & Lesser, V. (2004). Organization-based cooperative coalition formation. In: *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 162–168.
- Shehory, O., & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101, 165–200.
- Tošić, P. T., & 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., & Adams, J. A. (2006b). Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4), 637–649.
- Dang, V. D., & Jennings, N. (2004). Generating coalition structures with finite bound from the optimal guarantees. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems*, pp. 564–571.
- Rahwan, T., Ramchurn, S., Jennings, N., & Giovannucci, A. (2009). An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research*, 34, 521–567.
- Rothkopf, M. H., Pekeč, A., & Harstad, R. M. (1998). Computationally manageable combinatorial auctions. *Management Science*, 44(8), 1131–1147.
- Gerkey, B., & Matarić, M. J. (2002). Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758–768.
- Shiroma, P. M., & 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*, pp. 4817–4824.
- Vig, L., & Adams, J. A. (2006a) Market-based multi-robot coalition formation. In: *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems*, pp. 227–236.
- DeJong, P. (2005). Coalition formation in multi-agent UAV systems. MS Thesis, University of Central Florida.
- Service, T. C., & Adams, J. A. (2010). Coalition formation algorithm taxonomy. Tech. Rep. HMT-10-03, Vanderbilt University.
- Kjærulff, U. B., & Madsen, A. L. (2008). *Bayesian networks and influence diagrams: A guide to construction and analysis*. *Information Science and Statistics*. New York, NY: Springer.
- Zhang, G., Jiang, J., Su, Z., Qi, M., & Fang, H. (2010). Searching for overlapping coalitions in multiple virtual organizations. *Information Sciences*, 180(17), 3140–3156.
- Weerd, M., Zhang, Y., & Klos, T. (2007). Distributed task allocation in social networks. In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 17–24.
- Gaston, M. E., & desJardins, M. (2005). Agent organized networks for dynamic team formation. In: *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 230–237.
- Sujit, P. B., George, J. M., & Beard, R. W. (2008). Multiple UAV coalition formation. In: *Proceedings of the American Control Conference*, pp. 2010–2015.
- Campbell, A., Wu, A. S., & Shumaker, R. (2008). Multi-agent task allocation: Learning when to say no. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 201–208.

21. Koes, M., Nourbakhsh, I., & Sycara, K. (2005). Heterogeneous multirobot coordination with spatial and temporal constraints. In: *Proceedings of the 20th National Conference on Artificial Intelligence*, pp. 1292–1297.
22. Ramchurn, S. D., Polukarov, M., Farinelli, A., Truong, C., & Jennings, N. R. (2010). Coalition formation with spatial and temporal constraints. In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1181–1188.
23. Service, T.C., Adams, J.A. (2011). Constant factor approximation algorithms for coalition structure generation. *Journal of Autonomous Agents and Multi-Agent Systems*, 23(1), 1–17.
24. Service, T. C., Sen, S. D., & 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*.
25. Cao, Y. U., Fukunaga, A. S., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1), 7–27.
26. Dudek, G., Jenkin, M., & Milius, E. (2002). A taxonomy of multirobot systems. In T. Balch & L. E. Parker (Eds.), *Robot teams: From diversity to polymorphism* (pp. 3–22). Natick, MA: AK Peters Ltd.
27. Farinelli, A., Iocchi, L., & Nardi, D. (2004). Multirobot systems: A classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(5), 2015–2028.
28. Gerkey, B. P., & Mataric, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9), 939–954.
29. Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5), 604–632.
30. Page, L., Sergey, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Tech. Rep. 1999–66, Stanford InfoLab, Stanford University.
31. Jolliffe, I. (1972). Discarding variables in a principal component analysis. I: Artificial data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 21(2), 160–173.
32. Song, F., Guo, Z., & Mei, D. (2010). Feature selection using principal component analysis. In: *Proceedings of the International Conference on System Science, Engineering Design and Manufacturing Informatization*, pp. 27–30.
33. Duntelman, G. H. (1989). *Principal components analysis*, vol. 69. Newbury Park, CA: Sage Publications.
34. Yu, Q., & Terzopoulos, D. (2007). A decision network framework for the behavioral animation of virtual humans. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 119–128.
35. Sen, S. D., & Adams, J. A. (2013). A decision network based framework for multiagent coalition formation. In: *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems*, pp. 55–62.
36. Cohen, M. D. (1984). Coping with complexity: The adaptive value of changing utility. *The American Economic Review*, 74(1), 30–42.
37. Cyert, R. M., DeGroot, M. H. (1979) Adaptive utility. In: *Expected Utility Hypotheses and the Allais Paradox*, pp. 223–241. Netherlands: Springer
38. Nielsen, T. D., & Jensen, F. V. (2004). Learning a decision maker's utility function from (possibly) inconsistent behavior. *Artificial Intelligence*, 160(1), 53–78.
39. Grüne-Yanoff, T., & Hansson, S. O. (2009). Preference change: An introduction. In: *Preference Change*, pp. 1–26. Netherlands: Springer
40. Neumann, L. J., & Morgenstern, O. (1947). *Theory of games and economic behavior*. Princeton, NJ: Princeton University Press.
41. Nokia Qt. (2012). <http://www.qt.io/>. Accessed 7 Jan 2012.
42. NORSYS: Netica application: A complete software package to solve problems using bayesian belief networks and influence diagrams. (2012). <http://www.norsys.com/netica.html>. Accessed 12 June 2012