I.J. Information Technology and Computer Science, **2022, 1, 9-19**
Published Online February 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2022.01.02

Decentralized Self-adaptation in the Presence of Partial Knowledge with Reduced Coordination Overhead

Kishan Kumar Ganguly
Institute of Information Technology, University of Dhaka
E-mail: kkganguly@iit.du.ac.bd

Moumita Asad
Institute of Information Technology, University of Dhaka
E-mail: bsse0731@iit.du.ac.bd

Kazi Sakib
Institute of Information Technology, University of Dhaka
E-mail: sakib@iit.du.ac.bd

Received: 16 September 2021; Accepted: 21 Novermber 2021; Published: 08 February 2022

Abstract: Decentralized self-adaptive systems consist of multiple control loops that adapt some local and system-level global goals of each locally managed system or component in a decentralized setting. As each component works together in a decentralized environment, a control loop cannot take adaptation decisions independently. Therefore, all the control loops need to exchange their adaptation decisions to infer a global knowledge about the system. Decentralized self-adaptation approaches in the literature uses the global knowledge to take decisions that optimize both local and global goals. However, coordinating in such an unbounded manner impairs scalability. This paper proposes a decentralized self-adaptation technique using reinforcement learning that incorporates partial knowledge in order to reduce coordination overhead. The Q-learning algorithm based on Interaction Driven Markov Games is utilized to take adaptation decisions as it enables coordination only when it is beneficial. Rather than using unbounded number of peers, the adaptation control loop coordinates with a single peer control loop. The proposed approach was evaluated on a service-based Tele Assistance System. It was compared to random, independent and multiagent learners that assume global knowledge. It was observed that, in all cases, the proposed approach conformed to both local and global goals while maintaining comparatively lower coordination overhead.

Index Terms: Decentralized Self-Adaptation, Partial Knowledge, Coordination Overhead, Q-learning.

1. Introduction

In this era of advanced computing, increased connectedness among systems has made decentralization an expected requirement. Currently, decentralization is also being considered in self-adaptive systems for better scalability and fault tolerance. Decentralized self-adaptive systems are formed by adding a control loop to each component of the decentralized system which is termed as locally managed system in this paper. Each control loop manages some local goals related to the locally managed system. All the control loops altogether also manage some global goals which emerge from the local behaviors of the systems. For example, a control loop may need to maintain a local goal of maximum 2 seconds response time for its locally managed system. Moreover, all the control loops may require maintaining a maximum total response time of 4 seconds when the locally managed systems work together, which is a global goal. The control loops may require coordination among themselves for goal conformance. Satisfying these local and global goals continuously is the primary concern for a decentralized self-adaptive system.

As global behavior emerges from local behavior for decentralized self-adaptive systems, multiple control loops cannot decide independently of one another. In the worst case, all the control loops may be interdependent on one another. As a result, each needs to observe the decisions made by the other ones. This is not feasible in practice because providing each control loop with a global view harms scalability [1]. Hence, scalable solutions demand making decisions using a partial view of the environment which is challenging. Moreover, literature shows that multiple dependent locally managed systems do not depend on one another in every state [2]. That is, some states require

Copyright © 2022 MECS

coordination and some states do not. In the states where coordination is not needed, the control loops can decide individually which largely reduces the coordination overhead. The challenge is to learn where coordination is required for goal conformance.

Weyns et al. proposed a reference model for decentralized self-adaptive systems where they captured coordination requirement using a coordination model [1]. They mentioned partial knowledge incorporation and coordination overhead reduction as an important research challenge [1]. Grassi et al. proposed a gossip-based approach for self-adaptive service assembly where each service transferred its details to peer services using gossip [3]. A peer service resolved its dependencies by using the received services with maximum total utility. However, utility function calculation required considering all the peers which opposes the afore-mentioned partial knowledge assumption. Wang et al. proposed a Q-learning based technique for decentralized self-adaptive systems where adaptation decisions were taken assuming a global view of the system which may hamper the scalability of the system [4]. Besides, none of the approaches in the literature consider reducing coordination overhead, specially using the concept presented in the previous paragraph.

The objective of this paper is to propose an approach that reduces the coordination overhead of the decentralized self-adaptative systems by assuming a more practical setting of partial knowledge. This paper makes the following contribution.

1. A single peer instead of multiple peers is used to take adaptation decisions. To do these, Q-learning under the Interaction-Driven Markov Game (IDMG) model is utilized rather than using Q-learning based on the traditional Markov game model [2]. However, the IDMG-based Q-learning in the literature uses multiple peers. In this paper, IDMG-based Q-learning is applied in a different setting namely decentralized self-adaptation and the paper empirically shows that the IDMG-based Q-learning can take effective adaptation decisions for decentralized self-adaptive systems even if single peer is used.
2. An experimental study on a service-oriented system named Tele Assistance System (TAS) where the local and global goals have been mathematically defined and states where coordination are necessary have been marked by monitoring global goal violation [7].

Local and global goal conformance are calculated using reward functions (See Definition 3 in Section 4). Local and global reward functions are defined which calculate local and global goal conformance respectively. The states where global goals are always satisfied but local goals may be violated are considered as non-coordinating states. This is because global goals are directly related to coordination and so, their continuous satisfaction indicates that coordination may not be required. Here, individual Q-learners are used for self-adaptation [5]. In states where global goals may not be satisfied, coordination is considered under IDMG-Based Q-learning [6]. In this case, each local control loop coordinates with a single peer rather than multiple peers. In this paper, we hypothesize that considering only a single peer while performing such type of learning does not significantly reduce goal conformance [6]. Thus, the proposed approach addresses both partial knowledge and coordination overhead.

The approach is validated on a service-based TAS [7]. A local control loop was added to each service to self-adapt some local and global goals. The proposed approach was compared to other two approaches - one that chooses adaptation actions randomly and the other that uses individual Q-learners only. The proposed approach resulted in higher goal conformance than these two. It was also compared to a joint Q-learning (Q-learning where all the locally managed systems coordinate) based approach. In this case, although the joint Q-learner should have performed better, the opposite situation was observed. This is because joint Q-learning resulted in high coordination overhead which delayed the adaptation. As a result, the adaptation actions were taken based on outdated learning model and so, joint Q-learner performed worse than the proposed technique.

The rest of the paper is structured as follows**.** In Section 2, the related works are given. Section 3 presents a motivational example of Tele Assistance System. In Section 4, the problem addressed in this paper is formulated along with some necessary definitions. Section 5 contains the proposed approach. Section 6 and 7 contain the experiments and the results respectively. Section 8 holds the conclusion and the future works.

2. Related Work

Although some decentralized self-adaptation mechanisms are present in the literature, none of those solve both of the problems addressed in this paper. Cheng et al. listed decentralization of self-adaptive systems as a research direction in [8]. Weyns et al. studied the challenges of decentralized self-adaptive systems and proposed a reference model [1]. In this model, each iteration performed some computations regarding the satisfaction of local goals and some coordination for the satisfaction of the global goals. The computation and coordination mechanism used four meta-level models - *system*, *concern*, *working* and *coordination model* [1]. The first three models are related to the computation, whereas the last one is associated with coordination. This model showed high interest in building decentralized self-adaptive systems satisfying local and global goals. Furthermore, the author addressed the problem of partial knowledge and coordination overhead as challenging [1]. Weyns et al. proposed five patterns for control loop interaction which are

Hierarchical Control, *Master/Slave*, *Regional Planner*, *Fully Decentralized* and *Information Sharing*. Among these, only the last two are fully decentralized [9]. However, the patterns cannot directly answer the mechanism of satisfying local and global goals in a fully decentralized knowledge setting.

Georgiadis et al. proposed an architecture-based self-organization technique where each component maintained an architectural model of the whole system [10]. When a component was added or removed from the system, self-organization took place where each component tried to resolve its dependency by binding with another component. To do so, the architectural model was needed to be updated using total ordered broadcast. This is analogous to joint state and action-based decision making. Sykes et al. proposed the gossip-based Flashmob approach to self-organization solving the problem of [11]. In this approach, each node transferred states to other nodes using the gossip protocol where a state is the global configuration that contained dependencies of all the components. For resolving these dependencies, each node selected the variant with the maximum local utility. However, the concepts of local and global goals discussed in this paper were not presented in Flashmob. Grassi et al. proposed a technique for self-adaptive service assembly by considering local and global goals [3]. Similar to the proposed technique in this paper, local and global reward function values were calculated and variant selection was based on the maximum weighted summation of reward. Nevertheless, calculating all these required considering the reward function values of all the dependent services. Thus, their approach did not support partial knowledge incorporation.

Some Reinforcement learning-based approaches also exist in the literature. Dowling et al. proposed a technique that broke the global optimization problem into multiple similar discrete optimization problems [12]. Next, each node either solved it by itself or delegated to its neighbour that can solve it more efficiently. Although minimization and maximization problems can be broken down in this way, threshold-based ones are much harder to break down [13]. For this reason, the current paper takes a different approach by breaking the global minimization and maximization problem into local ones and optimizing the threshold-based global goal jointly with peers. Wang et al. introduced the WSC-TMG Model based on multiagent Q-learning which is similar to the technique presented in Section 5.1 [4]. Therefore, it suffers from both the problems mentioned previously. Caporuscio et al. proposed an approach for self-adaptive service composition [14]. They used gossip protocol to transmit service configurations and a primary list of dependency resolving services with maximum utility, which is similar to [3]. A second-level scanning was performed using learned reward values from Q-learning which provided the final dependency resolving services. However, the authors did not explicitly mention whether individual or joint action learners were used. The technique to calculate global reward functions required dissemination of each local reward function values to each of the agents.

3. Running Example

In this paper, a TAS that provides medical service to people will be used as a running example to demonstrate the methodology. In TAS, several vital parameters of the patients are measured. A Medical Analysis Service analyses these and invokes either an Alarm Service or Drug Service. Each of these services is supplied by individual service providers. However, multiple variants of these services can be provided. Medical Analysis Service and Alarm Service have three variants each namely *MedicalAnalysisService1*, *MedicalAnalysisService2*, *MedicalAnalysisService3* and *AlarmService1*, *AlarmService2*, *AlarmService3*. Drug Service has two variants which are *DrugService1* and *DrugService2*.

Table 1. shows the goal specification of TAS. The first column lists the goal description. Some goals are local goals, for example, the first goal that states that each service must have . Some goals are global, for example, the third goal requires checking whether at least one service has $failure\,rate \leq threshold$. This requires considering all the services and so, it is a global goal. The goal type column lists the type of the goal according to the optimization problem considered. The proposed approach considers three types of goals which are threshold-based, minimization and maximization. Three threshold-based and one minimization goals are defined for TAS.

Table 1. Goal Specification of Tele Assistance System

Goal	Metric $failure\,rate \leq threshold$	Scope	Goal Type
Each service failure rate must be less than or equal to a specific local threshold	Failure Rate	Local	Threshold-based
Each service must have response time less than or equal to a local threshold	Average Response Time	Local	Threshold-based
At least one service must have failure rate less than a specific threshold	Failure Rate	Global	Threshold-based
Average cost per service must be minimized	Average Cost Per Service	Global	Minimization

4. Problem Formulation

In this section, the problem addressed in this paper is formalized. At first, some definitions are given.

Definition 1: *A **state** is a combination of encoded metric values. Here, encoded metric values are the ones classified*

into value ranges (e.g., 0-2 seconds response time as low, 2-4 seconds as medium and >4 seconds is high).

A state can be defined either for a single or multiple locally managed systems. When states are defined combining multiple locally managed system states, it is called the *joint state*. If $s_1, s_2, \ldots, s_n$ are the states of *n* locally managed systems, $S = s_1 x s_2 x \ldots x s_n$ is the joint state.

Definition 2: *An **action** is a variant selection by each of the locally managed systems. Joint action is defined similarly as Joint state.*

For example, each TAS service (locally managed system) calculates the state $s_i = fr_i, t_i, v_i$ where fr_i, t_i and v_i are the failure rate, response time and cost of the service in the i^{th} time instant. For two services s_1 and s_2, their joint state can be written as $S_i = \{\{fr_i^1, t_i^1, v_i^1\}, \{fr_i^2, t_i^2, v_i^2\}\}$. Additionally, the action set of the Medical Analysis Service is {*MedicalAnalysisService1*, *MedicalAnalysisService2*, *MedicalAnalysisService3*}. Decentralized self-adaptation executes the action with the maximum goal conformance.

Definition 3: *The function that measures goal conformance is called the reward function. Generally, better goal conformance leads to higher reward.*

In this paper, three types of reward functions are considered for the three types of aforementioned goals. These reward functions are mentioned below.

$$R_\theta = \begin{cases} \dfrac{2\times\theta - m}{2\times(\theta - m_{max}\}} & if\ \theta - m > 0 \\ \dfrac{1}{2} - \dfrac{\theta - m + 1}{2\times(\theta - m_{max} + 1)} & if\ \theta - m < 0 \\ \dfrac{1}{2} & if\ \theta - m = 0 \end{cases} \quad (1)$$

$$R_< = \frac{1}{m+1} \quad (2)$$

$$R_> = \frac{m - m_{max}}{m_{max} - m_{min}} \quad (3)$$

The first equation is for threshold-based goals where θ is the threshold. Observe that this reward function also provides output maintaining a threshold of 0.5. That is, A less than 0.5 value is generated if the goal is violated and the opposite otherwise. This is required for threshold-based global reward calculation (Equation (8)). The minimization and maximization reward functions are defined by Equation (2) and (3) respectively. Here, *m* is the current value of the metric.

A composite reward function is defined as the weighted summation of all the reward functions.

$$R_{sum} = w_1 \times R_1 + w_1 \times R_2 + \cdots + w_n \times R_n \quad (4)$$

Each local control loop aims to choose an action in a specific state that maximizes its composite reward. More formally, a $S\,x\,A \rightarrow R_i$ needs to be calculated for each i^{th} locally managed system. Then, in a specific joint state S, a joint action A is selected that maximizes each locally managed systems' rewards R_i. In this case, the following observations are made.

1. If there are n_e encoded metric values, n_m metrics and n_a actions for each n_s locally managed system, the total number of joint state, action combination is $n_e \times n_m \times n_a \times n_s^2$. As mentioned previously, a reward value needs to be maintained for each of the joint state, action combination in each locally managed system. As this is a large number, this indicates large coordination overhead.
2. For TAS, it is observed that in some states only local goals were violated but global goals were satisfied. As only local goals are violated, decisions can be taken individually in these states. It means that no coordination is required in these states.

Copyright © 2022 MECS

5. Methodology

This paper proposes a decentralized self-adaptation approach based on multiagent Q-learning and Interaction-Driven Markov Games (IDMG) [2]. Partial knowledge is considered by using only a single peer for action selection in states where coordination is required. In other states, action selection decision is taken individually. In the following sections, the traditional approach of self-adaptation using multiagent Q-learning and the proposed approach using IDMG are described.

5.1. Multiagent Q-learning

Before discussing multiagent Q-learning, Markov game needs to be defined. A Markov game is a tuple (n, S, $A_{1\ldots n}$, $R_{1\ldots n}$, λ) where,

1. n is the total number of locally managed systems
2. S is the set of joint states of all the locally managed systems
3. A_i is the set of actions of the i^{th} locally managed system
4. R_i is the reward function of the i^{th} locally managed system. If multiple goals need to be satisfied, R_i is calculated by accumulating all the reward functions following Equation (4).
5. λ is the transition function defined by $\mathrm{S}\times\mathrm{A}_1\times\mathrm{A}_2\times\mathrm{A}_3\times\mathrm{A}_\mathrm{n}\rightarrow\mathrm{u}(\mathrm{S})$. Here, $u(S)$ is the discrete probability distribution over the joint states [15].

Considering decentralized self-adaptive systems as Markov game, the target is to select the maximum reward joint action in a specific state for all the locally managed systems. This can be done using multiagent Q-learning. Here, each locally managed system maintains a long-term reward value, also called Q-value, for each joint state and action combination. After each iteration, it is updated using the following equation.

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha \times \left[R_i(s,a) + \gamma \max_{a_i} Q_i(s', a_i) - Q_i(s,a) \right] \tag{5}$$

Here, $R_i(s, a)$ is the reward value of the i^{th} locally managed system. Q-value is updated using the temporal difference, which is the difference between the discounted maximum Q-value in the next state and the Q-value in the current state. Here, γ is the discount factor which controls the importance of the future reward. Moreover, α is the learning rate. Both γ and α has values between 0 and 1 [15].

Each control loop selects its action based on the maximum Q-value in each of the states. That is, $\mathrm{a}_i = \mathrm{argmax}_{a_i}\mathrm{Q}_i(\mathrm{s},\mathrm{a}_i,\mathrm{a}_{-i})$ is chosen where a_{-i} is the actions of the other locally managed systems rather than the i^{th}. This multiagent Q-learning based approach is the common framework for decentralized self-adaptive systems. However, as Q-values are learned for each joint states and actions, the problems mentioned in Section 4 is prevalent.

5.2. IDMG-Based Decentralized Self-Adaptive System

The previous section indicates that using Markov games is not useful for designing scalable decentralized self-adaptive systems. So, the proposed technique approaches the decentralized self-adaptation problem following the Interaction-Driven Markov Game (IDMG) model [2]. In this case, each local control loop takes action selection decision considering the decisions of one of its peers. However, this joint action selection is done only in states where coordination is required. In other states where coordination is unnecessary, each local control loop decides individually. Therefore, it is evident that IDMG directly solves the problems put forward in Section 4.

An IDMG is defined as a tuple (M_1, M_2, I) where M_1 and M_2 are Markov Decision Processes (MDP) of the two peer locally managed systems and I is an interaction game [2]. Each MDP $M_i = (S_i, A_i, R_i, \lambda_i)$ is defined as follows,

1. S_i is the set of individual states of a locally managed systems.
2. A_i is the set of actions of the i^{th} locally managed system
3. R_i is the reward function of the i^{th} locally managed system
4. λ_i is the transition function defined by $S_i \times A_i \rightarrow u(s_i)$.

Hence, M_1 and M_2 are basically Markov games when joint states and actions are replaced by individual states and actions, that is, coordination is not considered. The interaction game $I=\left(2,S^I,\left(A_1,A_2\right),R^I,\lambda\right)$ is a Markov game with both of the peers as defined in the previous section. Here, $S^I \subset S_1 \times S_2$ is the set of states requiring coordination. These states need to be learned. In these states, Q-values are maintained considering the joint states and actions and in other states individual states and actions are used similar to the MDPs.

Each agent reward function in IDMG is as follows.

$$R_i = r_i\left(s_i,a_i,s_i'\right)+r^I\left(s,a,s'\right) \tag{6}$$

Here, i^{th} locally managed system has transitioned from state s_i to s_i' after selecting action a_i. Considering the action selection of both the peers, a joint state transition from s to s_i has occurred. $r_i\left(s_i,a_i,s_i'\right)$ is the reward value of the MDP and $r^I\left(s,a,s'\right)$ is the coordination reward value of the interaction game. If $s' \subset s^I$, $r^I\left(s,a,s'\right)\neq 0$; and $r^I\left(s,a,s'\right)=0$ otherwise [6]. Hence, in coordinating states, the interaction game reward function will provide a nonzero value.

As mentioned previously, local goals are specific to the locally managed systems and so, these can be satisfied without considering the coordination. So, in states where global goals are always satisfied, Q-learning is conducted individually. That is, in these states, the aforementioned MDPs are used. So, the reward function becomes $R_i = r_i\left(s_i,a_i,s_i'\right)$ in these states, where $r_i\left(s_i,a_i,s_i'\right)$ is the composite reward calculated according to Equation (4). Now, coordination is required where global goals are violated. However, although the violation of global threshold-based goal is easy to detect, identifying the global minimization and maximization goal violations is harder. So, the global minimization and maximization are partitioned into smaller local optimization problems. For example, the fourth goal from Table 1. can be partitioned into the problem where each of the services will attempt to minimize its own cost locally. This is represented by the following Equation.

$$R_<^i = \frac{1}{m_i+1} \tag{7}$$

Therefore, rather than minimizing the average total cost of all the services, each i^{th} service considers minimizing their own cost m_i. This is a local reward function which can be used similarly to the other local reward functions.

In case of threshold-based global functions, it has been discussed that in coordinating states this should return a nonzero value and zero otherwise. To support this, Equation (1) is modified as follows.

$$R_\theta^I = \begin{cases} 0 & if\ 0.5-R_\theta > 0 \\ -R_\theta & if\ 0.5-R_\theta < 0 \end{cases} \tag{8}$$

The equation returns a negative value if goal violation occurs. This can be considered as a miscoordination penalty. It helps to discourage the selection of action that leads to miscoordination penalty resulting in lower total reward from Equation (6).

Algorithm 1 shows the IDMG-Based Q-Learning for a locally managed system. At first, a pseudo-action σ is added to the available action set. This action helps to learn the coordinating states. Two types of Q-values are maintained which are Q_i and Q_i^I. The first one is for the states without coordination and the second one is for the coordinating states or for the interaction game. At first, an action is selected based on Q_i. If this action is the pseudo-action, then it is considered as a coordinating state. In this state, an action is selected based on the value of Q_i^I. As this is considered as a coordinating state, the action and the state of the peer are perceived and $Q_i^I(s,a)$ is updated using the joint action and state. In the Q_i^I update at line 11, two observations are made. Firstly, rather than choosing the next action that maximizes Q_i^I, the next action that maximizes Q_i is used. This is for maintaining the sparseness of Q_i^I. Secondly, observe that $r_i\left(s,a\right)$ is calculated from Equation (6). So, when $r^I=0$, the Q-value update becomes similar to the Q-value update at line 15. Hence, Q_i^I is updated considering the coordination reward only at coordinating states, which means that coordinating states are learned properly. However, if the selection action is not a pseudo-action the reward and next state is observed and the local control loop updates the Q_i individual Q-value.

Copyright © 2022 MECS

Algorithm 1 Algorithm for IDMG-Based Q-learning

1: $a_i = a_i \cup \{\sigma\}$
2: $Q_i(s_i, a_i) \leftarrow 0, \forall s_i, a_i$
3: $Q_i^I(s, a) \leftarrow 0, \forall s, a$
4: **while** $TerminationCondition \neq true$ **do**
5: $a_i \leftarrow selectAction(Q_i, s_i)$
6: **if** $a_i = \sigma$ **then**
7: $a_i^I = selectAction(Q_i^I, s)$
8: $a_{-i}^I \leftarrow receiveOtherAgentActions()$
9: $a \leftarrow a_i^I \cup a_{-i}^I$
10: $observe\ state\ transition\ s_i'\ and\ reward\ r_i$
11: $Q_i^I(s, a) \leftarrow Q_i^I(s, a) + \alpha \times [r_i(s, a) + \gamma \max_{b_i} Q_i(s_i', b_i) - Q_i^I(s, a)]$
12: **else**
13: $observe\ state\ transition\ s_i'\ and\ reward\ r_i$
14: **end if**
15: $Q_i(s_i, a_i) \leftarrow Q_i(s_i, a_i) + \alpha \times [r_i(s_i, a_i) + \gamma \max_{b_i} Q_i(s_i', b_i) - Q_i(s_i, a_i)]$
16: **end while**

Algorithm 2 shows the *selectAction* function. For action selection, the ε-greedy strategy is followed [5]. A ϵ value is prespecified and actions are selected greedily according to the Q-values with probability 1- ϵ (line 1-4). Otherwise, a randomly selected action is returned. This helps to explore rather than only to exploit, which may result in finding a more optimal action [5].

In this way, each local control loop chooses actions based on individually maintained Q-values in non-coordinating states and joint Q-values in coordinating states. Moreover, each local control loop coordinates with only one peer for Q-value calculation. So, the problems formulated in Section 4 is solved utilizing the IDMG-Based Q-learning technique.

Algorithm 2 Algorithm for selectAction Function

Require: Q and s
1: $\epsilon \leftarrow k$
2: $p_r \leftarrow random(0, 1)$
3: **if** $p_r < (1 - \epsilon)$ **then**
4: **return** $\arg\max_{a_i} Q(s, a_i)$
5: **else**
6: **return** $a_i \leftarrow randomSelect()$
7: **end if**

6. Experiment

To evaluate the proposed approach, the aforementioned TAS exemplar was used [7]. The existing TAS implementation is centralized. Hence, it was modified to support decentralized adaptation by adding a control loop to each service. The implementation was done in Java and it utilizes the Research Service Platform (ReSeP) for service-based systems by Weyns et al. [7]. Each local control loop contains reward calculation, state data reader, action selection and IDMG-Based Q-learning components. The reward calculation component uses current state information (i.e., current values of the metrics) to calculate total reward values using Equation (6). The state data reader component collects current values of the metrics and encodes these to get the current state. In case of coordinating states (i.e., when σ is selected), this component requests and fetches current state data from a predefined peer. The action selection and IDMG-Based Q-learning component implements Algorithm 1 and 2 respectively.

For TAS, among the metrics in Table 1, failure rate was calculated by dividing the total number of failures by invocation count. Average response time was calculated in milliseconds. The goal stating that average cost per service must be minimized was partitioned as mentioned previously. In this case, each service needs to calculate its own cost in each iteration. The cost of every service was predefined in the existing TAS implementation.

Algorithm 1 and 2 show that some parameter values need to be specified before running the experiment. After experimenting with the proposed approach on TAS, the parameter values that produced the best result were chosen. α and γ were chosen as 0.1 and 0.9 respectively for all the local control loops. ϵ was chosen as 0.9, 0.9 and 0.85 for Medical Analysis Service, Alarm Service and Drug Service local control loop respectively. Finally, TAS was run

comparing the proposed approach with two other approaches - random action selection and individual Q-learning. The first one selects actions randomly and the second one completely ignores coordination and learns individually. The proposed approach is also compared to a joint Q-learner similar to Section 5.1 and [4] to observe the effect of using partial knowledge instead of global knowledge. The objective is to evaluate whether the proposed approach can reduce coordination overhead without harming goal conformance. Following the literature, goal conformance is evaluated and compared using the total rewards accumulated for each adaptation decision [2,3,6]. Coordination overhead is calculated by comparing the learning time of both of these algorithms.

7. Results and Discussion

Fig.1. compares the total observed reward from the proposed technique, random action selection and individual Q-learning. It is seen that the reward values from the proposed approach are higher in most cases and more stable than the other two approaches. The reward values from random action selection vary a lot which is obvious due to the randomness in this technique. The variation in reward values is also higher in the individual Q-learning approach than the proposed one. Moreover, up to about 600 iterations, these three approaches perform almost similarly. However, the proposed approach results in overall higher total reward value in the subsequent iterations. A Wilcoxon rank sum test was performed to see whether the mean total reward value in the proposed approach is significantly higher than the other two approaches [16]. The p-values (1.14*e*-119 for random and 7.95*e*-117 for individual Q-learning) less than the significance level 0.05 in both cases indicate that the total reward obtained from the proposed approach is significantly higher. This high reward is obtained as coordination is taken into consideration in coordinating states, rather than individual Q-learning and random action selection which do not consider coordination at all.

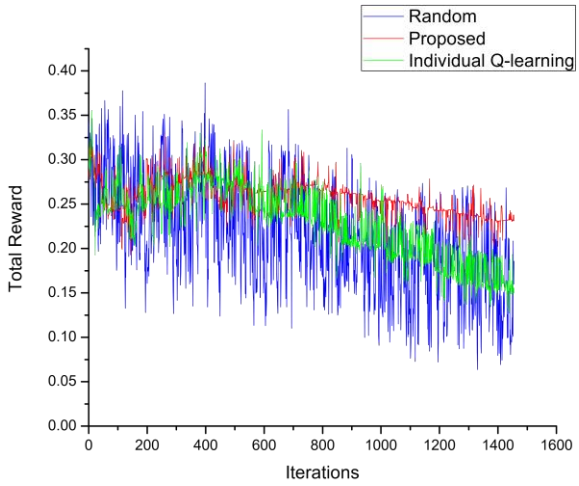

Fig.1. Comparison of Total Rewards among the Proposed Approach, Random Action Selection and Individual Q-learning.

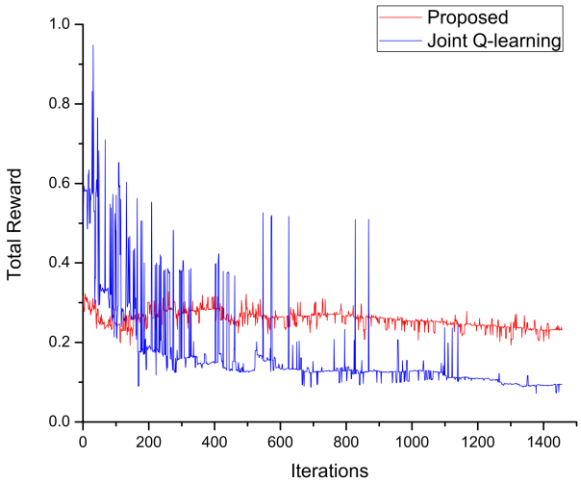

Fig.2. Comparison of Total Rewards among the Proposed Approach and Joint Q-learning.

Fig.2. shows the total rewards of the proposed approach using IDMG-Based Q-learning and joint Q-learning. The joint Q-learning approach considers global knowledge (i.e., all the locally managed systems) for learning. So, it should result in higher total reward than the proposed partial knowledge-based approach. However, the figure shows that

Copyright © 2022 MECS

although the total reward from the joint Q-learning is higher up to about 200 iterations, it is counterintuitively lower afterwards in most cases. This can be explained as follows. For considering the global state space, the joint Q-learning results in much higher message passing for coordination. Moreover, each local control loop needs to maintain a large table of Q-values for large global state space. As a result, the Q-value update becomes computation-intensive and time-consuming. So, as the Q-value update is much slower, the action selection continuously considers outdated Q-values which becomes nonoptimal soon. This situation resembles the timeliness-impact of making adaptation decisions described by Weyns et al. [1]. This situation is confirmed by Fig.3. which shows the learning time of the two approaches. The learning time of the proposed approach is much lower in most cases. The few cases where the proposed approach had high learning time were analyzed. It was seen that high learning time occurred either because the corresponding state was a coordinating state or due to implementation platform issues (e.g., JVM garbage collection).

From the results, it is evident that the proposed approach results in higher total reward than approaches where coordination is ignored or global knowledge is used for coordination. The results show that considering only a single peer does not significantly affect goal conformance. The proposed approach takes the decisions with better goal conformance leading to higher reward sooner than global knowledge based joint learner due to lower state-action space complexity.

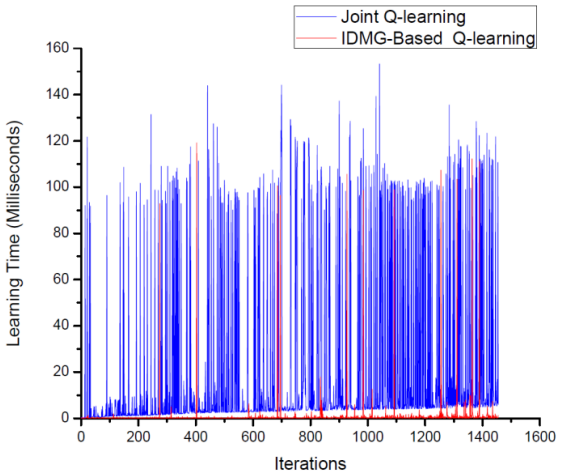

Fig.3. Comparison of Learning Time among the Proposed Approach and Joint Q-learning.

8. Conclusion and Future Work

In this paper, a decentralized self-adaptation approach has been proposed to reduce coordination overhead resulting from global knowledge incorporation. For this, the decentralized self-adaptation is considered as an IDMG and solved with Q-learning for this type of game. This IDMG-Based Q-learning uses only a single peer information which satisfies the partial knowledge assumption. However, it uses the peer information only in states where coordination is required. In other states, it learns individually. The proposed approach is applied to a Tele Assistance System where a control loop is assigned to each of the services. It is compared with a random action selection method and individual Q-learning without considering coordination. It was seen that the proposed approach resulted in much higher reward than the other two approaches. It was also compared to a joint Q-learning method which uses global knowledge. It was observed that the proposed approach reached goal conformance faster because joint Q-learning based technique has a large state-action space and adaptation does not occur in a timely manner. Therefore, it can be concluded that IDMG-based Q-learning, when applied with single peer for decentralized self-adaptation, can reduce coordination overhead and take effective adaptation decisions.

In this work, the incorporation of single peer control loop has been empirically shown to be effective. However, mathematical proof of convergence and optimality is an important future work. The selection of peer control loop is predetermined in our approach. However, rather than using prespecified peer control loops, these can be chosen in a probabilistic way so that the peer that results in highest goal conformance in near future is selected dynamically [17]. More research is needed in this regard to understand which peer selection method is the most optimal one. Another research direction is to extend IDMG-based Q-learning with deep reinforcement learning based approaches to take more optimal decisions in high-dimensional state space.

Acknowledgment

This research has been partially supported by The University Grant Commission, Bangladesh under the Dhaka

University Teachers Research Grant. Reference No: Regi/Admin-3/14857.

References

[1] D. Weyns, S. Malek, J. Andersson, On decentralized self-adaptation: lessons from the trenches and challenges for the future (2010) 84–93.
[2] M. T. Spaan, F. S. Melo, Interaction-driven markov games for decentralized multiagent planning under uncertainty, in: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 525–532.
[3] V. Grassi, M. Marzolla, R. Mirandola, Qos-aware fully decentralized ser-vice assembly, in: Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2013 ICSE Workshop on, IEEE, 2013, pp. 53–62.
[4] H. Wang, Q. Wu, X. Chen, Q. Yu, Z. Zheng, A. Bouguettaya, Adaptive and dynamic service composition via multi-agent reinforcement learning, in: Web Services (ICWS), 2014 IEEE International Conference on, IEEE, 2014, pp. 447–454.
[5] C. Claus, C. Boutilier, The dynamics of reinforcement learning in cooperative multiagent systems, AAAI/IAAI 1998 (1998) 746–752.
[6] F. S. Melo, M. Veloso, Learning of coordination: Exploiting sparse interactions in multiagent systems, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 773–780.
[7] D. Weyns, R. Calinescu, Tele assistance: A self-adaptive service-based system examplar, in: Proceedings of the 10th International Symposium on Soft-ware Engineering for Adaptive and Self-Managing Systems, IEEE Press, 2015, pp. 88–92.
[8] R. De Lemos, H. Giese, H. A. Muller, ¨M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, et al., Software engineering for self-adaptive systems: A second research roadmap, in: Software Engineering for Self-Adaptive Systems II, Springer, 2013, pp. 1–32.
[9] D. Weyns, B. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, K. M. Goschka, ¨On patterns for de-centralized control in self-adaptive systems, in: Software Engineering for Self-Adaptive Systems II, Springer, 2013, pp. 76–107.
[10] I. Georgiadis, J. Magee, J. Kramer, Self-organising software architectures for distributed systems, in: Proceedings of the first workshop on Self-healing systems, ACM, 2002, pp. 33–38.
[11] D. Sykes, J. Magee, J. Kramer, Flashmob: distributed adaptive self-assembly, in: Proceedings of the 6th International Symposium on Soft-ware Engineering for Adaptive and Self-Managing Systems, ACM, 2011, pp. 100–109.
[12] J. Dowling, R. Cunningham, E. Curran, V. Cahill, Building autonomic systems using collaborative reinforcement learning, The Knowledge Engineering Review 21 (3), 2006, pp. 231–238.
[13] Kishan Kumar Ganguly, Md. Saeed Siddik, Rayhanul Islam, Kazi Sakib, "An Environment Aware Learning-based Self-Adaptation Technique with Reusable Components", International Journal of Modern Education and Computer Science(IJMECS), Vol.11, No.6, pp. 53-64, 2019.DOI: 10.5815/ijmecs.2019.06.06
[14] M. Caporuscio, M. D'Angelo, V. Grassi, R. Mirandola, Reinforcement learning techniques for decentralized self-adaptive service assembly, in: European Conference on Service-Oriented and Cloud Computing, Springer, 2016, pp. 53–68.
[15] A. Nowe, ´P. Vrancx, Y.-M. De Hauwere, Game theory and multi-agent reinforcement learning, in: Reinforcement Learning, Springer, 2012, pp. 441– 470.
[16] H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, The annals of mathematical statistics (1947) 50–60.
[17] Bechar Rachid, Haffaf Hafid,"Distributed Monitoring for Wireless Sensor Networks: a Multi-Agent Approach", IJCNIS, vol.6, no.10, pp.13-23, 2014. DOI: 10.5815/ijcnis.2014.10.02

Authors' Profiles

Kishan Kumar Ganguly received the Bachelor of Science in Software Engineering (BSSE) and Master of Science in Software Engineering (MSSE) degrees from the Institute of Information Technology, University of Dhaka. He is currently working as a lecturer in the Institute of Information Technology, University of Dhaka. His research interest includes applications of machine learning and software engineering for self-adaptive systems.

Moumita Asad received the Bachelor of Science in Software Engineering (BSSE) and Master of Science in Software Engineering (MSSE) degrees from the Institute of Information Technology, University of Dhaka. She is currently working as a lecturer in Independent University, Bangladesh. Her research interest includes automated program repair, software testing and metrics.

Kazi Sakib is a Professor at the Institute of Information Technology (IIT), University of Dhaka, Bangladesh. He received his Ph.D. in Computer Science at the School of Computer Science and Information Technology, RMIT University. His research interests include software engineering, cloud computing, software testing, software maintenance, etc. He is an author of a great deal of research studies published at national and international journals as well as conference proceedings.

How to cite this paper: Kishan Kumar Ganguly, Moumita Asad, Kazi Sakib, "Decentralized Self-adaptation in the Presence of Partial Knowledge with Reduced Coordination Overhead", International Journal of Information Technology and Computer Science(IJITCS), Vol.14, No.1, pp.9-19, 2022. DOI: 10.5815/ijitcs.2022.01.02

Copyright © 2022 MECS