Computing
DOI 10.1007/s00607-013-0337-x

Grid resource discovery based on distributed learning automata

Mohammad Hasanzadeh · Mohammad Reza Meybodi

Received: 29 January 2013 / Accepted: 26 May 2013

Abstract This paper focuses on resource discovery problem for Grid. Grid is a devices and services environment that has evolved with the goal of resource sharing. Grid resource discovery encompasses locating and retrieving computational resources. Existing resource discovery solutions are not well adapted to the dynamicity and heterogeneity of Grid. Query propagation is a novel approach that forwards an unsupported query from its resident peer to an adjacent peer. The concept of next generation intelligent Grid environments needs intelligent modules for resource discovery. Learning automaton is a stochastic tool with learning ability which simply adapts to the progressive environmental changes. The proposed method utilizes a distributed learning automata (DLA) which is a network of learning automata (LA). Here, multiple DLA are used for forwarding domain-specific queries. Different Grid scales are utilized for evaluation of the proposed method. Results demonstrate that the resource discovery based on DLA optimizes resource utilization, maximizes throughput, minimizes response time and avoids overload. Moreover, the algorithm is also scalable, fully distributed and failure-free.

Keywords Grid computing · Resource discovery · Learning automata (LA) · Distributed learning automata (DLA)

Mathematics Subject Classification 68 Computer Science · 68T05 Learning and adaptive systems

M. Hasanzadeh (✉) · M. R. Meybodi
Soft computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
e-mail: mdhassanzd@aut.ac.ir

M. R. Meybodi
e-mail: mmeybodi@aut.ac.ir

1 Introduction

Distributed computing is a community of computational resources which are get together in order to achieve a common goal. Grid computing is a modern issue of distributed computing. Grid computing [1] is a new field of distributed computing that focuses on large scale resource sharing, exquisite applications and high performance computing. Design objectives and target applications for a Grid motivate the architecture of Grid resource discovery module [2]. These properties are placed Grid systems into three categories: computational Grid, data Grid and service Grid. In all of these Grid types, applications require strategies for discovering available resources, configuring their structure and determining their status [3]. Grid resources specify from their characterization of capabilities and categorize into three classes of computational, storage and network resources. Moreover, resources should provide enquiries that show their structure and status.

Resource discovery strategies [4] provide some special quality of service requirements of Grid resources for Grid users. In outer layers, these strategies enable users to query their required resources and in inner layers they use network oriented search mechanisms to locate users required resources. Also resource discovery module provides maintenance, manipulation and propagation of resource information of Grid environment.

The evolving technology of Grid suggests flexible solutions for complex networks such as coordinated resource sharing and problem solving in dynamic, multi institutional virtual organizations. Resource sharing in such a heterogeneous environment needs predefined conditions in terms of users and permissions. A set of institutions, groups, users and resources defined by such sharing policies is nominated as Virtual Organization (VO) [1]. Goals and requirements of all VO members are similar.

Information services are a vital part of any Grid infrastructure which provides fundamental mechanisms of resource discovery and monitoring. Moreover, Grid information system (GIS) [5] is the central component of VO that focused on management and discovery of Grid resources. GIS supports arbitrary views on VO member resources with user specific range queries. GIS is responsible for resource information dissemination through the Grid overlay. The architecture of information propagation could be centralized, decartelized [6,7] peer to peer (P2P), hierarchal and super-peer [8]. In decentralized architecture, resource information of each VO is locally preserved by its associated GIS. Decentralized GIS architecture can alleviate the burden of single server bottleneck by requests multiplexing on different Grid resources.

Learning automata (LA) [9,10] are probabilistic, simplex and self-operating decision making gadgets. While placing in an unknown environment, the automaton probabilistic structure gives it free will to reach the supreme goal by taking actions and receiving feedbacks from the environment. Because of the automaton simplicity, the supreme goal defines as detecting the optimal action from a set of actions. The goal of automaton is reached after several episodes of self-acting towards the environment. Since the first introduction of LA by Narendra in 1970s [9], there have been many applications of this learning method such as designing guard channel algorithms [11], solving coverage problem in directional sensor networks [12], deployment of mobile wireless sensor networks [13] and implementing evolutionary algorithm [14,15].

Distributed learning automata (DLA) is a new network of interconnected LA which is introduced by Beigy [16] and researchers find its valuable applications in solving shortest path problem, finding connected dominating set [17], determining web pages recommendation [18], designing wireless push systems [19], solving maximum clique problem [20] and etc.

The proposed model is a multi-layer architecture for locating users requested resources. The model considers a layer for each key component of Grid. The resource discovery module is handled by DLA layer. This layer is a graph isomorphic to the Grid topology. Each node of DLA represents a VO of Grid and each edge of DLA indicates a neighbor VO. In case of an unsupported request, the associated automaton of current VO is responsible for sending resource query to another VO. The resource request is forwarded until query expiration time.

The reminder of this paper is organized as follow: Sect. 2 provides the background of resource discovery heuristics. Section 3 briefly introduces LA theory. Section 4 describes the Grid resource discovery based on DLA in detail and presents the associated algorithm. Section 5 gives the simulation and performance evaluation. Finally, Sect. 6 concludes the paper and outlines future researches.

2 Grid resource discovery strategies

The method for resource discovery in Grid systems presented in Imantchi and Foster [6,7] is based on three statements that optimize the system performance: (1) one or more peers per VO, (2) self-centered resource indexing by each peer and (3) different request forwarding strategies. The reminder of this section reviews some recently proposed systems that adopt request forwarding approach to Grid resource discovery.

2.1 Request forwarding strategy

In [6,7] Imantchi and Foster propose four request forwarding strategies. Each participant of the proposed architecture has a VO that stores and provides access to local resources for other nodes. Users send their requests to a locally known node. The node responds with the matching resource if it has them locally, otherwise it forwards the request to another node. In these strategies in addition to neighbors' addresses, each node can maintain a history of its neighbor's responses. These strategies choose the appropriate node to forward the resource request to it. These strategies provide a range of different request forwarding mechanisms, including: (1) *random walk*, (2) *experience-based* + *random*, (3) *best-neighbor* and (4) *experience-based* + *best neighbor*.

2.2 Genetic algorithm strategy

In [21] Boroumand et al. propose a genetic algorithm (GA) based resource discovery mechanism. Their proposed architecture uses three units to locate the requested resource with optimal value of hop count and success rate. (1) The *statistical tables unit* that store information about number of successful results and average number of hops for the previous queries through each peer in the neighborhood, (2) the *resource checking and query evaluation unit* which is used to observe peer resources whenever a query is created and (3) the *genetic operation unit* which applies genetic operations to the population of peer based on the information of statistical tables.

2.3 Ant colony strategy

Multi-agent system (MAS) is a computational system where a set of autonomous agents interact within an environment [22]. Nowadays, ant colonies are used as MDSs to solve different problems. In [23] Deng et al. propose a large-scale P2P Grid system which employs an ant colony optimization (ACO) algorithm to locate the required resources. The system is designed of an overlay network which is built on the top of existing network. Each node of the system is modeled as one nest that can initiate ant. Each nest is composed of four modules: (1) *Grid service module* monitors and manages the Grid services, (2) *routing table module* saves the routing information of ants, (3) *ant management module* is responsible for generating and killing ants and (4) *communication module* manages the movement of ants between nests. Also each ant consists of two modules: (1) *life cycle management module* keeps the TTL value of ant and (2) *memory management module* stores the requested resource and the path which it has passed.

2.4 Super-peer strategy

A super-peer operates both as a centralized server at the cluster level, where cluster size is the number of nodes in the cluster and a regular node at the super-peer level [24]. In [8] Mastroianni et al. adopt a super-peer Grid resource discovery protocol. The Grid overlay network is partitioned into several Physical Organizations (POs). Each PO (cluster) is included a set of Grid nodes within one administration domain. Also, a set of high available Grid nodes in each PO can be chosen as super-peers.

3 Learning automata

A learning automaton [9,10] can be modeled as an abstract entity with finite number of actions. Learning automaton operates by selecting one of its actions and employing it to the environment. The action is evaluated by the environment and the learning automaton uses the environment response to select the next action. Along this procedure learning automata gradually learns to select optimal action. LA have wide range of applications in cellular mobile networks [11], wireless sensor networks [13], Grid computing [25], wireless mesh networks [26] and etc. Moreover, learning automaton has application in global optimization problems [27], such as particle swarm optimization (PSO) [28,29].

3.1 Variable structure learning automata

Variable structure learning automata (VSLA) [9,10] can be modeled as a quadruple $\{\alpha, \beta, P, T\}$ where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of actions, $\beta = \{\beta_1, \beta_2, \ldots, \beta_r\}$

is the set of inputs, $p = \{p_1, p_2, \ldots, p_r\}$ is the set of action probabilities and $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ is the learning algorithm. If $p(n+1)$ is a linear function of$p(n)$ the schema is called linear, otherwise it is nonlinear. In the simplest form of VSLA consider an automaton with r actions in a stationary environment where, $\beta \in \{0, 1\}$ is included as inputs. After selecting an action by automaton, the *positive* and *negative responses* ($\beta = 0$ and $\beta = 1$) are updated through (1) and (2), respectively:

$$p_j(n+1) = \begin{cases} p_j(n) + a.(1 - p_j(n)) & if\ i = j \\ p_j(n).(1 - a) & if\ i \neq j \end{cases} \tag{1}$$

$$p_j(n+1) = \begin{cases} p_j(n).(1 - b) & if\ i = j \\ \frac{b}{r-1} + (1 - b).p_j(n) & if\ i \neq j \end{cases} \tag{2}$$

In (1) and (2), a and b are called *learning parameters* and they are associated with the reward and penalty signals. A linear reward–penalty ($L_{R\text{-}P}$) learning algorithm is considered that a and b are equal.

3.2 Distributed learning automata

A DLA [16] is a network of LA which collectively cooperate to solve a particular problem. A DLA can be modeled by a directed graph in which the set of graph nodes constitutes the set of LA and the set of outgoing edges for each node constitutes the set of actions for corresponding learning automaton. When a learning automaton selects one of its actions, another learning automaton on the other end of edge corresponding to the selected action will be activated.

Formally a DLA with n LA can be defined by a graph (A, E), where $A = \{A_1, A_2,\ldots,A_n\}$ is the set of automata and $E \subset A \times A$ is the set of edges in the graph in which an edge (i, j) corresponds to action α_j of automaton A_i. Let action probability vector for learning automaton A_i represents by $\widehat{p_i}$ where a component$\widehat{p}_i^j$ of $\widehat{p_i}$ denotes the probability of choosing action α_j, that is the probability of choosing edge (j, m).

Since introducing DLA by Beigy and Meybodi [16], DLA has received a lot of applications in solving various problems such as: shortest path problem [16], traveling salesman person (TSP), Steiner tree problem, minimum spanning tree, vehicle routing problem, connected dominating set (CDS) problem in backbone formation of wireless sensor networks [17] and web page recommendation [18].

4 Grid resource discovery based on distributed learning automata

Assume a Grid with n peers. Each peer in Grid contains a set of resources that contributes to a VO. The registered resources of each VO are managed by a GIS. Also each peer directly connects to a set of arbitrary peers with connection links and can communicate with them. If a user wants to join the Grid, it will connect to the nearest VO of Grid and the user's resource request will send to VO's local GIS. Then, VO searches its local resources by querying its designated GIS. If the requested resource is found, the request will service successfully. Otherwise, the request will forward to other peers of Grid. Please note that for the sake of simplicity, from now on each VO of Grid is considered as a peer.

In the following the details of proposed resource discovery algorithm will be discussed. The proposed algorithm utilizes a DLA (network of LA) in order to fulfill the task of resource discovery. Each automaton is associated to a GIS and its role is to forward the resource request to one of its adjacent neighbors, if the request could not processed by local resources.

4.1 Request forwarding

In request forwarding strategy, the problem is to select the proper node while forwarding the resource request to another node from the local neighborhood. In order to select a proper node, in the following a new request forwarding strategy is introduced which utilizes learning automaton as the determiner of the node which the request is transmitted to. At first, a DLA associated with the Grid is generated in which the set of nodes in the Grid constitutes the set of automata and the set of outgoing edges for each node constitutes the set of actions for corresponding automaton to them the request could be forwarded. In other words, each VO maintains a learning automaton entity for query forwarding. When an automaton selects one of its actions, the resource request will be forward to the VO on the other end of the edge corresponding to the selected action. Furthermore, the resource request will be process by the local GIS of corresponding VO. Please note that, the GIS architecture is considered fully decentralized and the resources information of each VO is maintained locally by its associated GIS. This type of architecture removes the bottleneck of a centralized directory for resource searching.

4.2 Resource query

The resource discovery service supports multi-attribute range queries per each resource type. This feature enables resource discovery service to search the requested resource more efficiently and accurately. Assuming R resources of D distinct types, all resources have the same frequency $F = R/D$. The total range of each resource type is divided into multiple sub-ranges. Also, assume a DLA corresponding with each sub-range. So, for D distinct resource types that their ranges are divided into r sub-ranges, $D \times r$ groups of DLA is needed for request routing. After initiating each query, the corresponding DLA of specified resource sub-range is responsible for resource query forwarding. Also by finishing the request, the corresponding probability vectors of DLA will be updated.

4.3 Proposed resource discovery model

The embedded learning automaton of each node has the functionality to choose a peer and forwards the request packet to it. Figure 1 shows different layers of

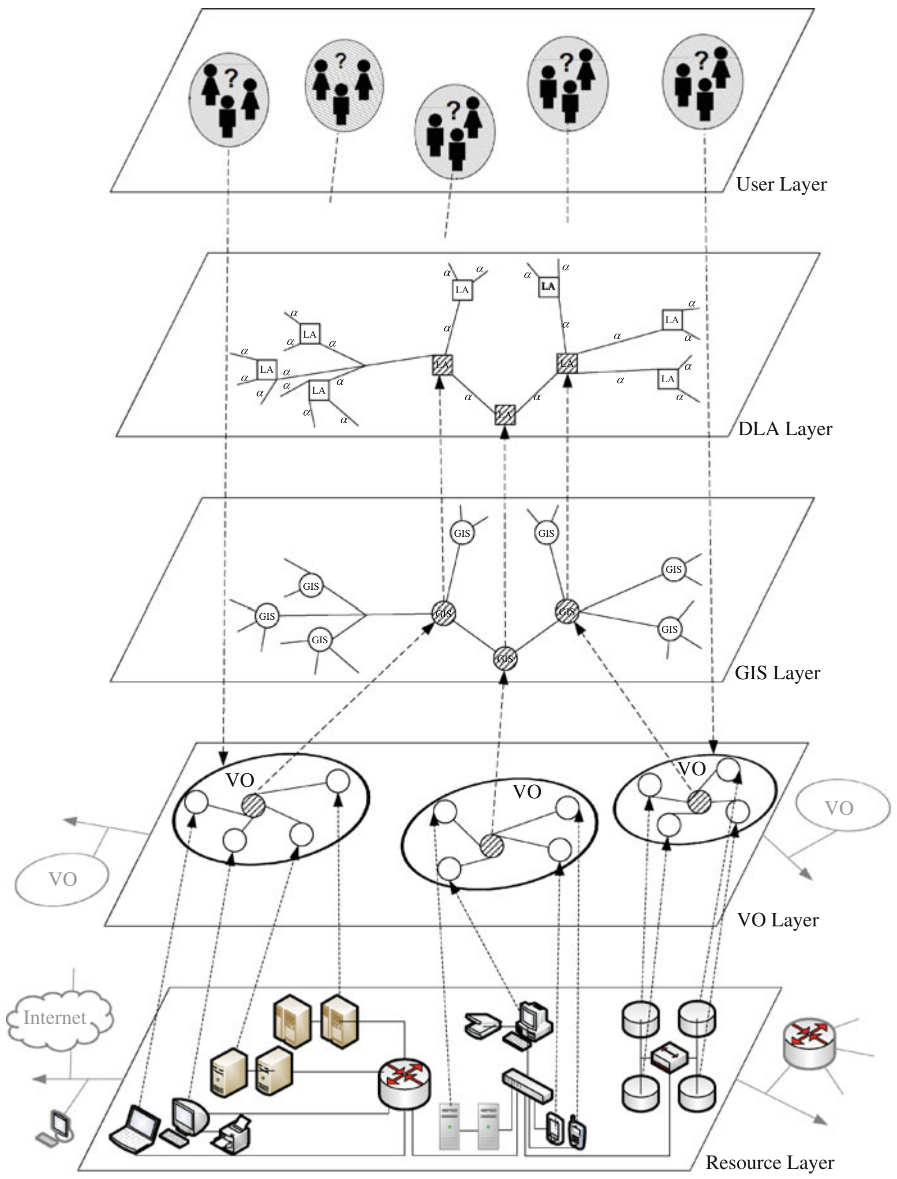

Fig. 1 Multi-layer resource discovery architecture

proposed multi-layer decentralized resource discovery model, including: (1) the *resource layer* that consists of various Grid resource types, (2) the *VO layer* that enforce same policies into member resource/users, (3) the *GIS layer* that updates and retrieves local resource information, (4) the *DLA layer* that forwards the user resource requests and (5) the *user layer* that connects users to the nearest VO.

4.4 Proposed resource discovery workflow

The following is the key procedure of resource discovery algorithm. The resource discovery algorithm based on DLA is nominated as RDDLA. Algorithm 1 is the pseudocode of RDDLA. Description of this algorithm is as follows:

(1) The *Grid overlay* consists of resources that are assigned to VOs and the GIS entity of each VO is responsible for them. (2) The *VO membership* is done through random assignment of resources to VOs. (3) The *user connection* is established through connecting the nearest VO. (4) The *query range resolution* divides the resource characteristics into sub-ranges. (5) The *learning automata assignment* provides different classes of DLA for different characteristics of different resources. (6) The *query preprocessing* is done to invoke the corresponding class of DLA responsible for request routing. (7) The *local resource processing* is done through the checking of local resources of starter VO by its GIS module. (8) The multiple resources of VO are processed by first find first served (FFFS) *processing policy*. (9) The learning automata module of VO will forward an unsupported request to one of neighbor peers in order to *global resource processing*. 10) The *query expiration* is happened if the TTL expires. (11) The *path planning* is done through saving the traversed path of request. (12) The *feedback deployment* of DLA is deducted through series of learning automata actions corresponding to the stored path.

Algorithm 1 RDDLA algorithm

```
define
Construct a graph $G = (V,E)$ where $V$ is a set of Grid Information Services (GISs) and $E$ is a set of network links
Consider start node $v_s$, end node $v_e$, hop number $k = 0$ and maximum hop number $K$
Construct a DLA from graph $G$
begin
    for each user request do
        let $v_u$ be source node $v_s$
        while (resource in not find AND the number of traversed edges is less than $|V|$AND $k$ is lower than $K$) do // Resource discovery
            Automaton $A_u$ Selects an adjacent node to $v_u$ from its action set $\alpha_u$ according to its action probability vector $P_u$ and denote it $v_w$
            if ($v_u$ is not visited)
                if (GIS$_u$ is enable)
                    Check GIS$_u$ local resources based on FFFS policy
                end if
                Add ($v_u$, $v_w$) to the traversed path and set $v_w$ to $v_u$
            end if
            Increament hop number $k$
        end while
        set $v_w$ to $v_e$
        Let $\pi = \{v_s, \ldots, v_e\}$ be the traversed path
        if (the resource is find) then // Reward path
            Reward the selected actions of activated automata along path $\pi$
        else // Penalized path
            Penalize the selected actions of activated automata along path $\pi$
        end if
    end for
end
```

5 Numerical analysis

5.1 Experimental setup

This section is devoted to evaluate the performance of RDDLA algorithm in GridSim simulator. Gridsim [30] is a java-based Grid simulation toolkit which developed by CLOUDS laboratory of university of Melbourne under the supervision of Professor Dr. Buyya.

The initial Grid topology is derived from Waxman model [23] that connects nodes by *Waxman's probability model*. The connectivity degree is the key parameter of this topology is the. This parameter indicates the outdegree and indegreee of each node. All links have same bandwidth and are full-duplex. Also, 3 different Gird sizes are used for the experiments: small, medium and large scale Grids.

There are 5 different types of resource are randomly scattered in the simulated Grid, including: processor, memory, storage, operating system and network media. Each resource is drawn from Poisson distribution with $\lambda = 7$. Moreover, 100 resources of each type are dedicated to the Grid overlay and each resource range is divided into two subranges.

The user requests distribution follows Poisson distribution with $\lambda = 7$. Also, there are 30 users in resource discovery scenario where a same set of 250 (50 requests per each resource) requests belong to them. Furthermore, in each experiment the initial node is chosen randomly from a set of ten nodes. Finally, the TTL value of requests is set to 100 hops.

The proposed algorithm compares with three other state-of-the-art resource discovery strategies, including: (1) the *random walk resource discovery* (*RDR*) [31] that randomly chooses a peer to forward the request, (2) the best-neighbor resource discovery (RDBN) [6,7] that records a history about previous successful request and forwards the request to the most respondents neighbor and (3) the *flooding resource discovery* (RDF) [32] that spreads out the request based on flooding protocol.

The *Hop number*, *query hits* and *resource utilization* are three resource discovery performance criteria most widely used for calibration and evaluation of Grid systems. (1) the *hop number* is the request traversed distance, (2) the query hits is the percentage of successful requests and (3) the *resource utilization* is the percentage of resources that are used by users. Also, please note that the learning algorithm of DLA is $L_{R\text{-}P}$ where $\alpha = \beta = 0.1$.

Also the experiments circumstance includes different Grid size: small, medium and large scale Grids. The small scale Grid is composed of 500 and 1,000 peers. The medium scale Grid is composed of 1,500 and 2,000 peers. Finally the large scale Grid is composed of 2,500 and 3,000 peers.

5.2 Experiment 1: hop number

The distance between two nodes is measured by *hop number* criterion. Each hop is equal to traverse a network link. Figure 2 shows the mean hop number of resource discovery algorithms in six different scales. As shown in Fig. 2, RDDLA algorithm that

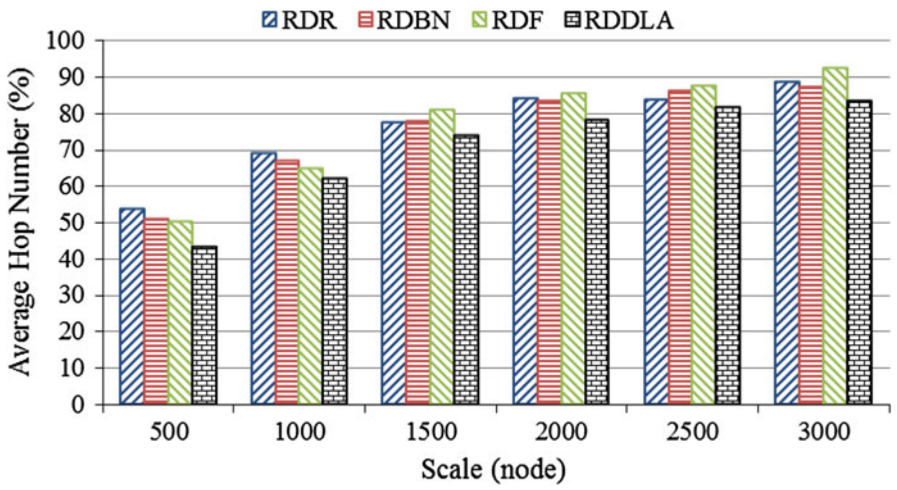

Fig. 2 Average hop number of 30 users with 250 requests in different scales

uses an $L_{R\text{-}P}$ learning algorithm takes the least number of hops in all Grid scales; this observation implies that RDDLA is a scalable and cost-effective resource discovery strategy. Since the resource distribution of small scale Grids is more compressed than medium and large scale Grids, all algorithms locates the requested resources easier. Also, RDDLA and RDF algorithms take the lowest and largest number of hops to discover requests, respectively.

RDR [31] algorithm traverses a stochastic path for each user request and its hop number changes so much by increasing Grid size. RDF [32] algorithm processes the user requests by flooding forwarding of resource requests to adjacent peers of request initiator peer. Hence, the algorithm inevitably meets undesirable peers in the resource discovery procedure. Also, RDBN [6,7] algorithm keeps track of beneficial neighbors of peers; therefore it presents low variation of hop number in the large scale Grids.

5.3 Experiment 2: query hits

Query hits criterion has direct relation with user satisfaction. RDDLA algorithm tries to discover routes that successfully answer high portion of user queries. Figure 3 represents a comparison of average query hits of RDDLA with RDR [31], RDBN [6,7] and RDF [32] algorithms. In small scale Grids, RDDLA outperforms its counterpart algorithms. Also, in medium and large scale Grids RDDLA preserve its performance and also shows superior results compare with other resource discovery algorithms. Moreover, by increasing the Grid size the hit ratio of RDDLA deteriorates, because of not scaling the resource density in terms of Grid scale. In small scale Grids, the resource distribution is compact and peers have low connectivity degree. So, each automaton associated with each GIS has smaller action set and consequently has larger action probabilities than medium and large scale Grids. Furthermore, automaton with small action set has fast convergence speed into optimal action rather than automaton with large action set.

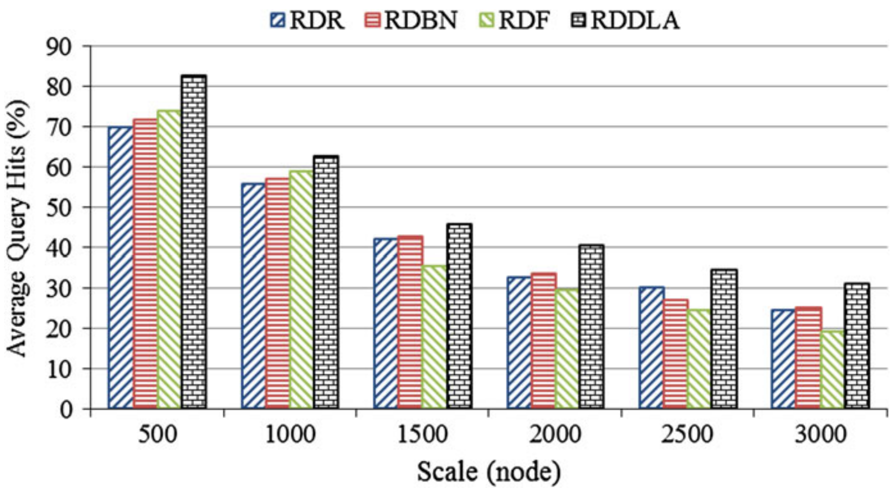

Fig. 3 Average query hits of 30 users with 250 requests in different scales

In small scale Grids, RDF algorithm has larger query hits than RDBN and RDR algorithms. This algorithm always tries to disseminate the resource request from its adjacent neighbors. By increasing the Grid size, RDBN performs better than RDR and RDF. The algorithm could records the number of requests answered by each node and uses this information for processing an incoming query. Finally, the RDR forwarding strategy has the advantage of storing no record history about peers, but also it is the least efficient one.

5.4 Experiment 3: resource utilization

Resource sharing is one of the most important goals in Grid evolution. A key characteristic of resource discovery algorithm is to propagate resource query among various Grid peers and assigns request in diverse set of Grid peers. Figure 4 summarizes the average resource utilization for different resource discovery algorithms. RDDLA algorithm shows superior results in all scales. The best result of this algorithm is 32 % which happens in a Grid with 500 peers. The RDDLA assists $L_{R\text{-}P\text{-}}$ learning strategy to DLA which applies the reinforcement signal equally from originator node to supplier node. Also, RDDLA could utilize various paths to satisfy different user requests in small scale Grids. By growing the Grid size, RDDLA couldn't learns different routes and tries to concentrates on close resources. So, the resource utilization is degraded compares to small scale Grids.

From Fig. 4, the RDBN algorithm suppress RDR algorithm in small and medium size Grids but in large scale Grids RDR algorithm outperforms RDBN algorithm. Moreover, since RDF takes some worthless hops, its resource utilization will diminish. Also in large scale Grids, there are plenty of choices for random forwarding a resource request, thus the algorithm shows better results than RDBN algorithm. RDF algorithm has the lowest resource utilization in all Grid scales.

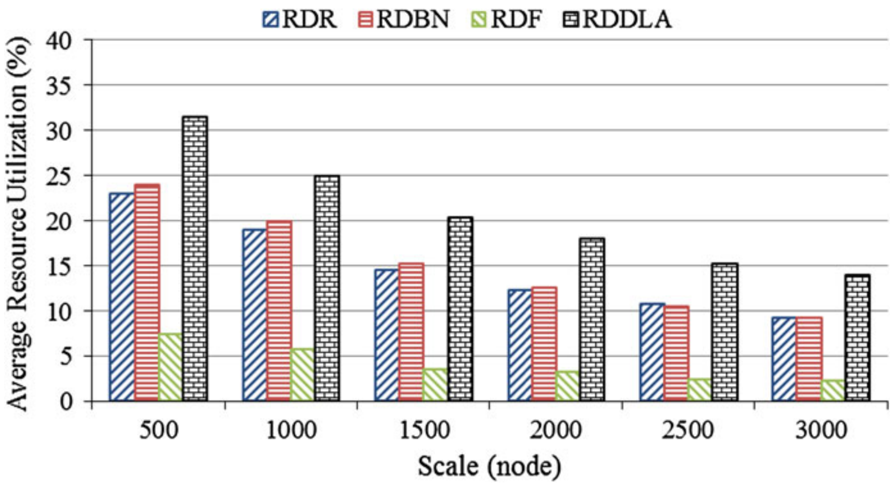

Fig. 4 Average resource utilization of 30 users with 250 requests in different scales

6 Conclusion

This paper develops a distributed learning automata (DLA) approach to Grid resource discovery. Here, a multi-layer architecture for the relationship between the Grid entities and DLA is constructed. First of all, the resource layer at the bottom starts with various available Grid resources and then the Virtual Organization (VO) layer enforces local policies to all member resources. So the Grid Information System (GIS) layer stores and organizes the information of the VO member resources and the DLA layer performs Grid resource discovery. Finally, the user layer at the top provides an engine to communicate with Grid VOs and initiates resource requests.

In this paper resource discovery by DLA (RDDLA) approach is compared with three other resource discovery heuristics. Analysis and experiments indicate that the DLA approach enables the RDDLA to adaptively cope with Grid changes. Furthermore, RDDLA has the best performance in small, medium and large Grid scales in term of hop number, query hits and resource utilization. Especially in small scale Grids while the action sets of DLA are small, the proposed algorithm is more accurate than large scale Grids.

There are lots of improvements which may enhance the performance of RDDLA. As long as the path that is generated by LA with fixed action set may contain loops, using variable action set LA will overcome this weakness and will produce a loop-free path. Also, designing an adaptive learning algorithm that applies the reinforcement signal adaptively from originator node to supplier node may improve the RDDLA learning capabilities. Finally, conducting new experiments with different resource distribution will reveal different tradeoffs between resource distribution and performance of RDDLA algorithm.

Acknowledgments This work was supported partly by the CyberSpace Research Institute (CSRI) of Iran Grant, under Contract Number T/19258/500.

References

1. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid. Int J Supercomput Appl 15(3):1–25
2. Krauter K, Buyya R, Maheswaran M (2002) A taxonomy and survey of grid resource management systems for distributed computing. Softw Pract Exp 32(2):135–164
3. Foster I, Kesselman C, Nick JM, Tuecke S (2002) Grid services for distributed system integration. Computer 35(6):37–46
4. Karaoglanoglou K, Karatza H (2011) Resource discovery in a Grid system: directing requests to trustworthy virtual organizations based on global trust values. J Syst Softw 84(3):465–478
5. Czajkowski K, Fitzgerald S, Foster I, Kesselman C (2001) Grid information services for distributed resource sharing. In: Proceedings of the 10th IEEE international symposium on high performance distributed computing, 2001, pp 181–194
6. Iamnitchi A, Foster I (2001) On fully decentralized resource discovery in Grid environments. In: Lee C (ed) Grid computing—GRID, vol 2242, Springer, Berlin/Heidelberg, pp 51–62
7. Iamnitchi A, Foster I, Nurmi DC (2003) A peer-to-peer approach to resource location in grid environments. In: Internaional series in operaions research and management science, vol 64, pp 413–430
8. Mastroianni C, Talia D, Verta O (Oct. 2008) Designing an information system for Grids: comparing hierarchical, decentralized P2P and super-peer models. Parallel Comput 34(10):593–611
9. Narendra KS, Thathachar M (1974) Learning automata: a survey. IEEE Trans Syst Man Cybern 4:323–334
10. Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice-Hall Inc., Englewood Cliffs
11. Beigy H, Meybodi MR (2011) Learning automata based dynamic guard channel algorithms. Comput Electr Eng 37(4):601–613
12. Mohamadi H, Ismail ASBH, Salleh S (2013) A learning automata-based algorithm for solving coverage problem in directional sensor networks. Computing 95(1):1–24
13. Esnaashari M, Meybodi MR (2012) Deployment of a mobile wireless sensor network with k-coverage constraint: a cellular learning automata approach. Wireless Netw, pp 1–24
14. Hashemi AB, Meybodi MR (Jan. 2011) A note on the learning automata based algorithms for adaptive parameter selection in PSO. Appl Soft Comput 11(1):689–705
15. Hasanzadeh M, Meybodi MR, Ebadzadeh MM (2013) Adaptive cooperative particle swarm optimizer. Appl Intell
16. Beigy H, Meybodi MR (2006) Utilizing distributed learning automata to solve stochastic shortest path problems. Int J Uncertain Fuzziness Knowl Based Syst 14(5):591
17. Akbari Torkestani J, Meybodi MR (2010) An intelligent backbone formation algorithm for wireless ad hoc networks based on distributed learning automata. Comput Netw 54(5):826–843
18. Forsati R, Meybodi MR (2010) Effective page recommendation algorithms based on distributed learning automata and weighted association rules. Expert Syst Appl 37(2):1316–1330
19. Kakali VL, Sarigiannidis PG, Papadimitriou GI, Pomportsis AS (2011) A novel adaptive framework for wireless push systems based on distributed learning automata. Wireless Personal Commun 57(4):591–606
20. Soleimani-Pouri M, Rezvanian A, Meybodi MR (2012) Solving maximum clique problem in stochastic graphs using learning automata. In: Proceedings of 4th international conference on computational aspects of social networks (CASoN), pp 115–119
21. Noghabi HB, Ismail AS, Ahmed AA, Khodaei M (2012) Opimized query forwarding for resource discovery in unstructured peer-to-peer grids. Cybern Syst 43(8):687–703
22. Campos J, Esteva M, López-Sánchez M, Morales J, Salamó M (2011) Organisational adaptation of multi-agent systems in a peer-to-peer scenario. Computing 91(2):169–215
23. Deng Y, Wang F, Ciura A (2009) Ant colony optimization inspired resource discovery in P2P Grid systems. J Supercomput 49(1):4–21
24. Beverly Yang B, Garcia-Molina H (2003) Designing a super-peer network. In: Proceedings of the 19th international conference on data engineering 2003, pp 49–60
25. Akbari Torkestani J (2012) A new approach to the job scheduling problem in computational grids. Cluster Comput 15(3):201–210
26. Jahanshahi M, Dehghan M, Meybodi MR (2013) LAMR: learning automata based multicast routing protocol for multi-channel multi-radio wireless mesh networks. Appl Intell 38(1):58–77

27. Mora-Gutiérrez RA, Ramírez-Rodríguez J, Rincón-García EA, Ponsich A, Herrera O (2012) An optimization algorithm inspired by social creativity systems. Computing 94(11):887–914
28. Hasanzadeh M, Meybodi MR, Shiry S (2011) Improving learning automata based particle swarm: an optimization algorithm. In: Proceedings of the 12th IEEE international symposium on computational intelligence and informatics, Budapest
29. Hasanzadeh M, Meybodi MR, Ebadzadeh MM (2012) A robust heuristic algorithm for cooperative particle swarm optimizer: a learning automata approach. In: Proceedings of the 20th Iranian conference on electrical engineering (ICEE), pp 656–661
30. Buyya R, Murshed M (2002) Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurr Comput Practice Exp 14(13–15):1175–1220
31. Jeanvoine E, Morin C (2008) RW-OGS: an optimized randomwalk protocol for resource discovery in large scale dynamic Grids. In: Proceedings of the 9th IEEE/ACM international conference on Grid computing. Washington, DC, USA, pp 168–175
32. Dimakopoulos VV, Pitoura E (2006) On the performance of flooding-based resource discovery. IEEE Trans Parallel Distrib Syst 17(11):1242–1252