International Journal of General Systems, Vol. 35, No. 3, June 2006, 347–360

Taylor & Francis
Taylor & Francis Group

An adaptive social network-inspired approach to resource discovery for the complex grid systems

L. GAO†, Y. DING†‡* and H. YING§

†College of Information Sciences and Technology, Donghua University, Shanghai 201620, PR China
‡Engineering Research Center of Digitized Textile and Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, PR China
§Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA

(Received 26 July 2005; in final form 19 November 2005)

This paper applies the principles and concepts in social networks to designing a decentralized, survivable and adaptive resource discovery approach in complex grid systems. The simulation results show that our approach can: (i) form relationship among clusters and significantly improve the discovery performance; (ii) adapt well to different resource distributions and user request patterns; (iii) survive from the changes of dynamic environments, including variable-biased user requests and agent amounts as well as partial failure of the agents. Our approach is not only a beneficial experience on dynamic resource discovery of complex grid systems, but also a further attempt to exploit one type of complex systems-inspired approach to build useful services in another type of complex systems.

Keywords: Complex systems; Dynamic resource discovery; Adaptive; Social networks; Ecological networks

1. Introduction

A grid (Foster *et al.* 2002), defined as a coordinated resource-sharing and problem-solving environment in dynamic, multi-institutional virtual organizations (VOs), is a multifaceted system with many components and innovative features. Next-generation grid systems are heading for globally collaborative, service-oriented and live information systems (De Roure *et al.* 2003; Cannataro and Talia 2004). The next-generation grid systems have the hallmarks of complex systems (Ottino 2004), namely, adaptation, self-organization and emergence: no one designed the whole grid or the metabolic-like processes within users and resources. For example, by focusing on one instance of this kind of grid systems, namely file-sharing systems, and examining user behaviours, a self-organizing "small world" pattern can be discovered (Iamnitchi *et al.* 2004).

In such a complex system, resource discovery is a critical activity: given a description of desirable resources, a discovery mechanism can return a set of (contact addresses of) resources that match the description in a dynamic grid environment. Resource discovery is challenging

*Corresponding author. Email: ysding@dhu.edu.cn

International Journal of General Systems
ISSN 0308-1079 print/ISSN 1563-5104 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/03081070600661069

owing to: (i) the potentially myriad resources and users; (ii) highly heterogeneous resource types (e.g. computer power, network bandwidth and online instruments) and highly variable resource attributes (e.g. central processing unit (CPU) load changing); (iii) highly spontaneous sharing characteristics (e.g. users and resources join and leave the grid frequently); and (iv) unbalanced resource distribution and variable user request patterns. So, it is destined that the discovery activity is lack of global centralized authority because aggregation of multiple VOs does not naturally support a central point of control. More importantly, the discovery mechanism needs to exhibit autonomous, flexible behaviours for surviving and adapting to uncertain environment factors (such as different resource distribution and variable user request patterns) (Cannataro and Talia 2004).

However, traditional resource discovery services such as directory services have not addressed all of the above challenges of dynamic resource discovery. For example, the Globus's monitoring and discovery service (MDS) (Czajkowski 2001) is just a centralized service that realizes the tree-like metadata management. Therefore, it is valuable to seek discovery approaches in other fields.

Social networks, another type of vast and complex systems, are a natural way for people to go about seeking information or resources. There is evidence that search depending on acquaintances is remarkably effective in large social networks (Milgram 1967). This reveals not only that short paths exist (among people Watts and Strogatz 1998; Strogatz 2001) but also that ordinary people can find these short paths (Kleinberg 2000). Also, some researches have explained social network searchability (Watts *et al.* 2002). These studies demonstrate that social relationships among individuals can provide a fully decentralized, naturally adaptive, survivable search approach. As such, we study an approach that places the intelligence on the grid entities, enabling the users to locate desirable resources based on social network-like collaborations. Agent-oriented computing (Jennings 2001) provides a big possibility of implementation for this solution. Our previous work (Gao *et al.* 2004) has viewed the grid as a number of interacting software agents and applied some key mechanisms of natural ecosystems to build a novel grid middleware system named ecological network-based grid middleware (ENGM) for solving some issues that complex grid systems face. The work reported in this paper presents further results.

The current paper gives out our further attempts on how to impose the ability of solving complexity deriving from a type of natural complex systems on complex grid systems. We originally apply the key mechanisms and properties of social networks to design a resource discovery approach. We have developed a simulator and modelled some relevant scenarios to evaluate our approach. The results demonstrate that, via our solution, the survivability and adaptation to dynamic grid environments can emerge from autonomous agents. The rest of the paper is organized as follows. The social network-inspired discovery approach, including its framework, key models, and algorithms, is demonstrated in details in Section 2. A simulation implementation and the result analysis are presented in Section 3. Finally, we conclude our research efforts in Section 4.

2. A social network-inspired discovery approach

This section first provides a brief overview of the ENGM system (Gao *et al.* 2004), on which a basic discovery framework for the complex grid systems is set up. Then, this section proposes a resource discovery approach inspired from observation on social networks.

2.1 The ecological network-based grid middleware

We presented a three-layered architecture of the ENGM system as shown in figure 1.

(1) *Heterogeneous and distributed resources* consist of different types of resources distributed in grids. An ENGM platform can run on a distributed system built in a network node.
(2) *ENGM* provides the services to support a common set of applications in grid environments. It is made up by ENGM functional modules, ENGM core services, grid agent survivable environment, emergent grid common service and grid pluggable developing kits. (a) ENGM functional modules layer deals with the management of networks and systems; (b) ENGM core services layer provides a set of general-purpose runtime services that are frequently used by agents, such as community niche sensing service; (c) Grid agent survivable environment is runtime environment for deploying and executing agents; (d) Emergent grid common services layer is kernel of middleware and responsible for resource allocation, information service and so on. These common services are emerged from the interactions among autonomous agents and their environment. (e) Grid pluggable developing kits layer provides pluggable toolkits for developing environment, containing low-level function developing, agent creation and so on.
(3) *Grid applications for virtual organizations* use developing kits and organize certain agents and common services automatically for special purpose applications.

The discussion on design philosophy, layer analysis, functional merits and message-based communication of the ENGM system is beyond the scope of this paper. Readers are referred to work by Gao *et al.* (2004) and Gao and Ding (2005).

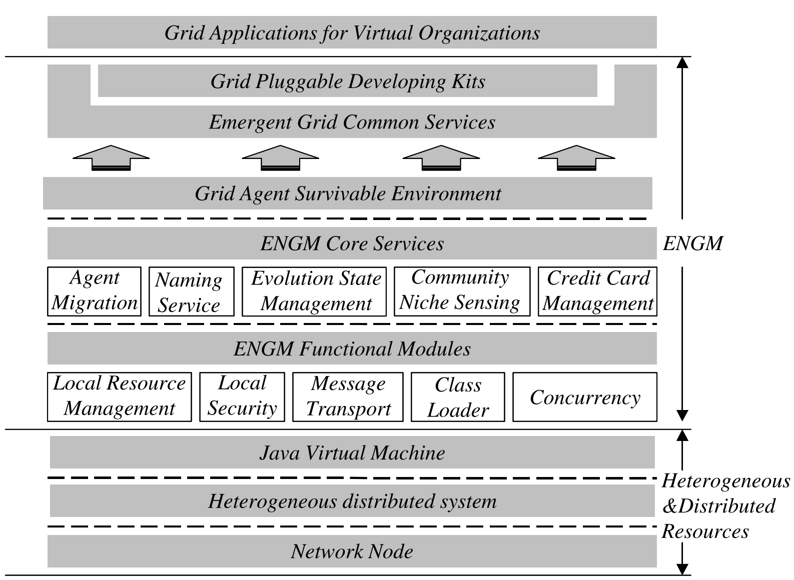

Figure 1. The architecture of ENGM.

2.2 A wide-area resource discovery framework based on ENGM system

Every grid resource supplier maybe has one or more servers that store and provide the access to local resources. We regard these servers as *nodes*. An ENGM platform-based wide-area resource discovery framework is set-up on these nodes. Furthermore, some nodes can form a *community niche* (Niche in ecology is defined as the site where organisms of a species can live, and the function performed by the species. Here, a community niche refers to a logically defined area where agents in a community can learn.) An agent may sense which agents are in the niche, what services they perform and which resources it can access. Here, a niche can be regarded as a VO.

Agents in this framework fall into two categories: grid user agents (GUAs) and grid service agents (GSAs). A GUA represents a kind of user tasks. GSAs are used to comprise the main components of ENGM, such as grid information service agent (GISA). An agent is represented on its unit function, which is defined as a metadata structure. The attributes in the metadata of an agent include *agentID* (its global unique identifier), *agentAddress* (its location), *serviceType* (its service type), *serviceDescription* (service description information), and *relationshipDescription* (the information about its acquaintances, the agents that known by this agent are called *acquaintances*). Besides basic information of its acquaintances (*agentID*, *agentAddress*, *serviceDescription* and *relationshipDescription* of the acquaintances), *relationshipDescription* of an agent still consists *trustCredit* (indication of reliability to acquaintance) and *collaborationRecord* (collaboration history records with acquaintance).

At the beginning of the discovery, a GUA makes user's search instructions into request messages and sends them to a GISA in the local niche. The GISA responds with the matched resource descriptions if it has them, otherwise it forwards the requests to another GISA outside the niche until the request hit returns or request time exhausts. This process is shown in figure 2. There is only one GISA in a niche. It is not feasible for a GISA to know all GISAs in a wide-area grid. If GISA *A* knows a subset of all the GISAs, then it can communicate with GISA *B* among them, and *B* delivers the request to one of GISAs it knows and so on. Broadcast-type search strategies will lead to high bandwidth cost, scalability problem and congestion constraints (Adamic *et al.* 2001). Therefore, a request is only passed onto one GISA at each step in our framework.

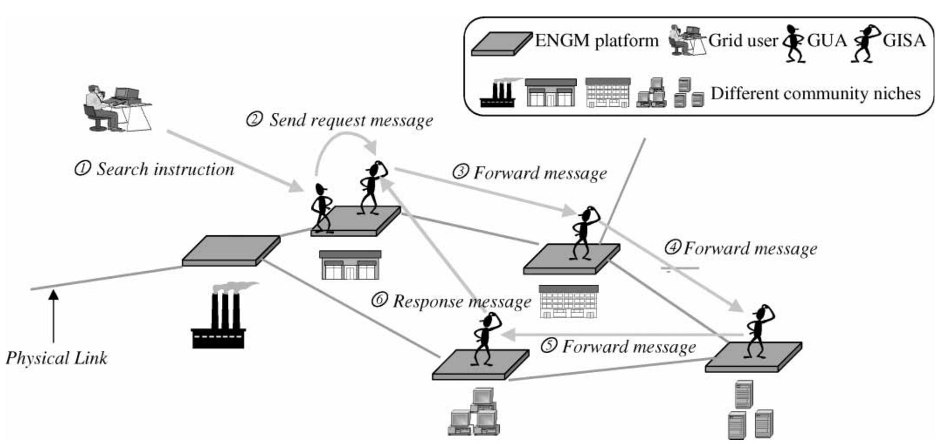

Figure 2. Delineation of a simple discovery process.

2.3 Proposed resource discovery approach

2.3.1 Properties of social networks. People often ask their acquaintances for interesting items to obtain advice, which shows a kind of collaboration in the real world. Some key properties of the collaboration draw our attentions.

2.3.1.1 Interpersonal communication

This performs well for propagating and seeking valuable information in a social environment. Our approach is expected to form an interpersonal communication-based social network among discovery agents (GISAs) where a self-organizing, adaptive and survivable discovery emerges. A challenge is how to decide the right person to ask.

2.3.1.2 Autonomy of the individuals

Discovery entities in grid environments are necessary to implement autonomous decision-making ability as individuals do in social networks. However, entities will not act to achieve their group interests if there is no coercion to make them act so. Here, accountability, a concept in human society, is introduced into the design of our approach. A GISA must be responsible for its activities such as service provision and resource utilization.

2.3.1.3 Trust

This is an important social concept and present in all human interactions. If a person wants to obtain advice, he will not ask all acquaintances but the reliable ones. If a person finds one of his acquaintances is getting along with him, he will trust the acquaintance more. Otherwise, trust less. A trust model is introduced to the discovery collaboration among GISAs.

2.3.1.4 Cluster

Individuals will cluster the social world through the long-term collaborations. In our approach, GISAs with related resources will be located nearby one another in the relationship network, facilitating resource discovery, which results into the clustering of relative GISAs.

2.3.2 A social network-inspired three-phase discovery approach. Based on the discovery framework and observation on social networks, a three-phase discovery approach is proposed: GISA relationship construction, request processing strategy and trust-based reconstruction of relationship.

(1) GISA relationship construction is responsible for collecting and updating information on currently participating GISAs and forming a relationship network. A GISA joins the grid by contacting a member GISA. Contact addresses of a member GISA can be learned via information sensing mechanism integrated in the ENGM platform (Gao *et al.* 2004). Once a GISA is found, a relationship is established. The GISA adds the new information to its *relationshipDescription* attribute list. A GISA contacted by a joining member responds with its GISA *agentAddress* attribute. Then, the joining GISA can access its information such as its service type and decide to join the existing niche or not. Thus, a set of relationships is built to form a social network over all GISAs. GISAs can update the changed relationships of their acquaintances.

(2) Request processing strategy performs the search itself. A request message contains the message ID, message type, information on a message originator and

message forwarders, a set of parameters to specify attributes of the target resources and a set of weights to describe the importance of each resource attribute. When a GISA receives a request message, it first examines the message ID to check whether it has seen the message before. If it has, it then discards the message. Otherwise, it creates a new entry in a message-processing list. After completing the list for the newly received request, it evaluates the request by *resource matching model* and *request forwarding model*. Then, the GISA decides to respond the request with its local resources, forward the request to its acquaintances, or reject the request. It can both respond and forward the request. Resource matching model and request forwarding model are given out as follows. The resources held by a GISA and the requested resources are both modeled as an attribute vector.

(a) *Resource matching model*. Given a request vector$R = \langle r_1, r_2, \ldots, r_n\rangle$, a weight vector $W = \langle w_1, w_2, \ldots, w_n\rangle$ (where $\sum_{t=1}^{n} w_t = 1$) that indicates the importance degree of each request attribute about R, and a grid resource vector $G = \langle g_1, g_2, \ldots, g_n\rangle$, the matching strength between R and G is defined as:

$$\mathrm{MS}_{\mathrm{vector}}(R, W, G) = \sum_{t=1}^{n} \mathrm{MS}_{\mathrm{single}}(r_t, g_t) w_t \tag{1}$$

where

$$\mathrm{MS}_{\mathrm{single}}(r_t, g_t) = \begin{cases} 1, & g_t \text{ satisfies } r_t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

For example, if a single request attribute value r_t represents "more than 128 MB of available memory" and a single resource attribute value g_t stands for "256 MB of available memory", according to equation (2), $\mathrm{MS}_{\mathrm{single}}(r_t, g_t) = 1$. The request originator will specify a threshold $\omega_i (0 \leq \omega_i \leq 1)$ for resource matching. If $\mathrm{MS}_{\mathrm{vector}}(R, W, G) \geq \omega_i$, there is a matching between R and G. For example, for a three-tuple vector, $\{\mathrm{MS}_{\mathrm{single}}(r_1, g_1), \mathrm{MS}_{\mathrm{single}}(r_2, g_2), \mathrm{MS}_{\mathrm{single}}(r_3, g_3)\} = \{1, 1, 0\}$, and $W = \langle 0.5, 0.3, 0.2\rangle$. According to equation (1), $\mathrm{MS}_{\mathrm{vector}}(R, W, G) = \sum_{t=1}^{3} \mathrm{MS}_{\mathrm{single}}(r_t, g_t) w_t = 1 \times 0.5 + 1 \times 0.3 + 0 \times 0.2 = 0.8$. If $\omega_i = 0.8$, there exists a matching. If $\omega_i = 0.9$, there does not. Grid users can set different weight vectors in a request message to emphasize their desired resource attributes.

(b) *Request forwarding model*. When a GISA processes a request message, it not only examines whether its local resource satisfies the request, but also decides which acquaintance to forward by the calculation of *the optimal forwarding strength*. Through *trustCredit* values with which a GISA labels its acquaintances, it relies on some GISAs and mistrusts in other ones. In addition, information about similar requests that user evaluated previously in *collaborationRecord* attribute may help users get desired resources.

The forwarding strength that GISA A_i imposes on its acquaintance A_{i_j} is relative to *trustCredit* and *collaborationRecord*. The optimal matching strength on requests in *collaborationRecord* is defined as $\mathrm{MS}_{\mathrm{opt}}(R, W, R_{i_j}^k) = \max_{k=1}^{m} \mathrm{MS}_{\mathrm{vector}}(R, W, R_{i_j}^k) \cdot \delta_{i_j}^k$, where R and W have the same meaning in resource matching model, $R_{i_j}^k = \langle r_{i_j}^{k_1}, r_{i_j}^{k_2}, \ldots, r_{i_j}^{k_n}\rangle$ is one of m request vectors that A_{i_j} previously has answered and/or forwarded, and $\delta_{i_j}^k$ is a parameter in $[-1/2, 1]$ that indicates the user evaluation on request vector $R_{i_j}^k$. Given a weight η to *trustCredit* and *collaborationRecord*, the forwarding strength that GISA A_i imposes on its acquaintance A_{i_j} is defined as $\mathrm{FS}_{\mathrm{vector}}(R, W, A_i, A_{i_j}) = \eta \cdot \mathrm{trust}_{i,i_j} + (1 - \eta) \cdot \mathrm{MS}_{\mathrm{opt}}$ $(R, W, R^k_{i_j}))$, where trust_{i,i_j} is a number in $[0, 1]$ representing *trustCredit* value that A_i has on A_{i_j}.

The optimal forwarding strength $\mathrm{FS}_{\mathrm{opt}}(R, W, A_i, A_{i_k}) = \max^n_{k=1} \mathrm{FS}_{\mathrm{vector}}(R, W, A_i, A_{i_k})$ can help A_i decide which acquaintance to forward. Before A_i forwards the request message, it will register its information in the message. Furthermore, A_i will specify a threshold θ_i $(0 \leq \theta_i \leq 1$ and usually $\theta_i \leq \omega_i)$. When evaluating the resource request, if $\mathrm{MS}_{\mathrm{vector}}(R, W, G_i) \geq \omega_i$, A_i will respond the request with local resources. Further, if $\mathrm{FS}_{\mathrm{opt}}(R, W, A_i, A_{i_k}) \geq \theta_i$, A_i will forward the request message. A_i will discard the message if it neither responds nor forwards the request.

In our approach, users are only required to decide three parameters: η, ω_i, and θ_i. The effect of η will be studied in Section 3 to help users chose a reasonable one. ω_i and θ_i are introduced into the proposed approach to bring GISAs an accountability mechanism. If a GISA sets an ill-suited value of ω_i or θ_i, it will get a penalty for its unaccountable behaviors that results in dissatisfied discovery. We use time to live (TTL) to describe the time limitation of forwarding request. The GISA that matches a given search request may respond directly back to the search originator as soon as they are available.

(3) Trust-based reconstruction of relationship makes the necessary preparations for a more efficient search. This mechanism on the reconstruction of relationship can contribute to resource discovery. It updates and strengthens the relationship network by establishing and changing the *trustCredit* values among the GISAs.

On receiving a request hit, the request originator returns a defray message including a collaboration record and a credit that stands for the user evaluation of the request hit. If the relationship of a GISA does not contain the corresponding collaboration record, the collaboration record including initial user evaluation, is added to *relationshipDescription* attribute. A credit could be a reward or a penalty, which indicates that the degree of a user's preference to the received request hit. This message is propagated through the same path where the discovery request has been originally forwarded. When an intermediate GISA on the path receives message, it adjusts the *trustCredit* value of the relationship that has been used to forward the original discovery request. *trustCredit* value is increased for a reward (i.e. high degree of the request originator's preference), and is decreased for a penalty (i.e. low degree of the request originator's preference). Given a credit $\gamma(-1/2 \leq \gamma \leq 1)$ that contained in a defray message, GISA A_i updates the *trustCredit* value of A_{i_j} using the formula:

$$\mathrm{trust}^+_{i,i_j} = \begin{cases} \mathrm{trust}^-_{i,i_j}(1 - \gamma^2) + \gamma^2, & \gamma \geq 0 \\ \mathrm{trust}i, i_j^- \left(2 - \frac{1}{1+\gamma}\right), & \gamma < 0 \end{cases},$$

where trust^-_{i,i_j} is a number in $[0, 1]$ that represents *trustCredit* value that A_i imposes on A_{i_j} before updating. Correspondingly, trust^+_{i,i_j} is the *trustCredit* value after updating. We have chosen the above formula with the purpose of remarking ratings rise slowly and fall quickly.

3. Simulation studies

3.1 Simulation set-up

To evaluate our approach, a simulator is developed on ENGM platform, which supports pluggable functions and provides a generic easy-to-use programming Application Program Interface (API).

3.1.1 Initial relationship network topology. We suppose that at each time there is only a request message sent by a GISA and the same message can be sent only once by the request originator during the entire simulation. Define that a cycle starts from a request message sent by a GISA and ends when all the relevant messages disappear in the system, and 100 cycles is a generation in the simulation.

In our simulated network, the nodes are GISAs and the edges connect pairs of GISAs that know each other. The relationship network defined by the GISA relationship construction can strongly affect discovery performance. Two characteristics, we think, must be well reflected in topology generation: (a) Internet is the carrier of grids, with the increasingly growing scale; (b) new nodes will preferentially connect to those nodes with more connections. To follow them, we adopt Barabási and Albert's method (Barabási and Albert 1999) for topology generating. The generated topology follows power laws that can provide the simulation of grid environments with basic referential standards. It should be pointed out emphatically that the statistic results from domain level and router level also follow the power law rule, and GISAs just play their roles at these two levels.

3.1.2 Resource distribution. In the simulation, the storage space in GISAs' attributes is supposed to be unlimited. Two thresholds involved, θ_i and ω_i, are respectively set to be 0.2 and 0.8. Actually, each GISA can make decision for deciding different values of θ_i and ω_i. Here, our aim is to evaluate the performance of the proposed approach, so we make simplifications. When there are the occasions of different resource vectors that meet the same request, GISA will randomly choose one.

We make a set of common resources (contains 20,000 different resource vectors) and a set of new-type resources (contains 2000 different resource vectors that are completely different from common resources). We experiment on two resource distribution strategies. (a) Balanced distribution strategy: initially each GISA provides 3–5 resource vectors, which are randomly picked out (can be repeatedly picked out) from the common resource set. (b) Unbalanced distribution strategy: a few number of GISAs (about 8%) provide most (about 80% of the total) of the resource vectors, while the other 92% of GISAs share 20% of the resources. Here, the resource vectors come from common resource set. In both strategies, local resource vectors held by GISAs are replaced randomly with 100 new-type resource vectors per generation.

3.1.3 User requests and user evaluation. Requests are initiated at a fixed percentage of randomly selected GISAs and contain resource vectors. The resource attributes of each request have the same weights. In network environments, the request distribution complies with Zipf distributions (Zipf 1949). For simplification, we suppose that users are interested in a particular subset of the request vectors. Two user request patterns are studied: (a) unbiased user request scenario (asking for the total resource request vectors randomly) and (b) biased user request scenario (using the probability 0.8 to ask for about 5% special request vectors of the total resource request vectors available in the simulation network). In addition, we randomly assign a fixed threshold E_i (a random value between 0.6 and 1) to each GISA for indicating the satisfaction level of a request originator. On receiving a request hit, the request originator will examine the matching degree of responded resources. If the matched result is greater than or equal to E_i, the request originator returns the defray message with a credit $\gamma = 0.1$ to the resource responder; otherwise, the credit γ is equal to -0.1. The GISAs in the credit-propagated chain will be given the same amount of credit γ.

3.2 Simulation results

The simulation evaluates the survivability and adaptability of the approach in the following three aspects: (a) resource distributions; (b) user request patterns; and (c) the number and reliability of GISAs. Also, we evaluate discovery performance distinctions affected by different values of η (i.e. $\eta = 0$, 0.25, 0.5, 0.75, 1). Considering the randomness in the simulations, we repeat the experiments multiple times. The results given out are the averaged values of measurements.

We first experiment on discovery performance (as measured in number of hops that stands for node amount) with different η values in different resource distributions (balanced and unbalanced) and user request scenarios (biased and unbiased), as shown in figures 3 and 4. We also give out the performance measurement of a random forwarding approach used as a comparison with our approach in figure 3. Within unbalanced resource distribution and biased request scenario, we further evaluate the effect of the dynamic user requests on the discovery. The user requests (i.e. the biased user request set) are gradually changed in the following way: first, to set ten generations as an interval, then, to change the biased user request set with the ratios as $\xi = 0$ (no changes), 0.002, 0.02, 0.2 and 1 (the user requests completely change) at the beginning of each interval. The results are shown in figure 5(a).

To evaluate the effect of the number of GISAs on discovery approach, the simulation is conducted with the numbers from 1000 to 10,000 in different conditions. Figure 5(b)

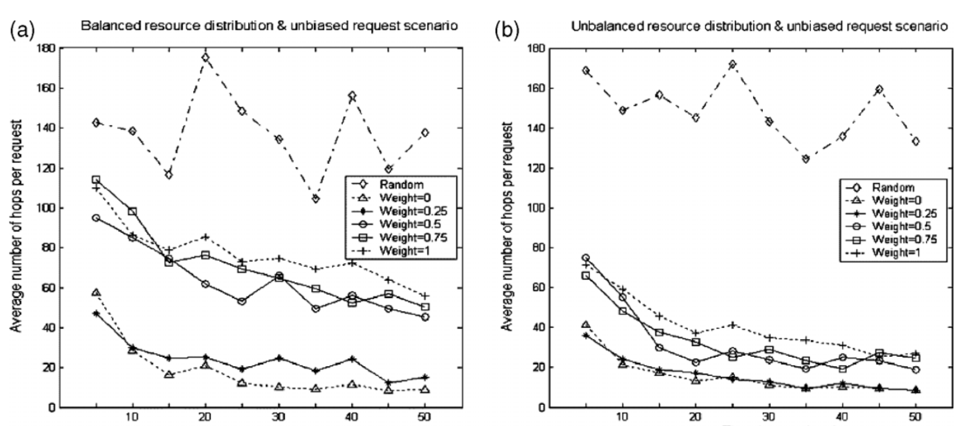

Figure 3. Average number of hops per request with different resource distributions in unbiased user request scenario. (Left: balanced. Right: unbalanced). The number of GISA is 5000. The value of TTL is 200.

demonstrates such a result in unbalanced environment for biased user request pattern in 50 generations. We experiment on the adaptability of the approach in the condition where GISAs are likely to be unavailable. We adopt the same simulation environment as figure 5(b) to conduct a comparison. At each generation, we chose 1% of GISAs randomly in the simulation and set them unavailable for discovery (they will be recovered available in the next simulation generation). A measure of the adaptability to unreliable GISAs is measured as

$$D = \frac{N_{\text{hop}}^{\text{unreliable}} - N_{\text{hop}}^{\text{reliable}}}{N_{\text{hop}}^{\text{reliable}}}$$

where $N_{\text{hop}}^{\text{unreliable}}$and $N_{\text{hop}}^{\text{reliable}}$ are respectively, the average number of hops per request for a certain number of GISAs in a static (all GISA are reliable) and dynamic scenario. The average result equals to

$$\frac{1}{N}\sum_{i=1}^{N} D$$

where $N = 10$ represents for ten experiment points on the number of GISAs with each value of η for all the numbers of GISAs are, respectively, 0.0373, 0.039, 0.0403, 0.0302 and 0.0429.

3.3 Analysis of simulation results

3.3.1 Clustering of discovery relationships. As shown in the figure 3, the random forwarding has the lowest efficiency though it is low-cost (stores no discovery information in GISAs). Its average number of hops in 50 generations with balanced distribution strategy is 137.26 hops, which is about 7.5 times of that as $\eta = 0$ and 1.8 times of that as $\eta = 1$ in the same experiment environment. With unbalanced distribution strategy, the average number of hops is 148.75, which is about 9.5 times of that as $\eta = 0$ and 3.7 times of that as $\eta = 1$ in the same experiment environment.

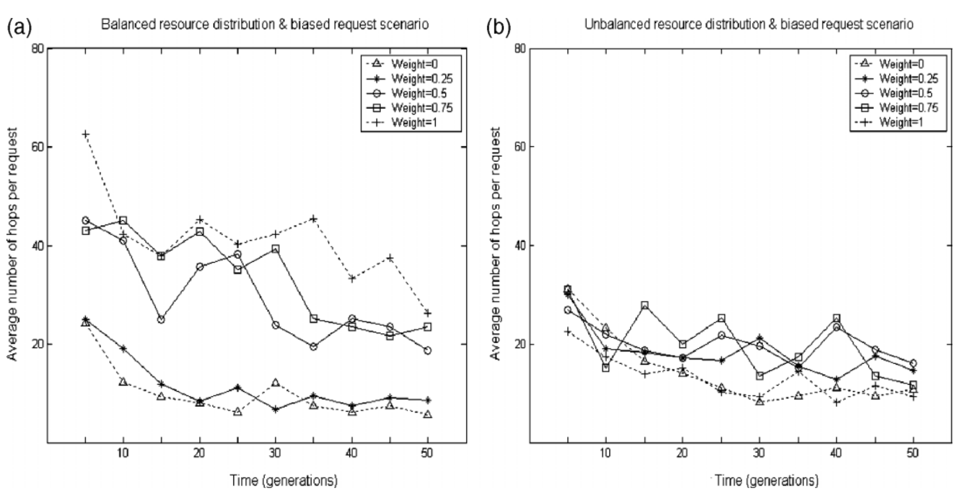

Figure 4. Average number of hops per request as a function of simulation time within biased user request scenario in two environments (Left: balanced. Right: unbalanced). The number of GISA is 5000. TTL value is set to 100.

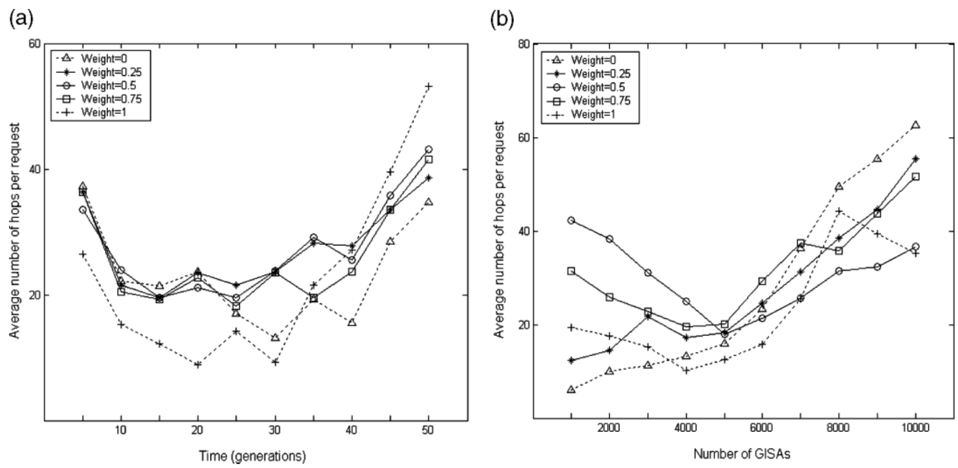

Figure 5. (a) The performance impacted by the varying user requests in unbalanced environment for biased user request patterns. GISA amount is 5000 and TTL value is 100. As the time is 10, 20, 30, 40 and 50 generations, the variable probabilities of user requests are 0, 0.002, 0.02, 0.2 and 1, respectively. (b) The performance with the increase of the number of GISAs in unbalanced environment for biased user request patterns. TTL value is 100.

The approach can improve the discovery performance evidently and adaptively. At the beginning of simulation, relationships are random, and discovery performs poorly. Many hops need to be visited to hit the target GISA. As some simulation cycles elapse, GISAs gradually obtain many relationships similar to themselves, leading to improved performance in discovery process. We find that such improvement results from the clustering of vector-matched GISAs. With enough hops, GISAs are likely to meet some relative resource clusters and enter the clusters to find the required resources. The clusters have not formed in the process of random forwarding simulation, and the discovery will go on aimlessly until it meets the matched GISA. The comparison results have proved that our approach can form clusters and improve the performance.

3.3.2 Effects of resource distribution. Generally speaking, the discovery has better performance in a highly unbalanced environment than in a balanced environment, as shown in figures 3 and 4. When a few number of GISAs hold most of the resources and so have rather more relationships, it is easier for them to form clusters.

Relative to the other η, the condition with $\eta = 1$ (discover totally according to *trustCredit* value) takes minimum overhead costs, but almost the lowest discovery efficiency. While $\eta = 0$ (discover totally according to *collaborationRecord*), it is with the highest discovery efficiency, but not greater consumption. In unbalanced environments, the efficiency of the *trustCredit*-based discoveries (with minor value of η) is much better than that in balanced environments.

3.3.3 Influence of user request scenario. Comparing figure 3 with figure 4, we can see that as a whole, the performance of our approach within biased request scenario is better than within unbiased request scenario. Especially, the performance with $\eta = 1$ has evidently been improved, which means that discovery fully according to *trustCredit* value can better support the special user request scenario. It shows very good efficiency in the unbalanced environments, where its response latency is almost half of that in the balanced environments. Similarly, the performance in the unbalanced environments has been improved comparing to that in the balanced environments as η equals 0.75, 0.5 and 0.25. Relative to the unbalanced environments, when $\eta = 0$, the performance is better in the balanced environments.

When changing biased user requests in an unbalanced resource environment, we find the performance of our approach first becomes good, then gradually turns bad, as shown in figure 5(a), The explanation is: originally, there is enough randomness in the relationship network to greatly improve the performance and specific requests will strengthen the clusters. However, as the GISAs acquire more relationships similar to some specific user requests, randomness is lost. As the requests are changed, the performance degrades greatly (As the time goes, the discovery will acquire good adaptability again.) As shown in figure 5(a), the discovery with $\eta = 1$ performs best when there are little or no changes in user requests and performs worse as the requests become more dynamic. During the whole process of simulation, the discovery with $\eta = 0$ performs best.

3.3.4 Influence of the number and reliability of GISAs. For all network sizes in our experiments, the discovery has obtained good performance, as shown in figure 5(b). The discovery with $\eta = 0$ performs predictably because its average number of hops per request increases with raise of the number of GISAs. While discoveries with other values of η prove to be slightly unpredictable in terms of performance. The experiment on adaptability to unreliable GISAs of relationship network shows the performance of our approach in the dynamic scenario is worse than that in the static scenario. However, the difference between the static and dynamic scenarios appears relatively small, which suggests that our approach could be survivable in the environment with unreliable GISAs.

4. Conclusions

We have originally applied the principles and concepts of social networks to the design of a dynamic discovery approach. The simulation results show that our approach forms relationship clusters and significantly improves the discovery performance. With balanced and unbalanced distribution strategies, the minimum average numbers of hops in 50 generations are about 13.3 and 10.5% of that of a random forwarding approach in the corresponding condition, respectively. The results also demonstrate the approach can well adapt to different resource distributions and user request patterns, and survive from the changes of dynamic environments as well as partial failure of GISAs. The performance distinctions of different values of η in all experiments can help users better understand the tradeoffs between overhead costs and performance, and chose the preferable parameters. Our approach is a further attempt to exploit one type of complex systems-inspired approach to build useful services in another type of complex systems.

Acknowledgements

This work was supported in part by the Key Project of the National Nature Science Foundation of China (No. 60534020), the National Nature Science Foundation of China (No. 60474037 and 60004006) and Program for New Century Excellent Talents in University.

References

L.A. Adamic, R.M. Lukose, A.R. Puniyani and B.A. Huberman, "Search in power-law networks", *Phys. Rev. E*, 64(4), pp. 46135–46143, 2001.

A.L. Barabási and R. Albert, "Emergence of scaling in random networks", *Science*, 286, pp. 509–512, 1999.

M. Cannataro and D. Talia, "Semantics and knowledge grids: building the next-generation grid", *IEEE Intell. Syst.*, 19(1), pp. 56–63, 2004.

K. Czajkowski, S. Fitzgerald, I. Foster and K. Kesselman, "Grid information services for distributed resource sharing", *Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing*, Washington DC, USA, 2001, pp. 181–194.

D. De Roure, N.R. Jennings and N.R. Shadbolt, "The evolution of the grid", in *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. Fox and A.J.G. Hey, Eds., Chichester: John Wiley, 2003, pp. 65–100.

I. Foster, C. Kesselman, J.M. Nick and S. Tuecke, "Grid services for distributed system integration", *Computer*, 35(6), pp. 37–46, 2002.

L. Gao and Y.-S. Ding, "A flexible communication solution to support grid service emergence", *Lect. Notes Comput. Sci.*, 3482, pp. 69–78, 2005.

L. Gao, Y.-S. Ding and L.-H. Ren, "A novel ecological network-based computation platform as grid middleware system", *Int. J. Intell. Syst.*, 19(10), pp. 859–884, 2004.

A. Iamnitchi, M. Ripeanu and I. Foster, "Small-world file-sharing communities", *Proceedings IEEE INFOCOM 2004,* Hong Kong, China, 2004.

N.R. Jennings, "An agent-based approach for building complex software systems", *Commun. ACM*, 44(4), pp. 35–41, 2001.

J.M. Kleinberg, "Navigation in a small world", *Nature*, 406, p. 845, 2000.

S. Milgram, "The small world problem", *Psychol. Today*, 2, pp. 60–67, 1967.

J.M. Ottino, "Engineering complex systems", *Nature*, 427, p. 399, 2004.

S.H. Strogatz, "Exploring complex networks", *Nature*, 410, pp. 268–276, 2001.

D.J. Watts and S.H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, 393, pp. 440–442, 1998.

D.J. Watts, P.S. Dodds and M.E.J. Newman, "Identity and search in social networks", *Science*, 296, pp. 1302–1305, 2002.

G.K. Zipf, *Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology*, Cambridge, MA: Addison-Wesley Publishing, 1949.

Lei Gao obtained a BS in electrical engineering from Donghua University, Shanghai, China in 2001. In 2002, he was selected for working towards a PhD degree majoring in control theory and control engineering at the same university. His current research interests include complex grid systems, nature-inspired technologies, mobile communication technologies and multi-agent systems.

Yongsheng Ding is a professor at College of Information Sciences and Technology, Donghua University, Shanghai, China. He obtained BS, MS and PhD degrees in Electrical Engineering from Donghua University, Shanghai, China in 1989, 1994 and 1998, respectively. From 1996 to 1998, he was a visiting scientist at the Biomedical Engineering Center, The University of Texas Medical Branch, Texas, USA. From March 2003 to May 2003, he was a senior visiting scientist at Ecole Nationale Supérieure des Arts et Industries Textiles, Roubaix, France. From February to April 2005, he was a visiting professor at Department of Electrical and Computer Engineering, Wayne State University, Michigan, USA. He has published more than 180 technical papers, two research monograph/advanced textbooks entitled *DNA Computing and Soft Computing* (2002) and *Computational Intelligence* (2004). His scientific interests include computational intelligence, network intelligence, nature-inspired technologies, bio-computing and bio-informatics, intelligent decision-making.

Hao Ying is a professor at Department of Electrical and Computer Engineering, Wayne State University. He was on the faculty of The University of Texas Medical Branch at Galveston between 1992 and 2000. He obtained BS and MS degrees in electrical engineering from Donghua University (formerly China Textile University) in 1982 and 1984, respectively, and a PhD degree in Biomedical Engineering from The University of Alabama at Birmingham in 1990. Professor Ying has published one research monograph/advanced textbook entitled *Fuzzy Control and Modeling: Analytical Foundations and Applications* (IEEE Press, 2000) and 65 peer-reviewed journal papers. He is an Associate Editor for *International Journal of Fuzzy Systems* and is on the editorial board of *International Journal of Approximate Reasoning*. He is an elected member of the *North American Fuzzy Information Processing Society (NAFIPS)* (three-year term beginning in 2005). He was a Guest editor for *Information Sciences*, *International Journal of Intelligent Control and Systems* and *Acta Automatica Sinaca*. He served as Program Chair for *The 2005 NAFIPS Conference* as well as for *The International Joint Conference of NAFIPS Conference, Industrial Fuzzy Control and Intelligent System Conference, and NASA Joint Technology Workshop on Neural Networks and Fuzzy Logic* held in 1994. He served as the Publication Chair for the *2000 IEEE International Conference on Fuzzy Systems* and as a Program Committee Member for many other international conferences.