
ECOLOGICALLY INSPIRED DISTRIBUTED SEARCH IN UNSTRUCTURED PEER-TO-PEER NETWORKS

Li Sa, Yongsheng Ding

***Abstract:** In this paper, we reported an ecosystem inspired algorithm for searching peer-to-peer networks. An agent-based model had been developed to solve the problems caused by the blind flooding-based search, such as inefficient search and traffic, etc. Reinforcement learning and evolution of the agents can change the agents population toward sampling areas of the environment that are close to resources by increasing the density of agents near those sources.*

***Keywords:** Peer-to-Peer Search, Agent-based System, Modeling and Simulation, Reinforcement learning.*

***ACM Classification Keywords:** C.2.6 Internet working*

1 Introduction

A peer-to-peer (P2P) network is distributed systems based on the concept of resource sharing by direct exchange between peer nodes (i.e., nodes having same role and responsibility).. Exchanged resources include content, as in popular P2P files are end systems in the Internet and maintain information about a set of other nodes (called neighbors) in the P2P layer. These nodes form a virtual overlay network on top of the Internet. Each link in a P2P overlay corresponds to a sequence of physical links in the underlying network.

Early search mechanisms primarily used flooding or k-random walk [Hoile, 2002] algorithms. In the flooding approach, each node propagates the query to all its neighbors. On a receipt of a query, the node searches in its local repository. If the object is found, it informs the query originator and further search in that path terminate. If not, the node further forwards the query to all its neighbors.

The flooding method generates a large amount of network traffic. To overcome this problem, random walk algorithms are often used. In Random Walks, the requesting node sends out k query messages to an equal number of randomly chosen neighbors. Each of these messages follows its own path, having intermediate nodes forward it to a randomly chosen neighbor at each step. These queries are also known as walkers. Random Walkers cannot learn anything from its previous successes or failures, displaying high variability in all ranges of requests.

The environments in which P2P applications are deployed exhibit extreme dynamism in structure and load. In order to deal with the scale and dynamism that characterize P2P systems, a paradigm shift is required that includes self-organization, adaptation and resilience as fundamental properties. Complex adaptive systems (CAS) commonly used to explain the behavior of certain ecological and social systems can be the basis of a new programming paradigm for P2P applications. Here we are concerned with the development of an ecosystem-inspired approach to the design of agent systems for searching in Unstructured Peer-to-Peer Networks.

Ecosystems have been a source of inspiration to a number of previous developers of agent systems [Lv, 2002, Moukas, 1997]. Moukas [Moukas, 1997] employs ecosystem inspired ideas in the Amalthaea architecture for information filtering. Paul Marrow and colleagues [Lv, 2002] have advocated an "information ecosystems" approach to support a variety of information management applications.

This paper developed an ecology-based model for managing a number of search agents on the P2P networks that can provide decentralized distributed robust control of agents in the dynamic P2P network environments. This kind of agents does provide a viable means of performing network resource discovery, which makes P2P more practicable. This paper is organized as follows: Sec.2 defines the model and states the attributions of agents; a detailed description of all aspects related to the EIDS algorithm. Sec.3, next to showing the performance of the algorithm in comparison to other approaches, presents experimental results and analysis. Sec.4 provides a discussion of related work on agent based model used for searching in peer-to-peer networks.

2 System Model

This section is divided into two parts. In the first part (Section 2.1) we describe the framework chosen to model the P2P environment. In the second part (Section 2.2) we describe the ecologically inspired search algorithm.

2.1 Environment Description

The model framework involves three concepts: peers, neighbors and search agents. The peers are typically computing devices that can maintain some state, or perform computations. A peer has a set of neighbors and is able to send search agents to its neighbors only.

The factors which are important for simulating P2P environments are the overlay topology. The overlay topology consists of a two-dimensional grid responsible for maintaining the neighborhood connections between the peers in the P2P network. Due to the grid structure, each peer residing in a particular node has a fixed set of eight neighbors. Each grid is conceived to be containing a heterogeneous distribution of peers of the P2P network with one or more distinct resources. We assume that there are 1024 unique resources; each of them can be represented by a 10-bit binary token as the resource information (RI). All agents take with also a 10-bit binary token as their searching information (SI) when they move across the environment. Similarity between a RI and a SI is measured by the number of bits that are identical. That is, $\text{sim}(\text{RI}, \text{SI}) = d - \text{HD}(\text{RI}, \text{SI})$, where HD is the Hamming distance between RI and SI. Zipf's distribution [Zipf, 1935] is chosen to distribute each of the 1024 unique resources in the network. These resources are gathered and consumed by the agent to survive. Each peer may be visited by at most one agent at any time. At any time during the evolution of the model, each agent has a distinct location on the environment, characterized by the peer the agent visits, and a distinct field of view, measured in grids. Each agent has perfect knowledge of resource levels within its field of view; an agent has no knowledge or memory of resource levels outside its field of view. All the agents have a fixed time to live. In addition, an agent has a characteristic metabolic rate for the resource it finds and consumes. Any resources found by an agent can be retained as energy without constraint. If an agent's energy diminishes to zero, the agent dies from starvation.

2.2 The EIDS Algorithm

The ecologically inspired search is a distributed algorithm in which queries are represented as agents. The agents are created at the peer who issued the query and travel over the P2P network in which peers are arranged in a grid-like topology, as in the Swarm simulator [Kaelbling, 1996]. At random times, each agent makes a random number of hops along the P2P network.

The search in our P2P network is initiated from the user peer. The user emanates search agents (message packets) to its neighbors-the packets are thereby forwarded to the surroundings. The method of spreading search agents forms the basis of the algorithm.

Ecosystems usually have several attractive qualities (such as dynamic decentralized control, self regulation, no single point of failure, robustness, and stability) that we require for P2P system. We propose a solution to the problem of controlling the number of agents appropriate for a search which is inspired by large ecosystems (Figure.1):

1. Each kind of resources (Resources could be files in a file-sharing system or CPU cycles in a computational grid.) in P2P system will be associated with energy.
2. Agents finding a resource successfully will collect the energy associated with the resource.
3. Agents consume energy over time to sustain their existence.
4. Agents that exceed the time to live or exhaust their supply of energy die.
5. Abundance of energy can cause a new agent to spawn.

All search operations are controlled by a set of micro-scale rules. The spatial distribution of resources are searched, the energy associated with the resources are gathered and consumed by the agents to survive.

An agent has two dynamic attributes behavioral attributes:

- The current energy level (EL): The current energy level of the agent. If this falls below zero the agent will die.
- The agent's age (AA): Measured in number of hops.

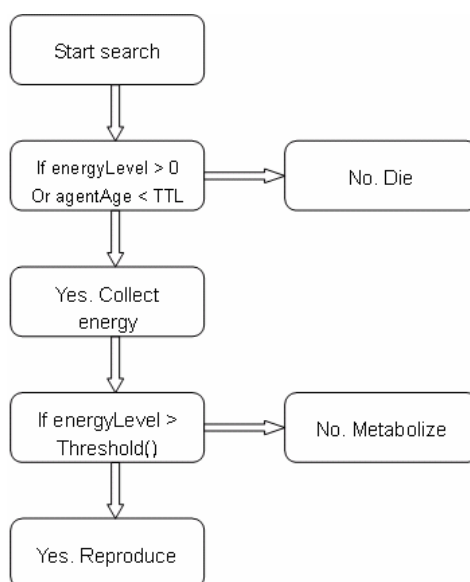


Figure.1 Flowchart of agent behavior

In addition, an agent has a number of static attributes that do not change during their lifetime:

- Search Profile (SP): Built from the informational interest of the peer by which the agent is made.
- Metabolic Rate (MR): The amount of energy it consumes during each hop.
- Energy received at Birth (EB): The energy the agent is born with.
- Energy need for reproduce (ER): The energy the agent needs to attain before it can reproduce.
- Time to Live (TTL): The agent's maximum possible age. It represents the maximum hop-distance a search agent can reach before it gets discarded.

As time evolves, four micro-scale behavioral rules control the search agents-*Query Start Rule (QSR)*, *Resource Search Rule (RSR)*, *Reproduce Rule (RR)* and *Die Rule (DR)*. These rules are explained as following:

Query Rule: A query is initiated by a randomly selected peer who requests for some kinds of resources. To obtain an answer to the request, agents are generated by the peer and flooded in its neighboring peers. The **Query Start Rule** is elaborated below.

Rule 1 QR: *Generate Search Agent (SA) /*Agents are generated by the peer in response to user requests. */*

The peer emanates search agent (SA) to its neighbors.

Search Rule: Once the search agents are emanated, they hop from peer to peer in subsequent time steps. Whenever a search agent moves to a peer, it checks whether the peer has earlier been visited or not. If not, then the agent moves to the peer. In this connection,

each peer maintains a field named visit (V), a field named resource profile (RP) and a field named new energy (NE). A successful search is reported if the required resource can be found in this peer. A flag will be set to true to indicate a successful search. An algorithmic form for the resource search rule (RPR) is presented as follow.

Rule 2 SR: *If (Search agent (SA)) Start*

AA++;

V++;

*If ((V = 1) AND (SP = RP) /*Report a match, V = 1 indicates first time visit by an agent. */*

flag = true;

EL = EL + NE;

Update;

Reproduce Rule: Once the current energy level of the search agent exceeds a threshold, the agent will spawn a new one and splitting its current energy in half.

Rule 3 RR: *If (Search agent (SA)) Start*

*If (EL >= ER) /*The agent get enough energy for reproduce. */*

Produce a new agent;

EL = EL/2;

Die Rule: When the agent's age reach the time to live (TTL) value, or the current energy level of the agent falls below zero, the agent will die.

Rule 4 DR: *If (Search agent (SA)) Start*

If ((AA >= TTL) OR (EL <= 0))

The agent die;

2.3 Performance Metrics

In this paper we focus on efficiency aspects of the processes solely, and use the following simple metrics in our abstract p2p networks.

- Hit rate is defined as the number of resources found for each agent within a given period of time.
- Average number of hops per successful query. This parameter depends on the topology of P2P network as well as on how effectively search mechanism uses it. The less hops is required in average to find requested data, the less traffic is generated and the less time is required for search.
- Population of agents. The number of agents needs to find the requested resource according to the distribution of resource in the uncertain network environments.

Based upon the above mentioned model and metric definition, we now present the experimental results. Simulation runs on Pentium 2.3 GHz with 1GB RAM under windows XP.

3 Simulation Result and Analysis

The experimental results illustrate the efficiency of the algorithms and the effect of controlling the number of agents dynamically based on ecologically inspired control mechanism. This mechanism is completely distributed, executed locally and uses only locally available information. Thus, no globally available information is required. The emergent behavior resulting from the individual localized control decisions will yield an optimal, or sufficiently optimal, solution at the global level. For comparison, we also simulate experiments with k-random walk. The time-step experiment is elaborated next

3.1 Search Efficiency

The search is initiated by a randomly selected node and the number of resource found each time-step from the commencement of the search is calculated. The value of resource hit rate provides the indication of search efficiency. Figure. 2 shows the result of running the time-step experiment for 20 generations (1 generation is 100 time-steps). The time-to-live parameter is set to 25, and k is set to 12, grid size is 100×100.

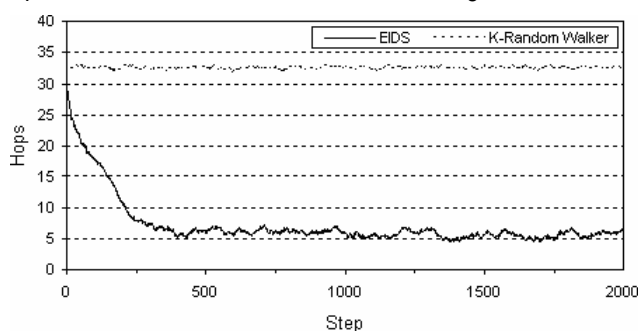


Figure.2 Average number of hops per successful query

We expect that if an agent employs spawn strategy, the total number of agents would increase, the hops required to complete a search would decrease. When EIDS and K-Random Walk are compared, EIDS requires much fewer hops (6 hops) than K-Random Walk (32 hops). In fact, K-Random Walk constantly requires a large number of hops.

The decision for agents spawn strategy is based on a parameter $\rho \in [0, 1]$. For example, if parameter is set to 0.8, each agent will employ the spawn strategy in 80% of the cases. If the parameter is set to 1, then the agent will spawn a new agent whenever its energy level is above a threshold; when the parameter is set to 0, the agent will never spawn any new one, in the fact, it employs the k-random walk strategy.

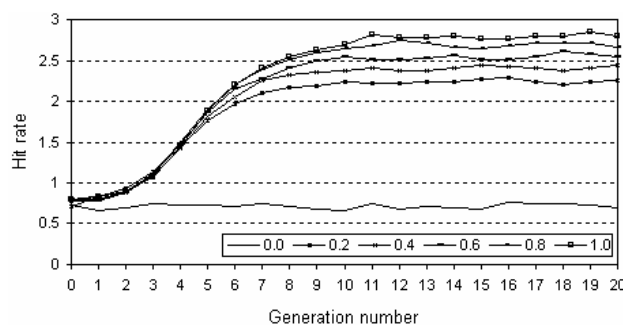


Figure.3 Hit rate per generation

The hit rate is dependent on parameter not only in the start-up phase, but also in the converged phase (Figure.3). The more search agents in the network, the higher the hit rate. The best result can be reached when setting $\rho = 1$. After ten generation, 2.8 resources on average are found in this case. All the five curves employing spawn strategy converge to the same limit over time. The worst result is obtained when setting $\rho = 0$, which is the k-random walk case, the performance stays constant with on average 0.6 resources are found. We see that search efficiency of spawn agents is almost 3-4 times higher than that of k-random walkers.

3.2 Ecosystem Inspired Control of Agent's Population

The system of search agents and the environment they inhabit, i.e., the P2P network, consist of an information ecosystem. All peers can be seen as both information producers and consumers; as consumers, peers send agents (queries) for searching information resources they required, these living agents survive in the context of limited information resources they can find in the network environment. Agent population is determined by the resources of P2P network, the "carry capacity" of the networked information environment, that is, by the size of the relevant set for the given query (Figure.4).

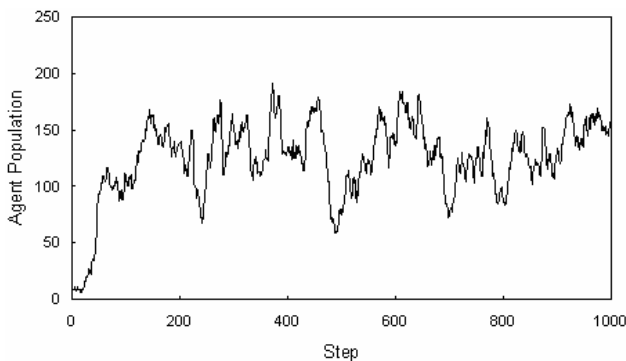


Figure.4 Agent population over time
(TTL=6, Size= 30 × 30)

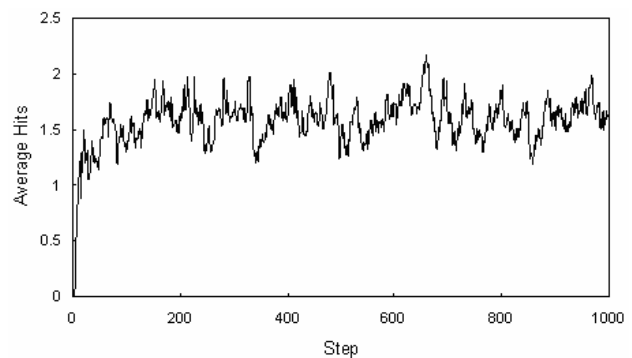


Figure.5 Average hits over time
(TTL=6, Size= 30 × 30)

As the number of agents in the system grows at the beginning of simulation, the resource they find increase rapidly; when the resource the agents can find reach the max amount of resource, agent population fluctuates within a definite scope, that's to say the population reach the carry capacity of network environment, and the hit rate maintain a steady average change (Figure.5).

3.3 Reinforcement Learning

Resources from the environment that are correlated with an agent's performance can be used as reward or penalty signals to adjust the agent's behavior during its life [6]. This is the basis of the reinforcement learning framework. Such signal corresponds to the energy change computed in Rule 2 of the algorithm. It could be computed as the time derivative of the agent's energy level.

The energy level throughout the lifetime of the agent is depicted in Figure.6. The agent reproduces around time 34, giving half of its energy to the offspring. Finally the agent runs out of energy and dies shortly after time 96. The Curve 1 plots the level of accumulated energy as a function of time, resulting from the instantaneous changes in energy plotted by the Curve 2. The selection threshold is $ER = 10$. With the exception of the reproduction event, the energy level is the integral of the Curve 1.

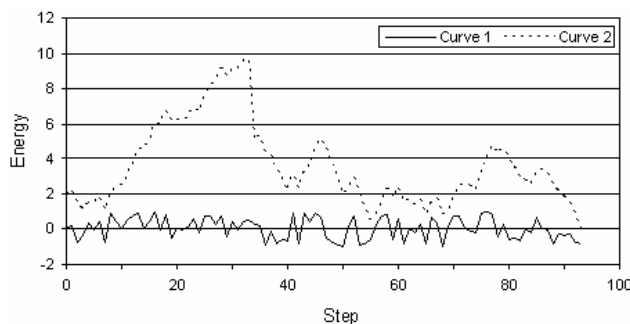


Figure.6 Illustration of energy dynamics in a typical agent's lifetime

Any reinforcement learning scheme can be used to adjust the behavior of the agent so that actions perceived to lead to rewards are reinforced, and action perceived to lead to penalties are discouraged. In this case, the prevailing penalties incurred between time 80 and 95 might have warranted changes leading to a delayed death; the reward incurred around time 34 leads to a new agent offspring, so as to find more peers with similar information resources around the neighborhood.

At the individual level, reinforcement learning biases an agent's behavior toward actions leading to a better knowledge of the environment and to increased payoff, or longer survival. At the collective level, reinforcement evolution biases the population toward sampling areas of the environment that are close to known resources, by increasing the density of agents near those sources.

4 Related Work

The majority of ecology-inspired systems are used to answer some question about real world ecosystems and its properties. Agent-based models are often used to simulating ecosystem. The most well known example is LAGER [Olson, 1995]. The term information ecosystem is used by analogy with natural ecosystems. Information ecosystems ideas can improve our understanding of information infrastructures. An example of agent based systems relevant to information ecosystems research is the InfoSpiders system developed by Menczer and Monge [Menczer, 1999]. This system implements a scalable information search algorithm by use of cooperative agents, drawing explicitly on ecological metaphors. By contrast, the Hive system [Minar, 1999] uses distributed agents to link networked resources on a local network. The MATS system developed by Ghanea-Hercock et al. [Hercock, 1999] uses mobile agents, in this case inspired by social insects, to control a distributed processing application over a network.

There are many agent based algorithms for searching in peer-to-peer networks. The famous Anthill [Babaoglu, 2002] is an open source framework for the design, implementation, and evaluation of ant algorithms in peer-to-peer networks. An Anthill system is an overlay network of interconnected nests (peers). Nests provide services like document storage, routing table management, topology management, and generation of ants upon user requests. In addition, the Anthill API provides a basic set of actions for ants that enables them to travel from nest to nest, and to interact with the services provided by nests. The ant algorithm is not specified by Anthill, but must be designed by the user of the framework according to the application scenario. In our approach, not only those query agents behave well be rewarded to undergo offspring, but also those behave bad be punished to death. So the number of queries can be maintained on a lower level.

5 Conclusions

In this paper, we have concentrated on developing an agent-based model for controlling query messages that are represented as agent; a search algorithm which derives its inspiration from natural ecosystem is presented.

Experiment results above show that this ecologically inspired algorithm is much more efficient search method than k-walker random walk. Each additional step in the search increases the number of nodes visited by only a constant. So exponentially increased over load on each visited node by flooding can be avoided. The basic strengths displayed by the EIDS algorithm need to be further explored and developed, by applying it in more realistic circumstances in the near future.

Bibliography

- [Hoile, 2002] C.Hoile, F.Wang, E.Bonsma, and P. Marrow. Core specification and experiments in diet: a decentralised ecosystem-inspired mobile agent system. In Proc. of AAMAS 2002, pp. 623-630.
- [Lv, 2002] Q.Lv, P.Cao, E.Cohen, and S.Shenker. Search and replication in unstructured peer-to-peer networks. In Proceedings of the 16th ACM Conference on Supercomputing, 2002.
- [Moukas, 1997] A.Moukas, Amalthaea: Information Discover and Filtering using a Multiagent Evolving Ecosystem. Proc. Conf. Practical Applications of Agents and MultiAgent Technology, 1997.
- [Zipf, 1935] G. K. Zipf. Psycho-Biology of Languages. Houghton-Mfflin, 1935.
- [Kaelbling, 1996] P. Kaelbling, L., Littman, M.L., and Moore, A.W. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 1996, pp. 237-285.
- [Olson, 1995] R. L. Olson, A. A. Sequeira. An emergent computational approach to the study of ecosystem dynamics. Ecological Modeling, 1995, pp. 95-120.
- [Menczer, 1999] F. Menczer, and A. E. Monge. Scalable web search by adaptive online agents: an InfoSpiders case study. In: Intelligent Information Agents, M. Klusch, (ed.), Springer, Berlin, 1999.
- [Minar, 1999] N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes. Hive: distributed agents for networking things. In: Proceedings of ASA/MA '99, 1999.
- [Hercocck, 1999] R.G.Hercocck, J.C. Collis, and D.T. Ndumu. Co-operating mobile agents for distributed parallel processing. In: Proceedings of Autonomous Agents 1999, Seattle, 1999.
- [Babaoglu, 2002] O.Babaoglu, H.Meling, and A.Montresor, "Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems," in Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 02). July 2002, IEEE.

Authors' Information

Li Sa - College of Information Sciences and Technology, Donghua University, Shanghai, P.R.China;
e-mail: sali@ustc.edu

Yongsheng Ding - College of Information Sciences and Technology, Donghua University, Shanghai, P.R.China;
e-mail: ysding@dhu.edu.cn