A Multi-Swarm Approach for Neighbor Selection in Peer-to-Peer Networks

Ajith Abraham
National Institute of Applied Sciences
INSA-Lyon, F-69621, France
ajith.abraham@ieee.org

Youakim Badr
National Institute of Applied Sciences
INSA-Lyon, F-69621, France
youakim.badr@insa-lyon.fr

Hongbo Liu
School of Information Science and Technology,
Dalian Maritime University,Dalian, China
lhb@dlut.edu.cn

Crina Grosan
Department of Computer Science
Babes-Bolyai University, Romania
cgrosan@cs.ubbcluj.ro

ABSTRACT

Peer-to-peer (P2P) topology has a significant influence on the performance, search efficiency and functionality, and scalability of the application. In this paper, we investigate a multi-swarm approach to the problem of Neighbor Selection (NS) in P2P networks. Particle swarm optimization algorithm share some common characteristics with P2P in a dynamic social environment. Each particle encodes the upper half of the peer-connection matrix through the undirected graph, which reduces the search space dimension. The synergetic performance is achieved by the adjustment to the velocity influenced by the individual's cognition, the group cognition from multi-swarms, and the social cognition from the whole swarm. The performance of the proposed approach is evaluated and compared with two other different algorithms. The results indicate that it usually required shorter time to obtain better results than the other considered methods, specially for large scale problems.

Categories and Subject Descriptors

Computing Methodologies [**I.2 Artificial Intelligence**]: I.2.8 Problem Solving, Control Methods, and Search

General Terms

Algorithms, Design, Performance

Keywords

P2P Swarming Networks, Neighbor Selection, Particle Swarm, Genetic Algorithm, Undirected Graph

CSTST 2008 October 27-31, 2008, Cergy-Pontoise, France

1. INTRODUCTION

Peer-to-peer computing has recently attracted great interest and attention of the computing industry and gained popularity among computer users and their networked virtual communities [1, 2], since it allows the implementation of large distributed repositories of digital information. Many P2P systems also have emerged as platforms for users to search and share information over the Internet [3]. All participants in a peer-to-peer system act as both clients and servers to one another, thereby surpassing the conventional client/server model and bringing all participant computers together with the purpose of sharing resources such as content, bandwidth, CPU cycles [4]. Peer-to-peer networks are applied to many fields, which includes communication and collaboration, distributed computing, Internet service support, database system, and content/data distribution, even service platform for public welfare (e.g. providing processing power to fight cancer) [5, 6, 7, 8, 9]. It is reported in a recent survey that Peer-to-Peer applications generate one-fifth of the total Internet traffic, and it is believed that it will continue to grow [10].

In pure P2P systems, individual machines communicate directly with each other and share information and resources without using dedicated servers. A node cannot realistically keep the addresses of all other peers, so an overlay network need be constructed where each node keeps addresses of a few other peers (called its neighbors) at the application level. A common difficulty in the current P2P systems is caused by the dynamic membership of peer hosts. The neighbor selection mechanism and topology control become very important topics in P2P networks [11].

On the other hand, the performance and availability of these systems relies on the voluntary participation of their users, and so they may be highly variable and unpredictable, which results in a large proportion of the participants (20 to 40% of Napster and almost 70% of Gnutella peers) share few or no files [12]. This phenomenon is known as free-loading: peers that consume more resources than they contribute. One of the reasons for this problem is that those users, called free-riders, benefit largely from contributions of other users but reduce the system performance for contributing users. Self-interested behavior of the peers has not been taken into account at the design stage. In fact, the P2P system's users

act rationally trying to maximize the benefits obtained from using the system's shared resources. Therefore, it will be necessary to find mechanisms that provide incentives and encourage cooperative behavior between the peers.

Particle Swarm Optimization (PSO) algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the Swarm Intelligence (SI) paradigm has emerged. In this paper, we explore the neighbor-selection problem based PSO for P2P Networks. We introduce the crossover neighborhood organization mechanism from the social networks to improve the swarm algorithm, which results in more mutual trust, mutual benefit, equality and cooperation among the participants.

2. NEIGHBOR-SELECTION PROBLEM

In a P2P system, all participating peers form a P2P network on top of an underlying physical network. A P2P network is an abstract, logical network called an overlay network. Based on existing research [4, 10, 13, 14, 15], we formulate the neighbor-selection problem for P2P overlay networks in this Section. According to Liu, *et al.* [13], a P2P network can be modeled based on the following assumptions:

- An overlay connection between a pair of peering nodes consists of a number of physical links which form a shortest path between the pair of end nodes in the physical topology, and Internet paths are relatively stable.
- The same size packets traversing the same physical link in a short period of time will have similar delay, as assumed by many other measurement applications.

2.1 Modeling P2P Networks

P2P overlay networks can be modeled by an undirected graph $G = (V, E)$ where the vertex set V represents units such as hosts and routers, and the edge set E represents physical links connecting pairs of communicating unit. And $f : V \to \{1, \cdots, n\}$ be a labeling of its nodes, where $n = |V|$. For instance, G could model the whole or part of the network. Given an undirected graph $G = (V, E)$ modeling an interconnection network, and a subset $X \subseteq V(G)$ of communicating units (peers), we can construct a corresponding weighted graph $D = (V, E)$, where $V(D) = X$, and the weight of each $(u, v) \in E(D)$ is equal to the length of a shortest path between peer u and peer v in G. D includes the connected edges, and is referred to as the distance graph of G. Usually we start with a physical network G (perhaps representing the Internet), and then choose a set of communicating peers X. The resulting distance graph D is the basis for constructing a P2P overlay graph $H = (V, E)$, which is done as follows. The vertex set $V(H)$ will be the same as $V(D)$, and edge set $E(H) \subseteq D(G)$. The key issue here is how to select $E(H)$. If $E = [e_{ij}]_{n\times n}$ is such that $e_{ij} = 1$ if $(i, j) \in E$, and 0 otherwise, i.e., E is the incidence matrix of G, then the neighbor-selection problem is to find a permutation of rows and columns which brings all non-zero elements of E into the optimal possible interconnection around the diagonal.

2.2 Problem Formulation

In P2P file sharing, an interested file is divided into many fragments. The size of each fragment ranges from several hundred kilobytes to several megabytes. When a new peer joins the network, it begins to download fragments from other peers. As long as it obtains one fragment of the file, the new peer can start to serve other peers by uploading fragments. Since peers are downloading and uploading at the same time, when the network becomes large, although the demands increase, the service provided by the network also increases. Given N peers, a graph $G = (V, E)$ can be used to denote an overlay network, where the set of vertices $V = \{v_1, \cdots, v_N\}$ represents the N peers and the set of edges $E = \{e_{ij} \in \{0, 1\}, i, j = 1, \cdots, N\}$ represents their connectivities : $e_{ij} = 1$ if peers i and j are connected, and $e_{ij} = 0$ otherwise. For an undirected graph, it is required that $e_{ij} = e_{ji}$ for all $i \neq j$, and $e_{ij} = 0$ when $i = j$. Let C be the entire collection of content fragments, and $\{c_i \subseteq C, i = 1, \cdots, N\}$ denotes the collection of the content fragments each peer i has. The disjointness of contents from peer i to peer j is denoted by $c_i \setminus c_j$, which can be calculated as:

$$c_i \setminus c_j = c_i - (c_i \cap c_j). \tag{1}$$

where $\setminus$ denotes the intersection operation on sets. This disjointness can be interpreted as the collection of content fragments that peer i has but peer j does not. In other words, it denotes the fragments that peer i can upload to peer j. Moreover, the disjointness operation is not commutative, *i.e.*, $c_i \setminus c_j \neq c_j \setminus c_i$. Let $|c_i \setminus c_j|$ denote the cardinality of $c_i \setminus c_j$, which is the number of content fragments peer i can contribute to peer j. In order to maximize the disjointness of content, we maximize the number of content fragments each peer can contribute to its neighbors by determining the connections e_{ij}'s. Define ϵ_{ij}'s to be sets such that $\epsilon_{ij} = C$ if $e_{ij} = 1$, and $\epsilon_{ij} = \emptyset$ (null set) otherwise.

In an overlay network, every node is a potential neighbor of each other node since the network's topology is a logical one. So the full connection is an ideal solution for the peer's connectivity. For the network, we also have to consider some constraints [16, 15]:

- based on the underlying network characteristics, i.e., delay or capacity of actual links;
- based on location of data and services;
- based on the nodes's capabilities of managing peers, e.g., the number of direct neighbors a node can maintain. some peers are tied down since they possess relative more content fragments. This resource constraint can be independent of the underlying network.

In the environment, the maximum number of each peer need to be considered, i.e., each peer i will be connected to a maximum of d_i neighbors, where $d_i < N$. Therefore, we have the following optimization problem:

$$\max_E \sum_{j=1}^{N} \Big| \bigcup_{i=1}^{N} (c_i \setminus c_j) \cap \epsilon_{ij} \Big| \tag{2}$$

Subject to

$$\sum_{j=1}^{N} e_{ij} \leq d_i \text{ for all } i$$
$$\sum_{i=1}^{N} e_{ij} \leq d_j \text{ for all } j \quad (3)$$

3. META-HEURISTICS FOR NEIGHBOR SELECTION

For applying the PSO algorithm successfully for any problem, one of the key issues is how to map the problem solution to the particle space, which affects its feasibility and performance. The constraint conditions have to be satisfied, and the particle would search the solutions in as efficient a search space as possible. In this section, a new approach to the problem space mapping is depicted for particle swarm optimization with reference to the neighbor-selection problem. For solving the problem, the upper half of the peer-connection matrix through the undirected graph is encoded to the particle's position, which reduces the search space dimension significantly. Since particle swarm shares some common characteristics with P2P in the dynamic socially environment, a multi-swarm interactive pattern is introduce to match the corresponding mechanism.

Given a P2P state $S = (N, C, M, f)$, in which N is the number of peers, C is the entire collection of content fragments, M is the maximum number of the peers which each peer can connect steadily in the session, f is to goal the number of swap fragments, *i.e.* to maximize Eq.(2). It is to be noted that the routing and connection between peers must satisfy the constraint in Eq.(3) because of bandwidth, etc. To apply the particle swarm algorithm successfully for the NS problem, one of the key issues is the mapping of the problem solution into the particle space, which directly affects its feasibility and performance. Usually, the particle's position is encoded to map each dimension to one directed connection between peers, *i.e.* the dimension is $N * N$. But the neighbor topology in P2P networks is an undirected graph, *i.e.* $e_{ij} = e_{ji}$ for all $i \neq j$, and $e_{ij} \equiv 0$ for all $i = j$. To reduce the space complexity, we set up a search space of D dimension as $N*(N-1)/2$. Accordingly, each particle's position is represented as a binary bit string of length D. Each dimension of the particle's position maps one undirected connection. The domain for each dimension is limited to 0 or 1.

PSO model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the D-dimension problem space to search the new solutions, where the fitness f can be measured by calculating the number of swap fragments in the potential solution. Each particle has a position represented by a position-vector $\vec{p}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\vec{v}_i$. Each particle remembers its own best position so far in a vector $\vec{p}_i^{\#}$, and its j-th dimensional value is $p_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\vec{p}^{*}$, and its j-th dimensional value is p_j^{*}. When the particle moves in a state space restricted to zero and one on each dimension, the change of probability with time steps is defined as follows:

$$P(p_{ij}(t) = 1) = f(p_{ij}(t-1), v_{ij}(t-1), p_{ij}^{\#}(t-1), p_j^{*}(t-1)). \quad (4)$$

where the probability function is

$$sig(v_{ij}(t)) = \frac{1}{1 + e^{-v_{ij}(t)}}. \quad (5)$$

At each time step, each particle updates its velocity and moves to a new position according to Eqs.(6) and (7):

$$v_{ij}(t) = wv_{ij}(t-1) + c_1 r_1 (p_{ij}^{\#}(t-1) - p_{ij}(t-1)) + c_2 r_2 (p_j^{*}(t-1) - p_{ij}(t-1)) \quad (6)$$

$$p_{ij}(t) = \begin{cases} 1 & \text{if } \rho < sig(v_{ij}(t)); \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Where c_1 is a positive constant, called as coefficient of the self-recognition component, c_2 is a positive constant, called as coefficient of the social component. r_1 and r_2 are the random numbers in the interval [0,1]. The variable w is called as the inertia factor, which value is typically setup to vary linearly from 1 to near 0 during the iterated processing. ρ is random number in the closed interval [0,1]. From Eq.(6), a particle decides where to move next, considering its current state, its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. The particle has priority levels according to the order of peers. The sequence of the peers are not changed during the iteration. Each particle's position indicates the potential connection state.

Some previous studies have discussed the trajectory of particles and its convergence [17, 18, 19]. It has been shown that the trajectories of the particles oscillate as different sinusoidal waves and converge quickly, sometimes prematurely. Various methods have been used to identify some other particle to influence the individual. Eberhart and Kennedy called the two basic methods as "gbest model" and "lbest model" [20]. In the gbest model, the trajectory for each particle's search is influenced by the best point found by any member of the entire population. The best particle acts as an attractor, pulling all the particles towards it. Eventually all particles will converge to this position. In the lbest model, particles have information only of their own and their nearest array neighbors' best (lbest), rather than that of the whole swarm. Namely, in Eq.(6), gbest is replaced by lbest in the model. The lbest model allows each individual to be influenced by some smaller number of adjacent members of the population array. The particles selected to be in one subset of the swarm have no direct relationship to the other particles in the other neighborhood. Typically lbest neighborhoods comprise exactly two neighbors. When the number of neighbors increases to all but itself in the lbest model, the case is equivalent to the gbest model. Some experiment results testified that gbest model converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while lbest model converges slowly on problem solutions but is able to "flow around" local optima, as the individuals explore different regions.

Many multi-swarm approaches illustrate good performance for some complex problems [21, 22]. To match the social characteristics, we introduce a multi-swarm search algorithm

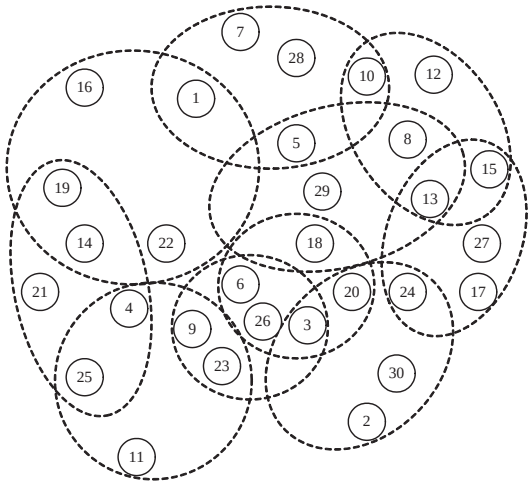

Figure 1: A multi-swarm topology

for neighbor-selection problem in P2P networks. In the proposed algorithm, all particles are clustered spontaneously into different sub-swarms of the whole swarm. Every particle can connect to more than one sub-swarm, and a crossover neighborhood topology is constructed between different sub-swarms. The particles in the same sub-swarm would carry some similar functions as possible and search for their optimal. Each sub-swarm would approach its appropriate position (solution), which would be helpful for the whole swarm to keep in a good balance state. Figure 1 illustrates a multi-swarm topology. In the swarm system, a swarm with 30 particles is organized into 10 sub-swarms, with each sub-swarm consisting of 5 particles. Particles 3 and 13 have the maximum membership level, 3. During the iteration process, the particle updates its velocity followed by the location of the best fitness achieved so far by the particle itself and by the location of the best fitness achieved so far across all its neighbors in all sub-swarms it belongs to. The process makes an important influence on the particles' ergodic and synergetic performance.

Since the positions of all the particles indicate the potential assigned solutions, the binary bit strings of length D can be "decoded" to the feasible solution. "1" denotes the two corresponding peers are selected in the neighborhood. On the contrary, "0" denotes the two corresponding peers are disconnected. The position may violate the constraint (3) after some iterations. We scan each column and row before the decoding procedure. The latest binary bits are set to "0" if $\sum_{j=1}^{N} e_{ij} > d_i$ or $\sum_{i=1}^{N} e_{ij} > d_j$. The scan direction are reversed after each scan. The pseudo-code for the multi-swarm search algorithm is illustrated as follows:

Step 1. Initialize the size of the particle swarm n, and other parameters. Initialize the positions and the velocities for all the particles randomly.
Step 2. Multiple sub-swarms n are organized into a crossover neighborhood topology. A particle can join more than one sub-swarm. Each particle has the maximum membership level l, and each sub-swarm accommodates default number of particles m.
Step 3. Decode the positions and evaluate the fitness for each particles.

3.01 For $s = 1$ to n
3.02 If ($reverse$)
3.03 For $i = 0$ to $N - 1$
3.04 $e = 0$
3.05 For $j = 0$ to $N - 1$
3.06 If $(j == i)$ $e_{ij} = 0$;
3.07 If $(j < i)$ $a = j; b = i$;
3.08 If $(j > i)$ $a = i; b = j$;
3.09 If $(e > d_j)$ $p_{[a*N+b-(a+1)*(a+2)/2]} = 0$
3.10 else
3.11 If $(p_{[a*N+b-(a+1)*(a+2)/2]})$,
3.12 Calculate $c_i \setminus c_j$; $e++$;}
3.13 End if
3.14 Next j
3.15 Next i
3.16 else
3.17 For $i = N - 1$ to 0
3.18 $e = 0$
3.19 For $j = N - 1$ to 0
3.20 If $(j == i)$ $e_{ij} = 0$;
3.21 If $(j < i)$ $a = j; b = i$;
3.22 If $(j > i)$ $a = i; b = j$;
3.23 If $(e > d_j)$ $p_{[a*N+b-(a+1)*(a+2)/2]} = 0$
3.24 else
3.25 If $(p_{[a*N+b-(a+1)*(a+2)/2]})$,
3.26 Calculate $c_i \setminus c_j$; $e++$;}
3.27 End if
3.28 Next j
3.29 Next i
3.30 End if
3.31 Calculate $f = f + \left| \bigcup_{i=1}^{N} (c_i \setminus c_j) \cap \epsilon_{ij} \right|$;
3.32 If ($rand(0,1) < 0.5$) $reverse = 0$
3.33 else $reverse = 1$;
3.34 Next s

Step 4. Find the best particle in the swarm, and find the best one in each sub-swarms. If the "global best" of the swarm is improved, $noimprove = 0$, otherwise, $noimprove = 1$. Update velocity and position for each particle at the iteration t.

4.01 For $m = 1$ to $subs$
4.02 $\vec{p}^{*} = argmin_{i=1}^{subs_m}(f(\vec{p}^{*}(t-1)), f(\vec{p}_1(t)),$
4.02 $f(\vec{p}_2(t)), \cdots, f(\vec{p}_i(t)), \cdots, f(\vec{p}_{subs_m}(t)))$;
4.03 For $ss = 1$ to $subs_m$
4.04 $\vec{p}_i^{\#}(t) = argmin(f(\vec{p}_i^{\#}(t-1)), f(\vec{p}_i(t))$;
4.05 For $d = 1$ to D
4.06 Update the d-th dimension value of $\vec{p}_i$ and $\vec{v}_i$
4.06 according to Eqs.(6) and (7);
4.07 Next d
4.08 Next ss
4.09 Next m

Step 5. If $noimprove = 1$, goto Step 2, the topology is re-organized. If the end criterion is not met, goto Step 3. Otherwise, output the best solution, the fitness.

4. EXPERIMENTAL RESULTS

To illustrate the effectiveness and performance of the particle swarm optimization algorithm, we illustrate an execution trace of the algorithm for the NS problem. A file of size 7 MB is divided into 14 fragments (512 KB each) to distribute, 6 peers download from the P2P networks, and the connecting maximum number of each peer is 3, which is represented as $(6, 14, 3)$ problem. In some session, the state

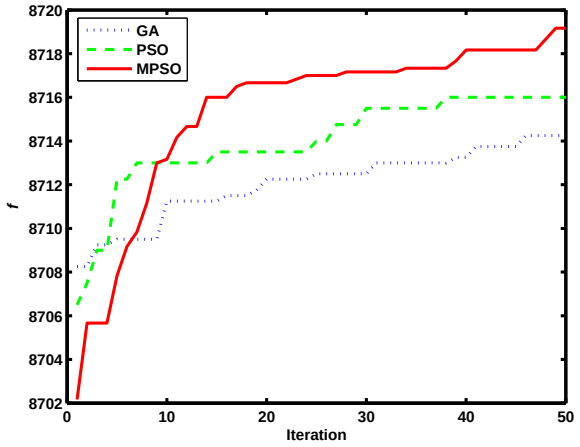

Figure 2: Performance for the NS $(25, 1400, 12)$

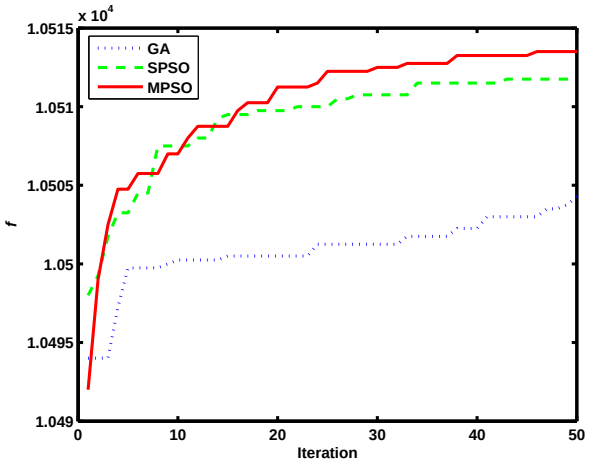

Figure 3: Performance for the NS $(30, 1400, 15)$

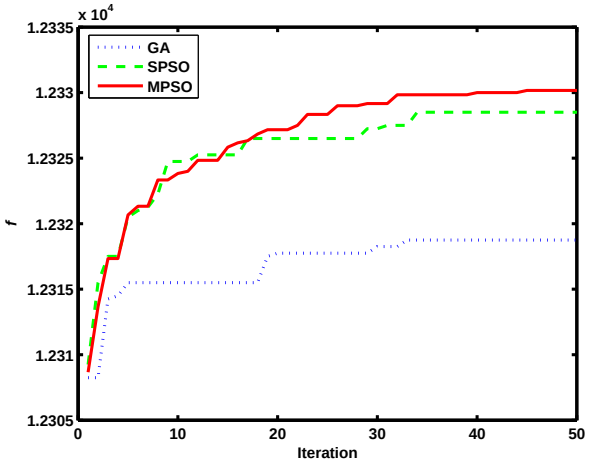

Figure 4: Performance for the NS $(35, 1400, 17)$

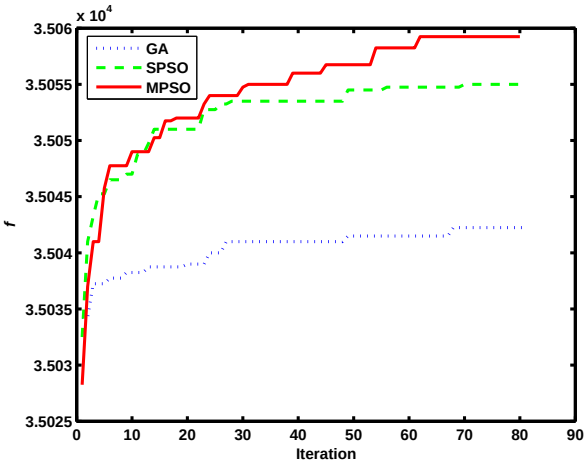

Figure 5: Performance for the NS $(100, 1400, 20)$

of distributed file fragments is as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 6 & 7 & 8 & 0 & 10 & 0 & 12 & 0 & 14 \\ 0 & 0 & 0 & 4 & 5 & 0 & 7 & 0 & 9 & 0 & 11 & 0 & 13 & 0 \\ 0 & 2 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 11 & 12 & 0 & 14 \\ 0 & 2 & 3 & 4 & 0 & 6 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 7 & 8 & 0 & 10 & 0 & 12 & 0 & 14 \\ 1 & 2 & 0 & 0 & 5 & 0 & 0 & 0 & 9 & 10 & 11 & 0 & 13 & 14 \end{bmatrix}$$

The optimal search result by the multi-swarm algorithm is 31, and the neighbor selection solution is illustrated below:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

We also tested three other representative instances (problem (25,1400,12), problem (30,1400,15), problem (35,1400,17) and problem (100,1400,20)). In our experiments, the algorithms used for comparison were mainly Standard PSO (SPSO) [20] and Genetic Algorithm (GA). These algorithms share many similarities. Genetic algorithms mimic an evolutionary natural selection process. Generations of solutions are evaluated according to a fitness value and only those candidates with high fitness values are used to create further solutions via crossover and mutation procedures. The considered algorithms were repeated 4 times with different random seeds. Each trial had a fixed number of 50 or 80 iterations. Other specific parameter settings of the algorithms are described in Table 1. $S = (even)(int)(10 + 2 * sqrt(D))$, where D is the dimension of the position. The average fitness values of the best solutions throughout the optimization run were recorded. The average and the standard deviation were calculated from the 4 different trials.

Figures 2, 3, 4 and 5 illustrate the performances during the search processes using the considered algorithms to solve the NS problems. The best values, mean values, the standard deviations for 4 trials are shown in Table 2. As evident, the multi-swarm algorithm obtained better results much faster

Table 1: Parameter settings for the algorithms.

Algorithm	Parameter name	Value
	Size of the population	S
GA	Probability of crossover	0.8
	Probability of mutation	0.01
	Swarm size	S
	Self coefficient c_1	$0.5 + log(2)$
PSO(s)	Social coefficient c_2	$0.5 + log(2)$
	Inertia weight w	0.91
	Clamping Coefficient ρ	0.5

Table 2: Performance comparison of the three algorithms.

Instance	Item	GA	SPSO	MPSO
	Best	8716.00	8717.00	8721.00
$(25, 1400, 12)$	Mean	8.714.30	8716.00	87192.00
	Std. dev.	1.7078	1.1547	1.3292
	Best	10513.00	10514.00	10515.00
$(30, 1400, 15)$	Mean	10504.00	10512.00	10514.00
	Std. dev.	6.3443	1.2990	1.2910
	Best	12321.00	12332.00	12332.00
$(35, 1400, 17)$	Mean	12319.00	12329.00	12330.00
	Std. dev.	1.7078	2.5166	1.1690
	Best	35047.00	35057.00	35061.00
$(100, 1400, 20)$	Mean	35042.25	35055.00	35059.25
	Std. dev.	3.6996	1.2247	1.0897

than other algorithms, especially for large scale problems.

5. CONCLUSIONS

In this paper, we investigated the neighbor-selection problem in peer-to-peer networks by using a swarm intelligence approach. Since particle swarm shares some common characteristics with P2P in the dynamic socially environment, a multi-swarm interactive pattern was introduced to match the corresponding mechanism. We evaluated the performance of the proposed approach and compared it with GA and SPSO. Empirical results indicates that multi-swarm approach usually obtained better results much faster than GA and SPSO, specially for large scale problems.

Acknowledgment

This work is supported partially by NSFC Grant 60873054 and DLMU Grant DLMU-ZL-200709.

6. REFERENCES

[1] Lua E K, Crowcroft J, Pias M, Sharma R and Lim S (2005) A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. IEEE Communications Surveys & Tutorials, 7(2):72–93

[2] Kwok S (2006) P2P Searching Trends: 2002-2004. Information Processing and Management, 42:237–247

[3] Huang X, Chang C and Chen M (2006) PeerCluster: A Cluster-Based Peer-to-Peer System. IEEE Transactions on Parallel and Distributed Systems, 17(10):1110-1123

[4] Belmonte M V, Conejo R, Díaz M and Pérez-de-la-Cruz J L (2006) Coalition Formation in P2P File Sharing Systems. Lecture Notes in Artificial Intelligence, CAEPIA'05, 4177:153–162

[5] Idris T and Altmann J (2006) A Market-managed Topology Formation Algorithm for Peer-to-Peer File Sharing Networks, Lecture Notes in Computer Science, 4033:61–77

[6] Pianese F, Perino D, Keller J and Biersack E W (2007) PULSE: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System. IEEE Transactions on Multimedia, 9(8):1645–1660

[7] Sigurdsson H M, Halldorsson U R and Hasslinger G (2007) Potentials and Challenges of Peer-to-Peer Based Content Distribution. Telematics and Informatics, 24:348–365

[8] Yang S and Chen I (2008) A Social Network-based System for Supporting Interactive Collaboration in Knowledge Sharing Over Peer-to-Peer Network. International Journal of Human-Computer Studies, 66:36–50

[9] Kim J K, Kim H K and Cho Y H (2008) A User-oriented Contents Recommendation System in Peer-to-Peer Architecture. Expert Systems with Applications, 34:300–312

[10] Sen S and Wang J (2004) Analyzing Peer-to-Peer Traffic Across Large Networks. IEEE/ACM Transactions on Networking, 12(2):219–232

[11] Ardizzone E, Gatani L, La Cascia M, Lo Re G and Ortolani M (2007) Enhanced P2P Services Providing Multimedia Content. Advances in Multimedia, 1–12

[12] Androutsellis-theotokis S and Spinellis D (2004) A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, 36(4):335–371

[13] Liu Y, Xiao L, Esfahanian A and Ni L M (2005) Approaching Optimal Peer-to-Peer Overlays. Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 407–414

[14] Ghanea-Hercock R A, Wang F and Sun Y (2006) Self-Organizing and Adaptive Peer-to-Peer Network. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 36(6):1230-1236

[15] Wang S, Chou H, Wei D and Kuo S (2007) On the Fundamental Performance Limits of Peer-to-Peer Data Replication in Wireless Ad hoc Networks. Journal on Selected Areas in Communications, 25(1):211–221

[16] Surana S, Godfrey B, Lakshminarayanan K, Karp R and Stoica I (2006) Load Balancing in Dynamic Structured Peer-to-Peer Systems. Performance Evaluation, 63:217–240

[17] Clerc M and Kennedy J (2002) The Particle Swarm - Explosion, Stability, and Convergence in A Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation, 6(1):58–73

[18] van den Bergh F and A.P. Engelbrecht A P (2006) A Study of Particle Swarm Optimization Particle Trajectories. Information Sciences, 176:937–971

[19] Liu H, Abraham A and Clerc M (2007) Chaotic Dynamic Characteristics in Swarm intelligence. Applied Soft Computing, 7:1019–1026

[20] Kennedy J and Eberhart R (2001) Swarm Intelligence, Morgan Kaufmann, CA

[21] Grosan C, Abraham A, and Nicoara M (2005) Search Optimization Using Hybrid Particle Sub-Swarms and Evolutionary Algorithms. International Journal of Simulation Systems, Science & Technology, 6(10-11):60–79

[22] Elshamy W, Emara H M, and Bahgat A (2007) Clubs-based Particle Swarm Optimization, Proceedings of the IEEE International Conference on Swarm Intelligence Symposium, 1:289–296