

A New Scheduling Mechanism of BitTorrent Streaming System Based on Improved PSO Algorithm

Yuqi Hu^{1*}, Qingqing Yang²

¹ School of Information Science and Engineering, Yanshan University. The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao Hebei 066004, China.

² School of Information Science and Engineering, Yanshan University, Qinhuangdao Hebei 066004, China.

* Corresponding author. Tel.: 137-0335-9364; email: hu_yuqi@sina.com

Manuscript submitted April 23, 2014; accepted July 10, 2014.

doi: 10.17706/jcp.10.1.34-44

Abstract: In order to provide streaming media service with higher quality, a new data scheduling mechanism based on improved particle swarm optimization algorithm is proposed. This mechanism applies the idea of particle swarm optimization to peer selection of BitTorrent (BT) streaming system. First, improved particle swarm optimization (IPSO) algorithm is developed to overcome the shortcoming of falling easily into local optimum. Second, particle encoding and decoding are defined in the context of streaming media data scheduling, and fitness function based on the service ability of a peer is proposed. Third, IPSO-based neighbor peer selection (IPSO-NPS) algorithm is designed, in which the optimal solution of IPSO determines the set of peers providing data block resources. Simulation results show that IPSO algorithm enhances the convergence speed and accuracy, and the new data scheduling mechanism of IPSONPS reduces the startup delay of system, enhances the media playing continuity and reduces the network load.

Key words: Particle swarm optimization, BitTorrent streaming system, data scheduling, peer selection.

1. Introduction

In recent years, video streaming service has become one of the most important services provided by Internet. Huge amounts of users are attracted by video streaming service, making rapid growth of Internet traffic. According to the latest report of Visual Networking Index (VNI) [1], annual global IP traffic will surpass 1.4 zettabytes by the end of 2017. 3D and HD Internet video users will increase by four times, and video on demand (VoD) traffic will increase by three times. Increasing users and massive multimedia data bring great challenges to large scale VoD service system. How to provide VoD users with high-quality services is a hot issue in streaming media applications.

Using idle computer storage and bandwidth resources to build P2P-based VoD system is a way to improve playback quality and reduce congestion at data servers. P2P systems rely on voluntary participation of peers to help each other. A peer not only requests data as a client, but also provides service to neighbors as a server, which makes P2P system transfer data quickly and efficiently, and effectively solves the server bottleneck problem of the C/S mode system. In P2P streaming system, proper scheduling policy not only achieves strong service capabilities, high throughput, short startup delay and good streaming continuity, but also reduces the waste of network bandwidth [2].

2. Related Works

There are four categories of classic P2P streaming data scheduling algorithm, Random policy [3], Rarest First (RF) policy [4], Nearest deadline First (NF) policy [5] and Weighted Round Robin (WRR) policy [6]. Pai *et al.* [3] proposed that the requesting peer select neighbors randomly. But this algorithm is too simple, without considering the physical location, bandwidth and other factors of peers. Zhang *et al.* [4] proposed scheduling the rarest pieces first, but this method ignores the urgency of video playing, which may bring a longer waiting delay. Zhou *et al.* [5] proposed downloading pieces near the playing point first, but it may lead to the fact that pieces near the end of the video are few, and peers in different progresses of watching are difficult to share data. Agarwal *et al.* [6] presented WRR policy for peer selection, and neighbors were selected by the proportion of bandwidth weights. But this algorithm may bring longer transmission delay, without considering how far the peer is.

BT, probably the most popular P2P file sharing system, is highly similar with mesh streaming system. Currently, significant research effort has focused on the effective use of BT architecture to provide large-scale VoD services [7]–[9]. But there are still some shortcomings and deficiencies. Liu *et al.* [10] designed LRP (Least Request Peer) policy. This policy selects the neighbors that have received the least number of requests, but it ignores the distance and stability of peers. Ma *et al.* [11] developed CAP (Closest-Ahead Peers) algorithm to make better use of the upload bandwidth of peers. Peers will request pieces from the closest-ahead peers whose sequential download process is just ahead of the requesting peer's. But CAP method may lead to the closest-ahead peer overloading situation, without considering the overall system performance. Rivero-Angeles *et al.* [12] proposed selecting neighbors according to their progress in the file download process relative to the progress of the requesting peers. Peers in the group of downloading the file at window w_i download pieces from peers in any group downloading the file at window w_j for $j > i$ but this policy ignores the upload bandwidth and physical location of peers. In this paper, we propose an IPSO-NPS scheduling mechanism to improve service quality of BT streaming system. Our contributions mainly include three aspects.

- 1) IPSO algorithm is developed by improving the iteration formula of original PSO algorithm.
- 2) A fitness function based on the service ability of peer is defined, and the methods of encoding and decoding applied to the data scheduling are proposed.
- 3) The IPSO-NPS algorithm for peer selection is designed. Experimental results show that IPSO algorithm enhances the convergence speed and accuracy, and the IPSO-NPS method not only improves the streaming continuity, but also reduces the startup delay and the network load.

3. System Model

There are two types of BT streaming media system, pure P2P system and hybrid system, depending on whether a media server exists. Hybrid systems deploy media servers to improve the quality of video playing, while pure P2P systems just take the architecture of BT protocol completely. Pure P2P system achieves the high-quality service by improving peer selection, piece selection and TFT algorithm. Notably, these improved algorithms are also applied to hybrid systems. Therefore, we choose the simplest pure P2P system to verify the effectiveness of the new scheduling mechanism proposed by us.

The system consists of three main components, tracker directory server, web server and peers, as shown in Fig. 1. When a new peer joins, it downloads the .torrent file of target video from a web site at first. This .torrent file contains the meta-data such as the address of tracker directory server. Then, the new peer contacts to the tracker to retrieve its neighbors, and the tracker returns a list of potential peers that might share the video with the requesting peer. The new peer then establishes the connections with the neighbors which it is willing to download from.

BT uses swarm techniques, in which the video file is split into many pieces (typically 256KB per piece). In

that way, peers can simultaneously download multiple pieces from other peers. But BT is not suited for VoD applications. The main reason is that pieces over the network are not distributed in a specific order, and the piece selection is based on a rarest-first rule. Streaming VoD system requires the data arrive timely and orderly, and the playing continuity index is an important evaluation. So we adapt its design to satisfy the fundamental quality-of-service requirements for streaming, while still maintaining the high efficiency of the original BT protocol. Based on [13], we design an adaptive double window mechanism for piece selection, considering the urgency and scarcity of pieces. There are two windows in the video file, emergency window and normal window. Emergency window contains an appropriate number of pieces close to the playback point waiting for play, and the sequential policy is used in this window to maintain smooth streaming. The emergency window will slide forward when a piece is played. Pieces outside the emergency window are belonged to the normal window and rarest-first policy is used to maintain piece diversity.

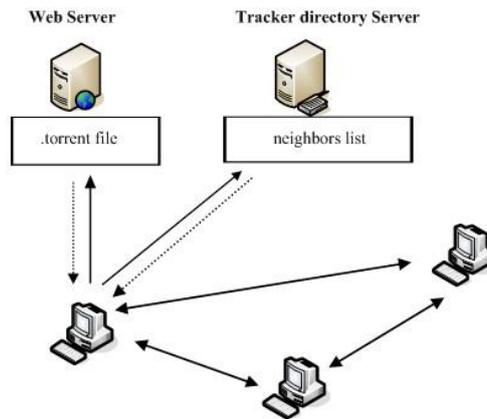


Fig. 1. System model.

In the data scheduling process, the peer requests the pieces missed in the emergency window first, and select $n - r$ pieces from the normal window to download using the rarest-first policy simultaneously, where n is the maximum number of pieces that a peer can request simultaneously, and r is the pieces missed in the emergency window.

The size of the emergency window affects the video playback continuity, too. A larger window makes the video become a sequential downloading, reducing pieces diversity. If the window is too small, the video will not play smoothly. We take an adaptive window strategy [14] that each peer dynamically computes its window size. When the user has fast-forward request or the playback buffer is near empty, it is necessary to shrink the emergency window and then re-request the pieces from the playback point. Each peer computes its emergency window as follows:

$$w = \max[k(d - p - \theta), 0] + w_{\min} \quad (1)$$

where w is the size of emergency window, k is a scaling coefficient, d is the last contiguous pieces available in the playback buffer before the emergency window sliding forward, p is the playback position, w_{\min} is the minimum size of emergency window and θ is a threshold that places a lower bound on the number of contiguous pieces required in the playback buffer before the emergency window grows.

4. Peer Selection Mechanism Based on IPSO

In the unmodified BT system, tracker returns a random list of neighbors to the requesting peer. But the random policy wastes the upload bandwidth of peer and reduces the system's efficiency. Peer selection problem is also the problem of pieces assignment in multiple neighbors, because the pieces may come from

different peers. In this section, we propose an IPSO-NPS mechanism and select peers by seeking particle swarm optimal solution.

4.1. IPSO Algorithm

PSO algorithm [15], a new swarm intelligence optimization algorithm following genetic algorithm, was first proposed by Kennedy and Eberhart in 1995. The traditional PSO algorithm is a simulation of the social behavior of birds or fish seeking for food [16]. The birds or fish are called as particles to solve the optimization problems. Particles are initialized with random positions x_{id} and velocities v_{id} in a D -dimensional space. The particle of i is represented as $x_{id} = (x_{i1}, x_{i2}, \dots, x_{iD})$ and the rate of position change (velocity) is represented as $v_{id} = (v_{i1}, v_{i2}, \dots, v_{iD})$. Each particle adjusts itself for the next step according to the best position of neighbors and its current position. The particles are manipulated by the following two equations:

$$v(t+1) = w * v(t) + c_1 r_1 (p_{best} - x(t)) + c_2 r_2 (g_{best} - x(t)). \quad (2)$$

$$x(t+1) = x(t) + v(t+1). \quad (3)$$

where w is the inertia weight, c_1 and c_2 are two positive constants, r_1 and r_2 are two random numbers in the range $[0, 1]$, $v(t)$ is the current velocity, $v(t+1)$ is the next time velocity, p_{best} is the best position found so far by the individual particle, and g_{best} is the best position found so far by all the particles.

Hu *et al.* [17] have proposed a simple PSO (SPSO) algorithm, which the particle does not have velocity in the evolutionary process. The theoretical derivation and experimental results show the efficiency of SPSO algorithm. And the position of each particle is updated using the equation:

$$x(t+1) = w * x(t) + c_1 r_1 (p_{best} - x(t)) + c_2 r_2 (g_{best} - x(t)). \quad (4)$$

SPSO algorithm avoids the problem of arbitrary velocity parameters that will affect the rate of convergence, but there are still local optimum value cases. Equation (4) shows that, the position of the individual particle is determined by its inertia, personal experience and social experience. The random parameters r_1 and r_2 directly determine the driven degree of its own experience and social experience. If r_1 and r_2 are too large, the particles will be driven too much by personal experience and social experience. If they are too small, personal and social experience is not fully utilized, and the entire group will not move to a better search area. In addition, the random inertia weight w will also affect the convergence of the particle swarm. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. Different from previous studies [18]–[20], we adjust the proportion of random parameters r_1 and r_2 . And the inertia weight w is replaced by the random parameter r_2 . Then, the position update formula of IPSO algorithm is:

$$x(t+1) = r_2 * x(t) + c_1 r_1 (1 - r_2) (p_{best} - x(t)) + c_2 r_2 (1 - r_1) (g_{best} - x(t)). \quad (5)$$

For comparison, six classic benchmark functions (Rastrigin, Ackly, Sphere, Griewank, Shaffer and Rosenbrock) [21] are used here. The experimental results show that IPSO algorithm improves the convergence velocity and precision in the evolutionary optimization.

4.2. Encoding and Decoding of the Particles

There are two coding modes, direct and indirect encoding [22]. In direct encoding, each particle corresponds to a solution of the problem, which in indirect encoding, each particle corresponds to a task assignment strategy. Considering the specific fragment mechanism of BT, we take the indirect encoding and propose numeric string encoding based on peer ID. Coding length depends on the number of pieces downloaded simultaneously in data scheduling process, and the value of each dimension of the particle does not exceed the total number of peers. The position of each particle represents pieces assigned between peers. We denote the encoding of each particle as $p = [p_1, p_2, \dots, p_n]$, where n is the number of pieces, and p_n represents the peer ID. For example, encoding [1, 5, 4, 3, 1, 2, 3] shows that the first piece downloaded from the first peer, the second piece downloaded from the fifth peer, and so on, the last piece downloaded from the third peer in this data scheduling process.

The PSO algorithm has tackled integer programming problems [23]. When decoding, we just take round-down to convert the real space to an integer space before calculating fitness function. In other words, we use round-down operation for each dimension of the particles in order to get the task assignments of peers corresponding to the particles.

4.3. Fitness Function of IPSO-NPS

If the IPSO algorithm is applied to the data scheduling of streaming system, another key is the fitness function. The fitness function is to determine whether the particle swarm to find the optimal solution. We propose a new fitness function based on the service ability of the peer. Each peer is defined as a two-tuples, $Ns = [Id, Capa]$, where $Capa$ represents the service capability of the Id_{th} peer. And $Capa$ is defined as a triple, $Capa = [Bw, Dis, T]$. Then we define the following parameters, as shown in Table 1.

Table 1. Parameters and Definitions

Parameters	Definitions
Id	ID of all peers in the overlay network
$Neigh_i$	All neighbors of peer i recorded at the Tracker directory server
T_i^j	Online time of the neighbor j of peer i
Dis_i^j	The relative distance between peer i and peer j
Bw_i^j	Upload bandwidth of the neighbor j of peer i
$Capa_i^j$	Service capacity of the neighbor j of peer i
b_{ij}^k	$b_{ij}^k = 1$ represents piece k requested by peer i is stored at peer j , while $b_{ij}^k = 0$ represents piece k is not at peer j .

In fact, shortening the startup delay is very important, which directly affects the performance of the on-demand system. If the system returns nearest neighbors in the topology to the downloading peer, it can reduce the transmission delay. Peers are autonomous in P2P network, and they join and leave at any time. In order to reduce the probability of connecting unstable peers, we select the peer first whose online time is longer. Thus, we define the calculated ability equation of a peer as follows:

$$Capa_i^j = \left[(Bw_i^j)^2 * T_i^j \right] / Dis_i^j. \tag{6}$$

where Dis_i^j represents the relative distance between peer i and peer j . As we known, each external IP

address corresponds to a certain geographical information, represented by latitude and longitude. According to Mercator projection algorithm [24], the geographical information corresponds to a coordinate (x, y) if mapped to the planar coordinate system. And at our simulation platform, each peer i corresponds to a coordinate (x_i, y_i) if mapped to the planar coordinate system, too. Although the best distance equals to the minimum routing hops in the actual network, we use Euclidian distance here. So the calculation formula of distance follows:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (7)$$

For data scheduling in streaming media system, the main factor of peer selection is the service ability of peer. So we take it as the evaluation parameter of the data scheduling. And the fitness function based on the peer's service capacity is defined as follows:

$$fitness = \sum_{j \in Neigh_i, b_{ij}^k=1} Capa_i^j. \quad (8)$$

Thus, the best neighbors are the peers that make the formula (9) reach the maximum value.

$$\max \sum_{j \in Neigh_i, b_{ij}^k=1} Capa_i^j. \quad (9)$$

4.4. Algorithm Description of IPSO-NPS

Steps of peer selection mechanism in BT streaming system are as follows.

- 1) When a peer sends a data request, the streaming system determines the pieces to be downloaded first, according to the adaptive double window piece selection algorithm mentioned above.
- 2) The tracker directory server returns the list of neighbors containing the pieces which satisfy the criteria in the first step.
- 3) According to peer selection algorithm, the tracker server selects the optimal neighbors to provide video resources.

Based on the above ideas, we propose the IPSO-NPS algorithm for peer selection problem. Let G denote a set of pieces requested by downloading peer in the data scheduling process. Let N , k and t denote the total number of particles, the maximum iterations and the unchanged time of fitness value. Let j be the best particle meets formula (9). The IPSO-NPS algorithm is described as Algorithm 1.s

Algorithm 1: IPSO-NPS algorithm

Require: G , N , k , t

Ensure: j

- 1: For each particle $i \in N$ do
- 2: Initialize(i)
- 3: End for
- 4: While steps of iteration $< k$ or the unchanged time of fitness value $< t$ do
- 5: for each particle $i \in N$ do
- 6: Update particle position according to the Equ.(5)
- 7: Calculate fitness value according to the Equ.(8)
- 8: If $fitness > p_{best}$ then
- 9: $p_{best} \leftarrow fitness$

```

10: end if
11: If  $fitness > g_{best}$  then
12:  $g_{best} \leftarrow fitness$ 
13:  $j \leftarrow i$ 
14: End if
15: End for
16: End while
17: Return the value of each dimension of  $j$ 

```

Then the set of neighbors providing video file is determined, according to the numeric string encoding based on peer ID proposed in Subsection 4.2.

5. Experimental Evaluations

5.1. Experiment of IPSO

Comparing the IPSO algorithm with the basic PSO (BPSO) algorithm [15] and the simple PSO (SPSO) algorithm [17], we use six classic benchmark functions. In order to facilitate the observation of the evolution curve, we take the base 10 logarithm for each fitness value and plus 10^{-7} as a cut-off value. The experimental results are shown in Fig. 2.

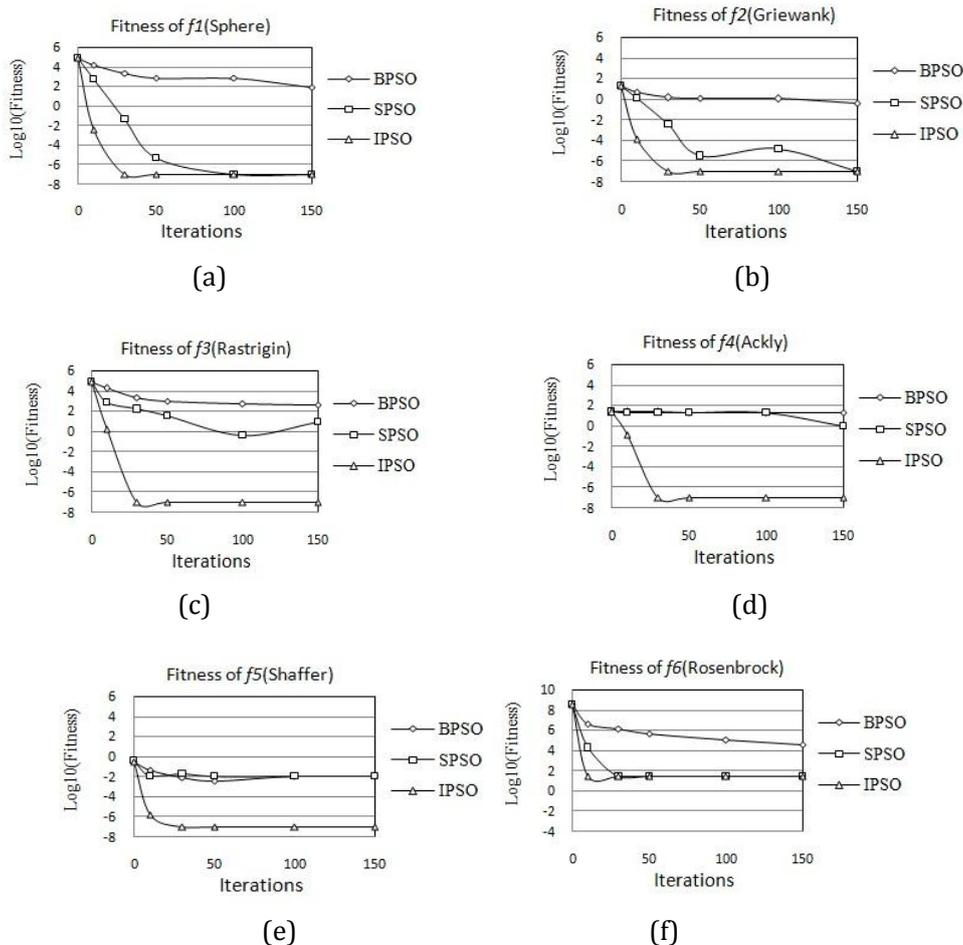


Fig. 2. Evolutionary curves of fitness function $f_1 - f_6$.

The Fig. 2 above shows that, IPSO algorithm has achieved better performance than SPSO and BPSO

algorithms. As shown in the figure (a) and (b), SPSO algorithm escapes from the local optima, but its rate of convergence is less than the IPSO algorithm. Iterating at around 50 times, IPSO algorithm gets the optimal solution, while BPSO and SPSO algorithms appears a local optimum phenomenon as shown from the figure (c) to (e).

5.2. Experiment of Scheduling Mechanism

To demonstrate the performance of our proposed approach, an evaluation is made using various metrics on the General Peer-to-Peer Simulator (GPS) [25] platform. GPS is currently the only one simulation platform supporting BT protocol without any extension. In order to support the streaming media playing, we modify the piece selection of BT. The piece selection is modified by the adaptive double window piece selection algorithm. Then, we modify the peer selection according to three different strategies, Random policy, Least Request Peer (LRP) policy and IPSO-NPS proposed by us. The shared media is 50MB with a piece size of 256KB, except the last piece. The video bitrate is 512Kbps. The bandwidth of peer can be set by GPS platform.

Then, we evaluate the three different scheduling strategies from the following indicators.

1) Startup Delay. The time used to download the pieces to be cached before video playing is defined as startup delay.

2) Continuity Index. During the video playing, the number of pieces that arrived before the playback deadline over the total number of pieces is defined as continuity index.

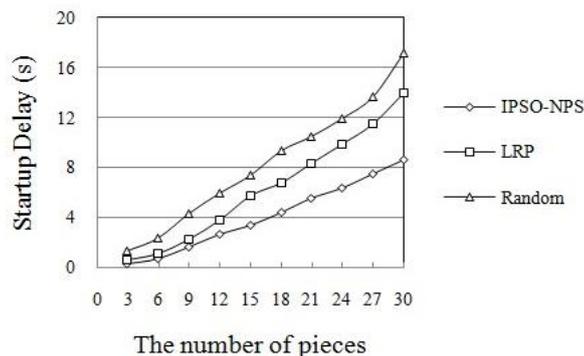


Fig. 3. The startup delay of system.

3) The network load. In the data scheduling process, we calculate an average value of each peer load.

The startup delay is presented in Fig. 3. When requesting the same number of pieces, IPSO-NPS policy has the smaller startup delay than LRP policy and Random policy. The startup delay of three policies grows with the increase of pieces required to cache, but the growth of IPSO-NPS is slower than others. When the requested pieces reach 30, the growth of IPSO-NPS is 10.2% lower than Random policy. This is because IPSO-NPS mechanism selects peers with higher service ability. In other words, IPSO-NPS policy can reduce the probability of connecting unstable peers, and reduce the transmission delay from the remote peer.

As shown in Fig. 4, the continuity index of media playing gets better with the expansion of the network scale, because the more peers in BT network, the faster the downloading rate and the higher the playing continuity. When the number of peers is 512, the continuity index of IPSO-NPS policy reaches about 0.99. But the LRP policy is less than 0.95 and the Random policy is even lower. This is because of the limited upload bandwidth of nodes. With suboptimal peer selection strategy, data may not promptly transmit to the requesting peer before video playback deadline, giving users a buffer waiting phenomenon or causing the loss of data. In addition, with the increase of peers, it also expands the search space for particles to make full use of advantages of the IPSO-NPS mechanism.

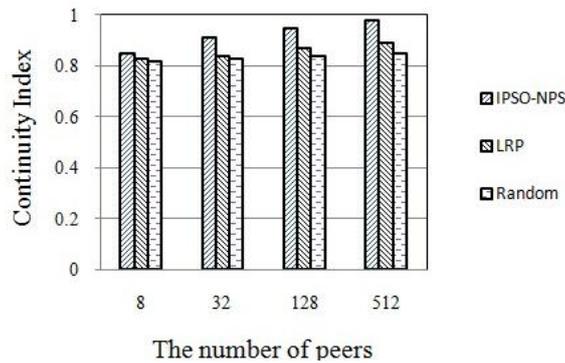


Fig. 4. Continuity index of media playing.

Fig. 5 shows that, with the increase of peers in the network, the network load of the three strategies is reduced. This is because the expansion of the network makes the copies of video resources increase, reducing the load pressure on a single node. IPSO-NPS mechanism measures the service ability of neighbor to reduce the load caused by an invalid request. LRP policy reduces the network load to some extent, but its effect is lower than IPSO-NPS policy when peers increase. When the number of peers is more than 500 in the network, the network load of IPSO-NPS is around 14%, lower than LRP policy and Random policy.

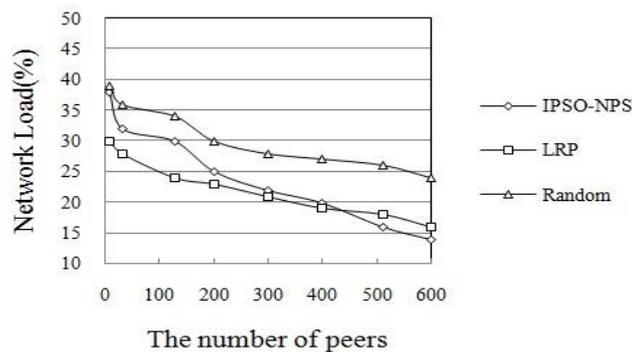


Fig. 5. The network load of system.

6. Conclusion

In order to optimize the video playback quality of BT streaming system, a new scheduling mechanism based on IPSO algorithm named IPSO-NPS is proposed. This new mechanism not only makes full use of the upload bandwidth of peers, but also selects peers whose online time is longer and the distance is closer. The IPSO algorithm is presented to improve the rate of convergence. Then, the fitness function based on the service ability of peer is designed, and the optimal peers are found through the IPSO-NPS algorithm. Experimental results show that, IPSO algorithm improves the convergence speed and overcomes the shortcoming of the original PSO algorithm. As the number of pieces increases, the startup delay of IPSO-NPS algorithm is much lower than the LRP policy and Random Strategy. When the network size expands, IPSO-NPS scheduling mechanism achieves better playing continuity and lower network load.

In the future, we will study the incentive mechanism of BT streaming system to reduce the free riding phenomena and encourage more peers to cooperate.

Acknowledgment

This work was supported in part by the Foundation of Specific Research, Science and Technology Development Center, Ministry of Education, China (No.2011109).

References

- [1] Index, C. V. N. (2013). *Cisco Visual Networking Index: Forecast and Methodology*. 2012-2017. Cisco System, USA.
- [2] Shen, S., & Li, S. (2010). P2P-based video-on-demand systems: a survey. *Chinese Journal of Computers*, 33(4), 613–624.
- [3] Pai, V., Kumar, K., Tamilmani, K., Sambamurthy, V., & Mohr, A. E. (2005). Chainsaw: Eliminating trees from overlay multicast. *Peer-to-peer Systems IV*, 127–140.
- [4] Zhang, X., Liu, J., Li, B., & Yum, T.-S. P. (2005). Cool Streaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies: Vol. 3* (pp. 2102–2111).
- [5] Zhou, Y., Chiu, D. M., & Lui, J. C. (2007). A simple model for analyzing P2P streaming protocols. *Proceedings of IEEE International Conference on Network Protocols* (pp. 226–235).
- [6] Agarwal, & Rejaie, V. R. (2005). Adaptive multisource streaming in heterogeneous peer-to-peer networks. *Proceedings of International Society for Optics and Photonics on Electronic Imaging* (pp. 13–25).
- [7] Magharei, N., Rejaie, R. & Guo, Y. (2007). Mesh or multipletree: A comparative study of live p2p streaming approaches. *Proceedings of 26th IEEE International Conference on Computer Communications* (pp. 1424–1432).
- [8] Rodrigues, C. K. D. S. (2014). Analyzing peer selection policies for BitTorrent multimedia On-Demand streaming systems in Internet. *International Journal of Computer Networks & Communications*, 6(1), 1402–2187.
- [9] Rocha, M. V., Da, S. R., & Kleber, C. On client's interactive behaviour to design peer selection policies for BitTorrent-like protocols. *International Journal of Computer Networks & Communications*, 5(5).
- [10] Liu, J. (2011). *Research and Implementation of Video Streaming System Based on BitTorrent*. Master's thesis, Zhejiang University.
- [11] Ma, Z., Xu, K., Liu, J., & Wang, H. (2012). Measurement, modeling and enhancement of BitTorrent-based VoD system. *Computer Networks*, 56(3), 1103–1117.
- [12] Rivero-Angeles, M. E., Rubino, G., Torres, I. O. O., & Martinez, L. A. (2013). Window-based streaming Videoon-Demand transmission on BitTorrent-like Peer-to-Peer networks. *Proceedings of Consumer Communications and Networking Conference* (pp. 1–6).
- [13] Dacunto, L., Chiluka, N., Vinko, T., & Sips, H. (2013). BitTorrent-like P2P approaches for VoD: A comparative study. *Computer Networks*, 57(5), 1253–1276.
- [14] Borghol, Y., Ardon, S., Carlsson, N., & Mahanti, A. (2010). Toward efficient on-demand streaming with bittorrent. *NETWORKING*, 53–66.
- [15] Kennedy, J., Eberhart, R. *et al.* (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks: Vol. 4, No. 2* (pp. 1942–1948). Perth, Australia.
- [16] Li, W., Li, Z., Li, H., Fu, B., & Li, H. (2012). Hierarchy and adaptive size particle swarm optimization algorithm for solving geometric constraint problems. *Journal of Software*, 18(4), 2567–2574.
- [17] Hu, W., & Li, Z. A simpler and more effective particle swarm optimization algorithm. *Journal of Software*, 7(11), 861–868.
- [18] Li, M. & Wu, B. (2013). An improved discrete particle swarm optimization algorithm for high-speed trains assembly sequence planning. *Assembly Automation*, 33(4), 360–373.
- [19] Cheng, Y., Ren, Y., & Tu, F. (2013). An Improved particle swarm optimization algorithm based on adaptive genetic strategy for global numerical optimal. *Journal of Software*, 8(6), 1384–1389.
- [20] Xia, X., Liu, J., & Li, Y. (2014). Particle Swarm optimization algorithm with reverse-learning and

local-learning behavior. *Journal of Software*, 9(2), 350–357.

- [21] Ji, Z., Liao, H., & Wu, Q. (2009). *Particle Swarm Optimization and Its Application*. Beijing: Science Press.
- [22] Zhang, T., & Yu, J. (2013). Improved particle swarm optimization algorithm for cloud computing task scheduling. *Computer Engineering and Applications*, 49(19), 68–72.
- [23] Laskari, E. C., Parsopoulos, K. E., & Vrahatis. M. N. (2002). Particle swarm optimization for integer programming. *Proceedings of the Conference on Evolutionary Computation* (pp. 1582–1587).
- [24] Wikimedia, Mercator projection. (2011). From http://en.wikipedia.org/wiki/Mercator_Projection
- [25] Yang, W., & Abu-Ghazaleh, N. (2005). Gps: A general peerto-peer simulator and its use for modeling bittorrent. *Proceedings of 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (pp. 425–432).



Yuqi Hu was born in Heilongjiang, China, in 1964. She received her PhD degree in School of Information Science and Engineering, North-East University, China, in 2005. Currently, she is an associate professor in School of Information Science and Engineering, Yanshan University, China. Her research interests include multimedia network, P2P network, mobile multimedia network.



Qingqing Yang was born in 1990. She received her B.S. degree in computer science and technology in 2012 at Yantai University, Yantai, China. Currently, she is a postgraduate student in School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. Her research interests include multimedia network and P2P network.