World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:1, 2015

A Fuzzy Swarm Optimized Approach for Piece Selection in Bit Torrent Like Peer to Peer Network

M. Padmavathi, R. M. Suresh

Abstract**—**Every machine plays roles of client and server simultaneously in a peer-to-peer (P2P) network. Though a P2P network has many advantages over traditional client-server models regarding efficiency and fault-tolerance, it also faces additional security threats. Users/IT administrators should be aware of risks from malicious code propagation, downloaded content legality, and P2P software's vulnerabilities. Security and preventative measures are a must to protect networks from potential sensitive information leakage and security breaches. Bit Torrent is a popular and scalable P2P file distribution mechanism which successfully distributes large files quickly and efficiently without problems for origin server. Bit Torrent achieved excellent upload utilization according to measurement studies, but it also raised many questions as regards utilization in settings, than those measuring, fairness, and Bit Torrent's mechanisms choice. This work proposed a block selection technique using Fuzzy ACO with optimal rules selected using ACO.

Keywords**—**Ant Colony Optimization (ACO), Bit Torrent, Download time, Peer-to-Peer (P2P) network, Performance.

I. Introduction

PEER-TO-PEER (P2P) is an alternative model to traditional client-server architecture. They use a decentralised model where each machine, called a peer, functions as client with own server functionality. A peer is a client and server simultaneously. In other words a peer initiates requests to other peers, simultaneously responding to incoming requests from other network peers. It is different from traditional client-server models where a client only sends requests to a server and waits for its response.

With a client-server approach, server performance deteriorates as clients requesting services from server increases. However, in P2P networks overall network performance improves when many more peers are added to network. They organise themselves into ad-hoc groups to communicate, collaborate and share bandwidth to complete tasks (e.g. file sharing). Each peer can upload/download simultaneously. New peers can join a group while old peers leave, at any time [1].

"Pure P2P networks" and "hybrid P2P networks" are the 2 classification of P2P networks. In the former, participating peers are equal with each peer being both client and server. There is no central server to control, coordinate or manage intra peer exchanges. Gnutella and Freenet are pure P2P network examples.

Padmavathi M. is with RMD Engineering College, Chennai, Tamilnadu, India (Phone: +91 9066002047; e-mail: padmavathim2000@gmail.com).
Suresh R M. is with S.M.I.T Engineering College, Chennai, Tamilnadu, India (e-mail: sureshrm.2013@rediffmail.com).

Hybrid P2P networks have a central server for "administrative" functions to facilitate P2P services. For example, server helps peers to "search for particular files and initiate a direct transfer between clients" in Napster. An available files catalogue is in the server, while actual files are scattered across network peers. Bit Torrent (BT) is an example, where central server named tracker coordinates communication among BT peers to complete downloads [2].

The main difference between both networks is that hybrid P2P networks have a central entity for specific administrative functions; there is no server in pure P2P networks. Pure P2P architecture is simpler with a higher fault tolerance level compared to hybrid P2P architecture. But, hybrid P2P architecture has reduced consumption of network resources and is scalable than pure P2P approach.

Napster, Gnutella, and Kazaa, P2P content sharing technologies are successful applications on the Internet. P2P gained public attention through Napster which supports music sharing on the Web. It is an emerging and interesting research technology having a promising product base. Intel P2P working group defined P2P as "the sharing of computer resources and services by direct exchange between systems". This gives P2P systems 2 key characteristics:

- Scalability: there is no algorithmic/technical limitation of system size, e.g. the system complexity should be constant regardless of the node's number in it.
- Reliability: malfunction on any node does not affect entire system – or even other nodes.

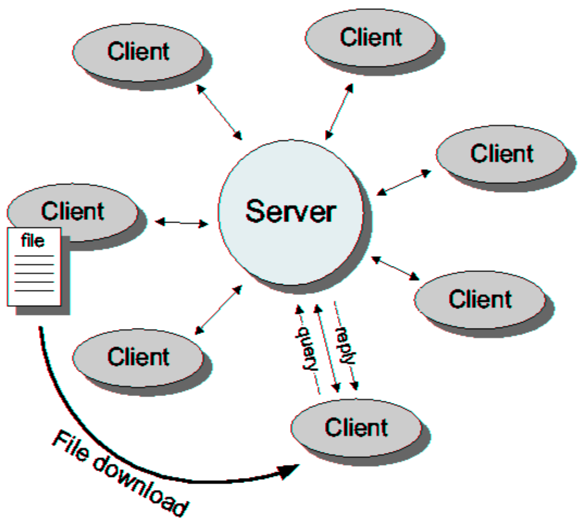

Fig. 1 A peer to peer network

File sharing networks like Gnutella are examples of scalability and reliability. Peers are connected first to a flat overlay network in Gnutella, where all peers are equal. Peers have direct links without a master-server arrangement and any malfunctioning node does not cause other system nodes also to malfunction [3].

The key to P2P system data-sharing usability is its challenging design aspect, where efficient techniques search, and retrieve data. Best system search techniques depend on application needs. For storage and archival systems focusing on availability, search techniques suit it as they guarantee content location if in existence, within bounded hops number. To achieve such properties, techniques tightly control data placement and topology in a network, currently supporting search by identifier [4].

Modeling P2P Networks

P2P overlay networks are modeled by an undirected graph $G = (V, E)$ where vertex set V represents units like hosts/routers, and edge set E represents physical links linking communicating unit pairs. And $f : V \rightarrow \{1,\ldots n\}$is a labeling of its nodes, where $n = |V|$. For instance, G can model whole/part of Internet. In an undirected graph $G = (V,E)$ modeling an interconnection network, and subset X μ V (G) of communicating units (peers), a corresponding weighted graph can be constructed $D = (V,E)$, where $V(D) = X$, and weight of each uv 2 $E(D)$ is equal to shortest path length between peer u and peer v in G. D includes connected edges and referred to as distance graph of G. A physical network G (representing Internet), starts and chooses a communicating peers set X. The resulting distance graph D is basis for constructing a P2P overlay graph $H = (V,E)$, as follows. The vertex set $V(H)$ is the same as $V(D)$, and edge set $E(H)$ μ $D(G)$. The main issue here is selecting $E(H)$. If $E = [eij]_{n \times n}$ is so that $eij= 1$ if $(i; j) \in E$, and 0 otherwise, i.e., E is incidence matrix of G, then neighbor-selection problem is to locate a rows/columns permutation bringing all non-zero elements of E into optimal possible interconnection around diagonal [5].

A P2P application, Bit Torrent enables efficient distribution of large files by leveraging downloading peers *upload* bandwidth. The basic Bit Torrent idea is to divide a large file (a few 100 MBytes long) into 256 KB size pieces each. The peers set trying to download the file connect several peers simultaneously and download different file pieces from various peers. To enable this, Bit Torrent uses a tracker, centralized software. A peer wanting to download a file first links to file tracker in a Bit Torrent network.

The tracker returns a random peers list of those with the file. The downloader establishes links to these other peers to locate what piece is with other peers [6]. The downloader requests pieces that it does not have from other peers to which it is linked. Each peer can upload to a fixed number (default is four) only at any time. In Bit Torrent, uploading is called unchoking. The peers to unchoke is determined by current downloading rate from the peers, i.e., each peer uploads to 4 peers which provide best downloading rate though it might have received requests from more than 4 downloaders. This mechanism deters free-riding.

The idea is to split the file into equal-sized *blocks* and have nodes download it from multiple peers simultaneously. Blocks are then further subdivided into *sub blocks* to ensure requests *pipelining* to mask request-response latency. Corresponding to each large downloadable file is a central component, the *tracker* which tracks system nodes. The tracker receives nodes updates periodically when they join or leave torrent. System nodes are either *seeds*, i.e., nodes with a complete file copy which are willing to serve others, or *leechers.*

Each node seeks chances to download and upload blocks to neighbours. It employs a *local rarest first (LRF)* policy in choosing blocks to download, trying to download a block least replicated among neighbours.

A *tit-for-tat (TFT)* policy guards against freeriding: a node uploads to neighbours who offer best download rates. Each node limits concurrent uploads to small number, usually 5. The mechanism which limits concurrent uploads is called *choking*, - a temporary node refusal to upload to a neighbour [7].

A peer user combines LRF policy and a random scheme for download scheduling strategy in a Bit Torrent network. This ensures uniform file pieces dissemination and prevents peers from waiting long to find last missing pieces. But, this helps only to distribute a complete file first as file parts must be obtained before it is played. Bit Torrent download scheduling strategy aims to minimize time for peers to get entire file. This fails to emphasize the need to get first pieces first. This results in the network using its resources to download later pieces when playback is interrupted as an earlier piece might not be available [8]. Also, peers may choose to use earliest-first to support real-time or uninterrupted playback. This naturally favors downloading earlier file parts first. This strategy may neglect exceptionally rare pieces in the network leading to bottlenecks in downloading.

To track which peers have what, Bit Torrent splits files to fixed size pieces, usually a quarter megabyte. Each downloader reports to all peers what piece it has. To verify data integrity, all pieces SHA1 hashes are included in torrent file and peers don’t report they own a piece till they’ve checked hash. Erasure codes were suggested as a technique which may help with file distribution, but this simpler approach has proved practical.

Peers download pieces continuously from all peers which can. They cannot download from peers they are not connected to. Also, sometimes peers do not have pieces they want or will not let them download. Choking, a strategy for peers disallowing downloading is discussed later. Other approaches to file distribution generally involved a tree structure, without using leaves uploading capacity. Simply having peers announce what they have uses up less than a tenth of a percent bandwidth overhead. It reliably uses all available upload capacity [9].

A Bit Torrent session begins with one server and the file being served and seeded. A client who downloaded the entire file also serves as seed for file until user aborts the application. The initial seeder need not be available always; the original seeder can withdraw after duplicates of all pieces exist. The result will have one less seeding host; other hosts continue session and serve new clients without problems. Having many seeds is beneficial, as they contribute to the session reliability by serving data, and not placing any load on it through downloading [10].

This paper proposes a procedure for fuzzy logic based block selection with ACO optimization. The new technique uses best features. Random piece selection and Bandwidth based piece selection ensure fairness. Network performance was determined using ACO optimization. The following sections deal with the related works, methodology and simulation results achieved.

II. Related Works

Kher et al. [11] suggested a model with 2 ranking schemes; one is network-specific and the other user-specific. The schemes used fuzzy logic to rank attributes based on various networks. The difference between them was in the dynamism offered by them. User-specific ranking criteria were flexible while network-specific scheme used fixed criteria. Network-specific scheme provided a structure to visualize networks overall performance index. User-specific scheme was adaptive as it catered to specific user's needs. Simulations showed that both schemes were light-weight, accurate, and could be easily implemented.

SemAnt, a novel ant-based multi-agent system meant for distributed query routing was presented by [12]. While ant metaphor was successfully applied to network routing in wireless and fixed networks, little was known about its applicability to query routing in distributed environments. The authors pointed out similarities and dissimilarities between data packets routing and query routing. The design of SemAnt was based on ACO meta-heuristic. For experimental evaluation, a P2P environment algorithm with real-world application scenario and its performance against the well-known k-random walker approach was compared. SemAnt benefits were that query routes were optimized according to popularity. The algorithm suited volatile environments.

Han and Xia [13] examined server selection for parallel download in overlay content-distribution networks. Node-selection produces optimal server set based on Worst-case Link Stress (WLS) criterion in a hyper cube like overlay network. Scaling to a bigger system was helped by algorithm due to its efficacy as it needs no topology collection, routing information or network measurement. In many fields, especially against random selection approach it ensured performance benefits. The congestion level at bottleneck link was first reduced, similar to improving high-throughput. Secondly, regarding many links utilized and network resources; total bandwidth usage was consumed less. Also, a low average, round-trip time to selected servers was possible. Finally, it facilitates heavy data exchange with neighbour nodes, the main objective of a content-distribution system.

Esposito et al. [14] proposed BUTorrent to enhance download time. Due to the need for global knowledge and overlay dynamics initial phase was delicate in content distribution (file sharing) scenario. This phase caused delays in attaining a steady state, and so maximized file download time which was unwise in piece dissemination. A new class of scheduling algorithms at seed based on proportional fair scheme was devised and implemented on a real file sharing client. Additionally, next to simulation results proposed file sharing client (BUTorrent) was validated as a standard file sharing protocol. Average download time was improved by 25%. Theoretical upper bounds over achievements that scheduling strategies can accomplish were provided.

Bonnel et al. [15] presented an information replication method in an unstructured P2P network. The author did not assume anything on network topology. Thematic agents moving randomly on network estimate redundancy level of specific information they dealt with. They can delete or create replicas if estimated redundancy is too high or too low. Experiments revealed that homogeneous information distribution in a distributed environment while achieving high level fault tolerance was possible.

Pouwelse et al. [16] contributed the following: first, authors add to understanding of operation of a P2P file sharing system that by its user-friendliness, quality of delivery content and performance, had right mechanisms to attract millions of users. Second, the paper's results can help in (mathematical) P2P systems modeling. For instance, it was always thought that arrival process and downloader's abort/departure processes are Poisson, something that contradicts the measurements. The author's conclusions were that a tension exists within P2P systems between availability, which improved when there were no global components, and data integrity, that benefits from centralization.

Bharambe et al. [9] introduced a study of simulation-based Bit Torrent which destroyed system's design and also estimated the effect of core mechanisms both independently and by grouping, under workloads diversity on entire system performance. Many metrics comprising file download time, peer link utilization, peers fairness, and volume of content served are focused in estimation. Bit Torrent performance was validated from results obtained that it was nearly-optimal regarding terms of download time, and uplink bandwidth utilization excluding intense conditions. The illustrations revealed that when high bandwidth peers were present, low bandwidth peers were able to download much more than they uploaded onto the network. To prevent unfairness, rate-based TFT policy is inefficient. To enhance fairness efficiently, basic tracker modification and a stricter, block-based TFT policy was introduced.

Fan et al. [17] suggested a method to offset the traditional Gossip protocol effects. A neighbour node selection algorithm was studied, and various factors that affect neighbour node selection in P2P networks were discussed. A new neighbour node selection policy was proposed, which was a fuzzy theory based reliable Gossip message transmission Protocol. Empirical simulation experiments with varied data analysis demonstrated efficiency improvement of proposed node selection strategy and the proposed algorithm's correctness in P2P networks.

Liao et al. [18] proposed a mathematical model which predicted average file download delay accurately in a heterogeneous Bit Torrent-like system. The new model was common and derived with least assumptions, needing least system information. A flexible token-based approach for Bit Torrent-like systems was proposed and applied to tradeoffs among entire system performance and fairness to excessive bandwidth users, by correctly setting parameters. To predict average file download delays in token- based system, and illustrating application of the model to determine scheme parameters that accomplish target performance and fairness for which the new model was extended.

An empirical Bit Torrent analysis was proposed by [19]. Bit Torrent measurements have 2 observations: (1) high bandwidth connection clients leave system shortly after file download, and (2) clients unable to accept incoming connections due to being behind a firewall, suffer from a highly reduced download speed. To combine features of Content Distribution Networks (CDNs) and P2P based networks, a hybrid CDN structure with P2P based streaming protocol in access network is proposed which attempts to overcome such issues by structuring streaming protocol to discourage freeloaders and incorporating a proxy based structure to avoid firewalls based performance problems. The proxy based structure enables incorporate caching, often lacking in P2P networks. This structure further reduces bandwidth usage by automatic cluster creation at the hierarchy's lowest levels.

A new hybridized model, which considers mobility characteristics of combined Bit Torrent protocol in a centralized way to enable partial mobility characteristics, was proposed by [20]. Here network clients use a distinct technique to differentiate mobile and static nodes. Mobility with file sharing is promising in today's run Anywhere, Anytime, Anything (3As) environments. The Bit Torrent file sharing protocol is rarely combined with mobility scenario framework as resources are unavailable due to dynamically changing network topology. So, mobility in P2P-oriented file sharing platforms degrades end-to-end efficiency and system performance. Parameters considered include round trip delays, diffusion process and seeding techniques targeting maximization of clustered swarm's average throughput to contain mobile peers. Partial mobility characteristics in a peer-tracker and peer-peer communication enhancement schema with partial mobility ensured an optimistic approach to attain availability and throughput response as seen in simulation results.

Neumann et al. [21] reviewed earlier results in this area introducing readers to common analysis procedures which were applied to get bounds for different ACO algorithms on pseudo-Boolean problem classes. ACO computational complexity is a rapidly growing research area and its finite-time dynamics is assessed with mathematical precision using bounds on (expected) time till an ACO algorithm locates a global optimum. The runtime bounds, which result further, clarified design issues from a theoretical perspective. Authors handle the question of replacing a current best-so-far solution by a new solution with similar quality. Later, ACO hybridization is discussed with local search and present examples where introducing local search results in tremendous speed-up and a dramatic performance loss.

III. Methodology

Fuzzy inference model has 4 basic parts: fuzzifier, Inference system, Rule base, and defuzzifier. The first step, Fuzzification takes inputs to determine the degree to which they belong to each appropriate fuzzy set through membership functions. As peers stay in system for very short periods, it is difficult to measure input variables exact values. The fuzzy inference system has 3 input variables: Quality, Popularity, and Shared file Size. It is hard to derive a mathematical formula using all three inputs for reputation calculation. So Fuzzy Logic (FL) calculates peers reputation when associated with the system [22].

Input variables are first fuzzified through membership functions. After input fuzzification, the fuzzy rule base is applied to arrive at fuzzy output. Fuzzy Inference Rule Base has varied Fuzzy Rules.

FL is a problem-solving control system methodology lending itself to systems implementation ranging from small, embedded micro-controllers to large, networked, multichannel PC and workstation-based data acquisition or control systems implementable in hardware/software or a combination. FL ensures an easy way to reach a definite conclusion based on imprecise, noisy, vague, ambiguous, or missing input information. FL's control problems approach mimics how a person makes decisions, only faster.

FL incorporates a rule-based IF X AND Y THEN Z approach to solving control problems rather than trying to model a system mathematically. FL model is empirically-based, relying on operator's experience and not their technical understanding of system. For example, instead of dealing with temperature control in terms such as "SP =500F", "T <1000F", or "210C <TEMP <220C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool process quickly)" are used. Though imprecise they yet are descriptive of what should happen. Consider what would be done if shower temperature is too cold: the water is made comfortable quickly with little trouble. FL can mimic this behavior but at high rates [23].

Fuzzy system outputs fuzzy rules. To optimize the selection of fuzzy rules, Ant Colony Optimization (ACO) is used in this work.

Proposed Binary ACO for Fuzzy Rule Selection

Ant Colony Optimization (ACO) algorithms form part of swarm intelligence, that is, research field which studies algorithms inspired by observation of swarm behaviour. Swarm intelligence algorithms comprise simple individuals which cooperate through self-organization, without any central control over swarm members [24].

ACO is part of a larger field of swarm intelligence where scientists study behavior patterns of bees, termites, ants and social insects to simulate processes. Insect swarms ability to thrive in nature and solve complex survival tasks appealed to scientists developing computer algorithms for solving similar complex problems.

ACO is based on real ants behavior possessing enhanced abilities like past actions memory and knowledge of distance to other locations. In nature, an individual ant cannot communicate or hunt for food effectively, but in a group, ants have the ability to solve complex problems and successfully locate and collect food. Ants communicate through use of pheromone, a chemical substance. An ant deposits a constant amount of pheromone when it travels enabling other ants to follow [25].

Each ant moves randomly, but when another ant locates a pheromone trail, it has to decide whether to follow it. If it does so, its own pheromone reinforces existing trail, and the increased pheromone increases probability of next ant selecting the same path. Hence, the more ants travelling on a path makes it attractive for subsequent ants which follow suit.

The original Ant System's (AS) importance is in being the prototype of many ant algorithms that collectively implements the ACO paradigm. AS already follows outline presented in previous subsection, specifying elements as follows:

Probability distribution defines probabilities $P_{i\psi}^{k}$ to be equal to 0 for infeasible moves (they are in tabu list of ant *k*;, a list containing moves infeasible for ants *k* starting from state i), otherwise they are computed through formula , where a and b are user defined parameters (0 < a,b < 1):

$$P^{k}_{i\psi} = \begin{cases} \dfrac{\tau^{\alpha}_{i\psi} + \eta^{\beta}_{i\psi}}{\sum_{(t\zeta)\notin tabu_k} \left(\tau^{\alpha}_{i\zeta} + .\eta^{\beta}_{i\zeta}\right)} & if\ (\tau\psi) \notin tabu_k \\ 0 & otherwise \end{cases}$$

In formula, $tabu_k$ is tabu list of ant *k*, while parameters α and β specify impact of trail and attractiveness, respectively. After each iteration *t* of algorithm, i.e., when all ants have finished a solution, trails are updated through the formula:

$$\tau_{\iota\psi}(\tau) = \rho\tau_{\iota\psi}(\tau - 1) + \Delta\tau_{\iota\psi}$$

where Δt_{iy} represents sum of contributions of ants that move (ιψ) to construct their solution, r, 0 < r < 1, is a user-defined parameter called *evaporation coefficient*, and Δt_{iy} represents sum of contributions of ants that used (iy) to construct their solution. The ants' contributions are proportional to quality of solutions achieved, i.e., the better a solution, the higher the trail contributions added to the moves it used [26]. In Binary ACO, the work of ants is divided and different kind of ants search the same edges and pheromone is updated. Each ant gets to choose one edge of the two.

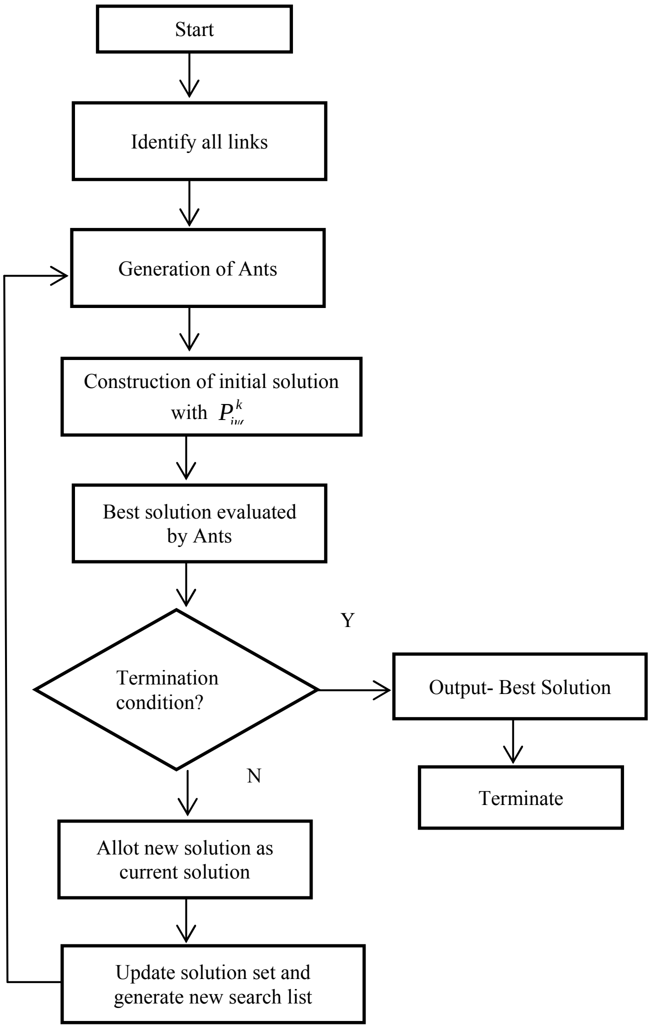

Fig. 2 Flowchart of ACO

The Framework of a Basic Algorithm [27]

input: An instance *P* of a CO problem model *P = (S, f, Ω)*.
InitializePheromoneValues(*T*)
$\mathfrak{s}_{bs}$ ←NULL
while termination conditions not met do
$\mathfrak{S}_{iter}$ ← ∅
for *j* = 1,…,$n_a$do
$\mathfrak{s}$ ←ConstructSolution(*T*)
if $\mathfrak{s}$ is a valid solution then
$\mathfrak{s}$ ←LocalSearch($\mathfrak{s}$) {optional}
if *(f (*$\mathfrak{s}$*) < f(*$\mathfrak{s}_{bs}$*))* or *(*$\mathfrak{s}_{bs}$ = NULL*)* then $\mathfrak{s}_{bs}$ ← $\mathfrak{s}$
$\mathfrak{S}_{iter}$ ← $\mathfrak{S}_{iter}$∪{$\mathfrak{s}$}
end if
end for
ApplyPheromoneUpdate(*T*, $\mathfrak{S}_{iter}$, $\mathfrak{s}_{bs}$)
end while
output: The best-so-far solution $\mathfrak{s}_{bs}$

To apply the ACO for fuzzy rule selection, the problem is represented as a graph. The graph is constructed by determining the rules defined by an antecedent combination and linking the rules to consequents. Heuristic information is then applied to select a specific consequent in each antecedent rule based on covering criteria. Initial pheromone value is obtained and solutions are built.

IV. Results and Discussion

An experiment to dynamically vary the bandwidth was also investigated. The downlink and uplink capability of each node was randomly varied from 20% to 100% of their rated capability. Experiments were conducted with ten repetitions. The performance obtained by the peer to peer network through fuzzy based ant colony optimization is tabulated in Table I.

Fig. 3 shows the mean download utilization of the network over time for fuzzy based ACO

TABLE I
Tabulation of Mean Download Utilization

Number of Nodes	Random	Fuzzy	Fuzzy with ACO
100	37.3	42.54	43.8936228
200	40.83	42.58	43.9348956
300	42.5	42.61	43.9658502
400	43.78	44.24	45.6477168
500	43.91	44.52	45.9366264
600	43.98	44.8	46.225536
700	44.14	44.85	46.277127
800	44.36	45.01	46.4422182
900	44.8	45.13	46.5660366
1000	44.8	45.22	46.6589004

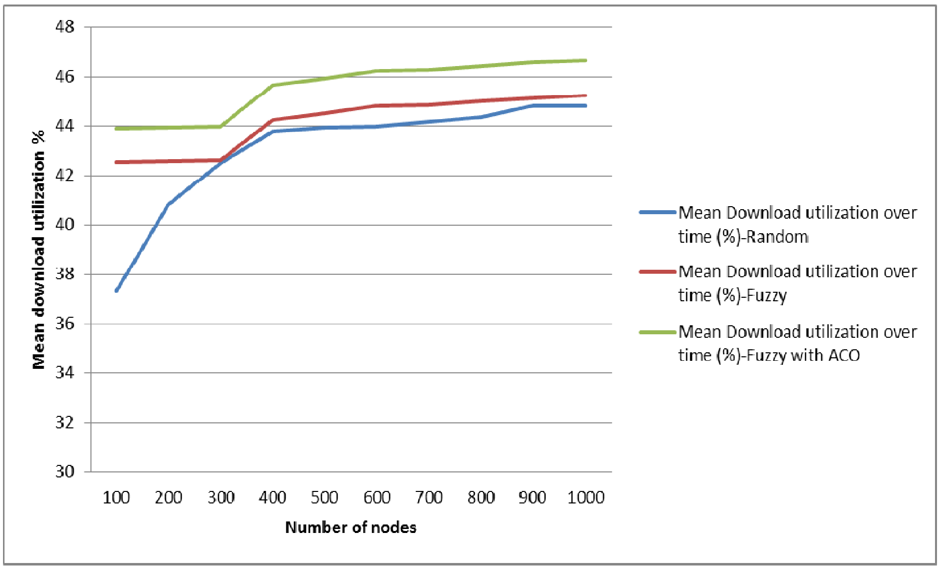

Fig. 3 Mean download utilization of the network over time for fuzzy based ACO

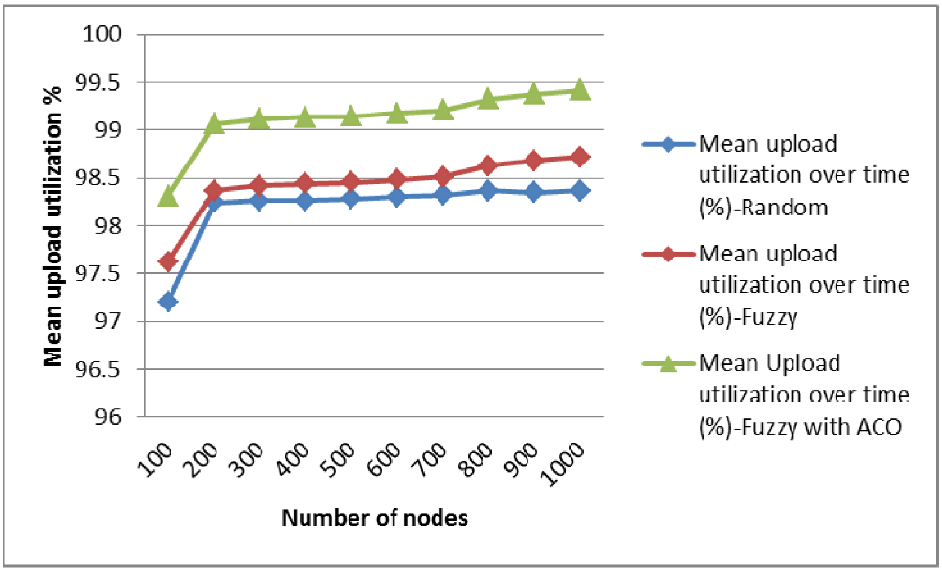

Fig. 4 Mean Upload Utilization over time using fuzzy based ACO

It is observed from Fig. 3 that the proposed Fuzzy based ACO achieves an improvement of 17.68% in mean download utilization for 100 nodes and 4.15% for 1000 nodes when compared with Random method. When compared to Fuzzy method, the proposed fuzzy based ACO achieves an average improvement of 3.18%.

TABLE II
TABULATION OF MEAN UPLOAD UTILIZATION

Number of Nodes	Random	Fuzzy	Fuzzy with ACO
100	97.2	97.62	98.31037
200	98.24	98.37	99.06567
300	98.26	98.42	99.11603
400	98.26	98.44	99.13617
500	98.28	98.45	99.14624
600	98.3	98.48	99.17645
700	98.32	98.51	99.20666
800	98.37	98.63	99.32751
900	98.35	98.68	99.37786
1000	98.37	98.72	99.41815

It is observed from Fig. 4 that the proposed Fuzzy based ACO achieves an improvement of 1.14% in mean upload utilization for 100 nodes and 1.07% for 1000 nodes when compared with Random method. When compared to Fuzzy method, the proposed fuzzy based ACO achieves an average improvement of 0.71%. The mean download and upload of a peer-peer network using fuzzy based ACO is enhanced.

V. CONCLUSION

Bit Torrent is widely used for quick and efficient distribution of large files by leveraging downloading peers upload bandwidth. Studies show that the upload utilization is good for Bit Torrent though utilization in settings and Bit Torrent's mechanisms choice can further be enhanced to improve the performance. In this paper, a novel technique for block selection based on fuzzy logic with ACO optimization is proposed. In the proposed technique, fairness is ensured by using the best features of Random piece selection and fuzzy based piece selection. The performance of the network was determined using ACO optimization. The proposed methods were evaluated by simulating a bit torrent like network and study it through various scenarios. The proposed optimization was compared with random policy and proposed fuzzy based policy to pick up blocks for downloading from neighbours. Simulations were carried out with a network having 100 nodes to 1000 nodes for varied bandwidth considerations. Results show that the proposed technique is better as it shows upload utilization to be better than random and fuzzy based policy.

REFERENCES

[1] Steinmetz, R. (2005). Peer-to-peer systems and applications (Vol. 3485). Springer-Verlag New York Incorporated.

[2] Peer-To-Peer Network. The Government of the Hong Kong Special Administrative Region.

[3] Ding, C. H., Nutanong, S., &Buyya, R. (2005). Peer-to-peer networks for content sharing. Peerto-Peer Computing: The Evolution of a Disruptive Technology, 28-65.

[4] Qiu, D., &Srikant, R. (2004). Modeling and performance analysis of BitTorrent-like peer-to-peer networks. ACM SIGCOMM Computer Communication Review,34(4), 367-378.

[5] Liu, H., Abraham, A., & Badr, Y. (2010). Neighbor selection in peer-to-peer overlay networks: a swarm intelligence approach. In Pervasive Computing (pp. 405-431). Springer London.

[6] Yang, B., & Garcia-Molina, H. (2002). Improving search in peer-to-peer networks. In Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on (pp. 5-14). IEEE.

[7] Hwang, K. W., Misra, V., & Rubenstein, D. S. (2008). Stored media streaming in bittorrent-like P2P networks.

[8] Bharambe, A. R., Herley, C., &Padmanabhan, V. N. (2005). Analyzing and improving BitTorrent performance. Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA, 98052, 2005-03.

[9] Cohen, B. (2003, June). Incentives build robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer systems (Vol. 6, pp. 68-72).

[10] Skevik, K. A., Goebel, V., &Plagemann, T. (2004). Analysis of bittorrent and its use for the design of a p2p based streaming protocol for a hybrid cdn. Delft University of Technology Parallel and Distributed Systems Report Series, Tech. Rep. Technical Report.

[11] Kher, S., Somani, A. K., & Gupta, R. (2005, October). Network selection using fuzzy logic. In Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on (pp. 876-885). IEEE.

[12] Michlmayr, E., Pany, A., & Graf, S. (2006, September). Applying ant-based multi-agent systems to query routing in distributed environments. In Intelligent Systems, 2006 3rd International IEEE Conference on (pp. 36-41). IEEE.

[13] Han, S. C., & Xia, Y. (2009). Optimal node-selection algorithm for parallel download in overlay content-distribution networks. Computer Networks, 53(9), 1480-1496.

[14] Esposito, F., Matta, I., Bera, D., &Michiardi, P. (2011). On the impact of seed scheduling in peer-to-peer networks. Computer Networks, 55(15), 3303-3317.

[15] Bonnel, N., Ménier, G., &Marteau, P. F. (2007, October). Information replication strategy in unstructured peer-to-peer networks using thematic agents. In Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on (pp. 697-702). IEEE.

[16] Pouwelse, J., Garbacki, P., Epema, D., & Sips, H. (2005). The bittorrent p2p file-sharing system: Measurements and analysis. In Peer-to-Peer Systems IV(pp. 205-216). Springer Berlin Heidelberg.

[17] Fan, X., Zhang, W., & Guan, L. A Fuzzy Algorithm of Peer Selection for P2P Streaming Media System Based on Gossip Transmission Protocol.

[18] Liao, W. C., Papadopoulos, F., &Psounis, K. (2007). Performance analysis of BitTorrent-like systems with heterogeneous users. Performance Evaluation,64(9), 876-891.

[19] Violaris, G. C., &Mavromoustakis, C. X. (2010). On the performance evaluation and analysis of the hybridised bittorrent protocol with partial mobility characteristics. arXiv preprint arXiv:1009.1708.

[20] Neumann, F., Sudholt, D., & Witt, C. Computational Complexity of Ant Colony Optimization and Its Hybridization.

[21] Mahapatra, A., &Tarasia, N. (2011). A Fuzzy Approach for Reputation Management using Voting Scheme in Bittorrent P2P Network. International Journal of Computer Science and Information Technologies, 2(2), 735-740.

[22] Mahapatra, A., &Tarasia, N. (2011). A Fuzzy Approach for Reputation Management using Voting Scheme in Bittorrent P2P Network. International Journal of Computer Science and Information Technologies, 2(2), 735-740.

[23] Kaehler, S. D. (1998). Fuzzy Logic-An Introduction. available at www. seattle robotics. org/encoder/mar98/fuz/fl_part1. html.Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. Evolutionary Computation, IEEE Transactions on, 1(1), 53-66.

[24] Blum, C. (2005). Ant colony optimization: Introduction and recent trends. Physics of Life reviews, 2(4), 353-373.

[25] Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 18(1), 41-48.

[26] Dorigo, M., Birattari, M., &Stutzle, T. (2006). Ant colony optimization. Computational Intelligence Magazine, IEEE, 1(4), 28-39.

[27] Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical computer science, 344(2), 243-278.

Padmavathi M is Associate Professor with the Department Information Technology at RMD Engineering College. She is currently pursuing his doctorate in Anna University, India.

Suresh A M is currently working as Principal in the S.M.I.T Engineering College, India.