Alpha Multipliers Breadth-First Search Technique for Resource Discovery in Unstructured Peer-to-Peer Networks

Azrul Amri Jamal[*], William John Teahan[#]

[*]*Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut, Terengganu, Malaysia*
E-mail: azrulamri@unisza.edu.my

[#]*School of Computer Science, Bangor University, Bangor, Gwynedd, United Kingdom*
E-mail: w.j.teahan@bangor.ac.uk

Abstract— **Resource discovery in unstructured peer-to-peer (P2P) networks is important in the field of grid computing. Breadth-first search (BFS) is widely used for resource discovery in unstructured P2P networks. The technique is proven to return as many search results as possible. However, the network cost of the technique is high due to the flooding of query messages that can degenerate the performance of the whole network. The objective of this study is to optimise the BFS technique, so that it will produce good search results without flooding the network with unnecessary walkers. Several resource discovery techniques used in unstructured P2P networks are discussed and categorised. P2P simulators that are used for P2P network experiments were studied in accordance with their characteristics such as, scalability, extensibility and support status. Several network topology generators were also scrutinised and selected in order to find out the most real-life like network generation model for unstructured P2P experiments. Multiple combinations of five-tuple alpha multipliers have been experimented to find out the best set to make α-BFS. In our test, the α-BFS increases the query efficiency of the conventional BFS from 55.67% to 63.15%.**

Keywords— **alpha multipliers; resource discovery; peer-to-peer networks; breadth-first search; unstructured networks**

I. INTRODUCTION

Computer networks can be classified into several topological types. In order to increase search efficiency, each network topology uses different techniques to discover resources. Resource discovery techniques are classified based on the network topology that utilises the techniques [1]. Breadth-first search (BFS) is one of the earliest resource discovery techniques, which derived from mathematical method [2]. In the BFS technique, the originator of the query sends one walker to each adjacent node and the receiving nodes will forward the query to every node adjacent to them. This action can be considered as flooding the network with search queries.

Apart from BFS, it discussed and provided pseudo codes to several resource discovery techniques in unstructured P2P networks such as Random Walk (RW), Restricted Random Walk (RRW), Intelligent BFS (Int-BFS), Depth-First Search (DFS), Adaptive Probabilistic Search (APS) and Blackboard Resource Discovery Mechanisms (BRDM) [2].

Several other unstructured P2P network resource discovery techniques are elaborated [3]. They are Napster, Gnutella, Qu et al., Masyaekhi and Habibi, Learning Automata-based Resource Discovery (LARD), Improved Adaptive Probabilistic Search (IAPS) and Discovery of Heterogeneous Multiple Compute Resources Framework (DHMCF) [4]-[8].

Napster, Gnutella, and Qu et al. resource discovery technique utilises the BFS technique in propagating the search queries. The remaining resource discovery techniques rely on controlling each query message based on information stored on the nodes such as reputable peers in Mashayekhi and Habibi, which score based in IAPS and machines' state in DHMCF.

Flooding the network with query messages is not desirable in resource discovery. The method wastes computer network resources that are often limited. Nevertheless, by flooding the network, the BFS technique is able to find the most resources in unstructured networks compared to other resource discovery techniques.

This paper will explain an uninformed resource discovery technique, Alpha Breadth-First Search (α-BFS) that consists of two new walker replication rules that significantly decreases the number of walker replications while maintaining good search results. The first rule is the implementation of alpha multipliers. The second rule is regarding the forwarding or queries based on neighbour selection.

Details regarding the two rules of α-BFS and the evaluations techniques will be discussed further in the following sections. This research is a continuation of an article journal published in International Journal of Digital Information and Wireless Communications that focuses on testing resource discovery techniques on the simulated P2P network with one million nodes [2].

A. *Resource Discovery in Unstructured P2P Networks*
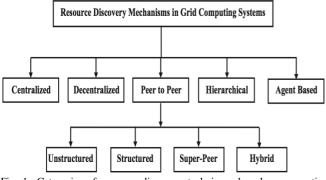

Fig. 1 Categories of resource discovery techniques based on computing systems [1]

As shown in Fig. 1, in [1] has classified resource discovery mechanisms into 5 parts namely centralised, decentralised, P2P, hierarchical and agent-based. Resource discovery for P2P is further classified as unstructured, structured, super-peer and hybrid. Unstructured P2P enables direct exchange and sharing of resources without even going through a central web server [9]. Thus, the networks are reliable in terms of query correctness and single point of failure and can tolerate node dynamicity [1].

Despite that, the complexity of resource discovery algorithms for unstructured P2P networks is around $O(N^2)$ where N is the number of nodes in the network. Furthermore, the time complexity for the unstructured network has a high order of growth compared to the scale of the network [1].

Among resource discovery techniques, BFS is the most used resource discovery technique for unstructured P2P networks being used by Napster, Gnutella, Int-BFS [2] and Learning Automata-based Resource Discovery (LARD). Napster and Gnutella use BFS that floods the network with queries. Int-BFS [2] floods the system during its initial stage but will send fewer queries once the network has learned from its previous queries. LARD uses BFS, but focus on the routing of the unstructured P2P network in order to minimise the effect of the query flooding [6].

A technique is developed to search for shared files over unstructured P2P networks. The main characteristic of the developed technique is that it is fully dynamic and divided the peers into groups, thus delivering a high load balancing and high fault tolerance technique [4]. Mashayekhi and Habibi developed a robust trust based resource discovery technique for the unstructured P2P network. The technique introduced uses semantic approach to determine the trust level of neighbouring peers [5]. Mashayekhi and Habibi's technique has an issue of fault positive error [3].

LARD is a technique for grid computing resource discovery in an unstructured P2P network. The technique utilises automation to find the shortest path to the target. In the initial state, the automaton chooses link randomly. It will

then check the path length and rewards the path if it is shorter than the average length, the path will be penalised otherwise [6]. LARD eases the problem of network flooding. However, by using TTL like Mashayekhi and Habibi's technique, LARD also has an issue of fault positive error [3].

Improved Adaptive Probabilistic Search (IAPS) utilises ant-colony optimisation to search space and search overhead [7]. The technique is proven to be better than popular random walk and Adaptive Probabilistic Search (APS) [5]. DHMCF is a resource discovery technique for a pure unstructured P2P network that responds to dynamic requests [8]. In this technique, there are 5 units to gather information, make decisions, find resources and balance the load. This technique, however, suffers from low robustness and security [3].

All recent resource discovery techniques can be classified as knowledge-based resource discovery techniques [4]-[8]. These techniques have a learning effect that the percentage of successful searches will increase the longer the technique is running in the network. This paper will discuss another approach to increase the percentage of successful search only by controlling the query message forwarding. The approach will be discussed further in the alpha breadth-first search section.

The flooding technique increases the number of walkers exponentially and causes a huge search overhead. Formally, for unstructured networks, the number of neighbours a node has is represented by k and the message number of hops as Time-To-Live (TTL). The equation for number of query message forwarding (F_{BFS}) can be written as follows [1]

$$F_{BFS} = k^{TTL} \qquad (1)$$

A peer-to-peer node usually has more than 20 neighbours. Therefore, having a 5-hop BFS resource discovery mechanism would forward 20^5 query messages or 3.2 million times. A 10-hop BFS in the network would replicate walkers up to 20^{10} or 1.024×10^{13} times. These examples are for networks where each node connects on average with other 20 nodes. In some P2P networks, each node may connect to hundreds of other nodes. In this case, the amount of walker replication is very large even when considering that the search query was initiated by only one node. In a real-world network application, there will be many peers initiating queries. Therefore, implementing a search technique that floods the network will certainly degenerate the network performance and might even collapse the whole P2P network.

B. *Informed and Uninformed Resource Discovery*

Resource discovery techniques have been classified into two main categories namely blind search and knowledge based [10]. Search techniques have been categorised into various categories such as informed and uninformed searches, adversarial, optimisation and evolutionary searches [11]. Concerning the first two categories, it has been named as blind search and knowledge based [10].

C. *Uninformed Resource Discovery*

Uninformed or blind search is a category of resource discovery techniques, where the walkers that bring queries

do not have any information regarding the network throughout the whole search. The queries will also not be processing any information of the network for their future searches. The query message sent using uninformed search is lightweight to the network and the nodes. The query message can be easily replicated or cloned, and distributed according to its forwarding rules behind the resource discovery techniques that are being used.

Despite the query messages being lightweight, the techniques in this category usually rely on the number of query messages in order to find the resources that they want to find. There are occurrences where the techniques send the same query messages to a node in the P2P network multiple times. This condition is called flooding the network. No matter how lightweight the query message, the number of queries will consume a lot of resources along the way.

Uninformed search resource discovery techniques are usually used in mobile networks or ad-hoc networks, where the computational or storage resources on each node are small or limited. Among search techniques that fall into this category are BFS, uniform cost, DFS, depth-limited, iterative deepening and bi-directional [11].

D. Informed Resource Discovery

Informed or knowledge-based search is a category of resource discovery techniques, where the techniques utilise some heuristic approach towards finding the resource. Resource discovery techniques that fall within the informed search category take into account the information of the network that the techniques have been working on. Techniques within the informed search category usually generate some information for their own technique's future references.

TABLE I
CHARACTERISTICS OF RESOURCE DISCOVERY TECHNIQUES

Techniques	Query Replication	Type of Search
RW [2]	Single	Uninformed
RRW [2]	Single	Uninformed
BFS [2]	Multiple	Uninformed
Int-BFS [2]	Multiple	Informed
DFS [2]	Single	Uninformed
APS [2]	Single	Informed
BRDM [12]	Multiple	Informed
Qu et al. [4]	Multiple	Informed
Mashayeki and Habibi [5]	Multiple	Informed
LARD [6]	Single	Informed
IAPS [7]	Single	Informed
DHMCF [8]	Multiple	Informed

The query messages of knowledge based resource discovery techniques often are not lightweight because it contains a lot of information regarding the network. Upon arriving at a new node, several checks need to be done by the node and sometimes the nodes need to be compared with the information that they already have. After the comparison, the information that the query messages initially brought is updated with the information that the current node has. Further processing by the node needs to be done in order to decide on where to forward the query messages to.

Informed search resource discovery techniques would usually require some computation and/or storage resources from the nodes. Together with good computation and storage management and intelligent decision making, informed search resource discovery techniques can reduce their consumption of network communication resources and subsequently the computational resources. The longer that the informed search resource discovery techniques run on a P2P network, the more they learn about the network. This will lead to a better resource discovery and this phenomenon can be considered to be a "learning effect".

Among resource discovery techniques that fall into this category are best-first search, greedy search, and A* search. All search techniques in this category utilise some kind of heuristic approach in order to select among all the alternatives that they have. Informed search strategies use problem-specific knowledge to their advantages according to the goal of the search. For example, the greedy search expands nodes closest to the goal and the A* search expands nodes on the least-cost solution path [11].

The blackboard resource discovery mechanism (BRDM) uses a different kind of informed search [12]. Rather than using a heuristic in the traditional sense, instead, the BRDM utilises the knowledge it has obtained through its past behaviour. This reference to knowledge obtained from past behaviour is also present in other resource discovery techniques such as the intelligent-BFS (Int-BFS) and adaptive probabilistic search (APS) [2].

Informed search resource discovery techniques are usually used in large and complex interconnected networks, where the nodes in the network have many computation and storage resources to spare for the maintaining of the network. Table I1shows the categorisation of resource discovery techniques based on the query replication characteristics and the type of search of either informed or uninformed.

E. Selecting P2P Network Simulator and Topology Generator

Several network simulators have been used for P2P researches. Among the simulators are 3LS, General Peer-to-Peer Simulator (GPS), Neurogrid, P2PSim, PeerSim, PeerThing, Query Cycle and RealPeer. The paper showed the results for several characteristics that are important in P2P researches, however, it did not include the cumulative marks for all of the simulators [13].

Nevertheless, combining the P2P simulator marks for each characteristic shows the most suitable simulator to be used for P2P network research. Table 2 shows the combination marks for all P2P simulators [13]. Each characteristic is marked for a minimum of 0 and maximum of 5 marks. The characteristics: usability and documentation; scalability; extensibility; runtime, status, and GUI are marked with (a), (b), (c) and (d) respectively.

The key characteristic of the research is the scalability of the simulator, followed by the extensibility and support status of the simulator [2]. Based on Table 2, it is decided that the most suitable simulator to be used for this research is PeerSim. PeerSim has the best scalability feature, it is extensible and still being actively supported by its developers and P2P simulator researchers [13].

PeerSim is developed by Montresor and Jelasity, in order to simulate peer-to-peer networks. It has two simulation engines, cycle driven and event driven. The latter can closely simulate the network, however, the number of nodes that it can simulate is limited. The former simulates the network in the manner of cycles. The cycle driven engine enables PeerSim to simulate a large number of nodes for P2P experiments [2].

<div align="center">
TABLE III
P2P SIMULATORS CUMULATIVE CHARACTERISTICS
</div>

Simulator	(a)	(b)	(c)	(d)	Overall Marks
PeerSim	4	5	3	4	16
Neurogrid	5	4	3	3	15
PeerThing	4	1	4	5	14
Query Cycle	2	5	2	4	13
RealPeer	4	2	3	2	11
P2PSim	1	1	2	4	8
GPS	1	1	2	4	8
3LS	0	0	1	3	4

PeerSim has several network topology generators such as Heuristically Optimised Trade-offs (HOT), Regular Rooted Tree (RRT), Star, Ring Lattice (RL), Watts-Strogatz, Scale Free Barabási-Albert (BA), Scale Free Dorogovtsev-Mendes (DM) and K-out topology generator. Selection of the topology generator is very important because it is important to have a network-like topology [14].

Many real-world scenarios follow the power law distribution. The power law is a functional relationship between two quantities, where one of the quantities varies as a power of another. In the example of P2P networks, the two quantities are the number of the nodes and the number of neighbours that each node has [15]. There will be many nodes that have a small number of neighbours, and there will be a small number of nodes that have many neighbours.

<div align="center">
TABLE IIIII
SUMMARY OF NETWORK TOPOLOGY GENERATOR
</div>

Generator	Scale-Free	Power Law	Structured
BA	Yes	Yes	No
DM	Yes	Yes	No
HOT	No	Yes	Yes
K-out	Yes	No	No
RL	No	No	Yes
RRT	No	No	Yes
Star	Yes	No	Yes
WS	No	Yes	No

Another important characteristic of network topology generator is being scale-free. A scale-free network topology would enable the number of nodes in the network being simulated to be increased or decreased at anytime of the simulation. BFS resource discovery technique is widely used in unstructured P2P networks [2]. Therefore, in order to imitate real-life unstructured network topology for the simulation, the network should follow the power law, scale-free and unstructured.

Fig. 2 shows the generated topology using HOT generator model. It shows a single point of failure in the middle of the topology. An unstructured network topology looks like Fig. 3 that are generated using scale free Barabási-Albert. Unstructured networks do not have a single point of failure and the nodes in the network can connect to any other nodes even though they are located far apart from each other.
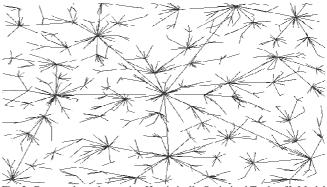

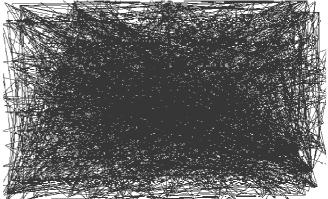Fig. 2 Generated topology using Heuristically Optimised Trade-offs Model

Fig. 3 Generated topology using Scale Free Barabási-Albert Model

Table 3 shows the summary of network topology generators' characteristics. HOT, RL, RRT and Star are structured network topology with a single point of failure. K-out, RL, RRT, and Star do not follow the power law. HOT, RL, RRT and WS are not scaled free. Therefore, it is not possible to add or subtract nodes from the initially generated topology.

Among all of the network topology generator models, only the scale-free Barabási-Albert and scale-free Dorogovtsev-Mendes models that fulfil all the requirements to run experiments on BFS. The BA and DM topology generators were then investigated based on the number of nodes and the number of neighbours that each node connected with. Fig. 4 and Fig. 5 show the graph of a number of nodes against a number of neighbours generated by BA and DM respectively.

Both BA and DM network topology generators show the power law characteristics. Nonetheless, there is an outlier in the DM's graph. Even though having one outlier might be permissible because the graph is in logarithmic scale, the outlier means that the difference is very big. This problem might reoccur when running the network generator in a bigger scale. Therefore, henceforth only the Scale Free Barabási-Albert will be used for the experiments.
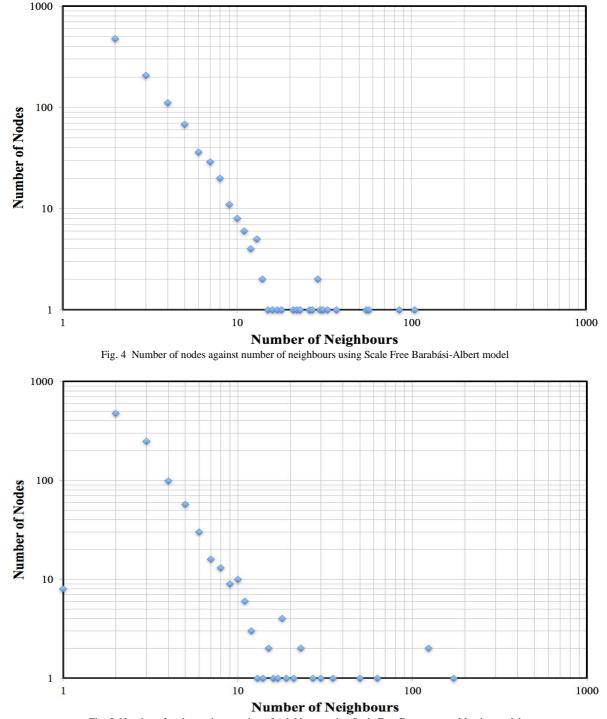
Fig. 4 Number of nodes against number of neighbours using Scale Free Barabási-Albert model

Fig. 5 Number of nodes against number of neighbours using Scale Free Dorogovtsev-Mendes model

F. Resource Discovery Query Efficiency

The main problem in information gathering is to get the right information with proper quality, reliability, and timeliness [18]. Two methods to assess resource discovery technique effectiveness namely search cost and total cost have been outlined. Search cost is the time complexity for the technique to find the solution. Total cost is the combination of the search cost and the path cost of the solution found. The path cost to the solution can also be considered as the number of message being forwarded and returned in the network [11].

Metrics to calculate the effectiveness of a search algorithm has been outlined. The efficient algorithm should not generate unnecessary messages and queries that were generated should have a high probability to find the target. The metrics proposed by the research is Query Efficiency (QE). Hence, QE will be represented as η (common Greek letter to show efficiency), the equation for η is as follows [16].

$$\eta = \frac{QueryHits}{\left(\frac{QueryMessages}{N}\right)}$$

(2)

$$= \frac{QueryHits}{MessagesPerNode}$$

where N is the number of nodes in the simulation. The query efficiency introduced is suitable for power-law P2P networks [16].

G. Alpha Breadth-First Search

The idea of alpha breadth-first search (α-BFS) is to contain the message flooding to an acceptable level while maintaining the same number of successful searches. Two approaches were taken in order to achieve this goal. The first approach is by implementing alpha multipliers; these changes according to the number of hops the query message has done. The second approach is to control the neighbour selection so that the message forwarding does not consider the neighbours that have already seen the message.

H. Alpha Multipliers

Alpha multipliers are a set of multipliers that dictate how many replications a query message can make of itself. The number of replications is based on two variables at each stage of the message forwarding. They are the number of neighbours that the node x has $L(n_x)$ and the alpha multipliers that are based on the number of hops that the query message has taken so far.

α_{hops} is a real number ranging from 0.0 to 1.0 and $hops$ is the number of hops that the query message will have α_{hops} can be summarised as follows:

$$\left(\alpha_{hops} \mid 0.0 \leq \alpha_x \leq 1.0 \ \& \ hops = 1,2,3,4,5,\dots \right) \quad (3)$$

For example, the value for the first until the fifth hops' alpha multipliers can be set as follows: $\alpha_1 = 1.0$, $\alpha_2 = 0.8$, $\alpha_3 = 0.6$, $\alpha_4 = 0.4$ and $\alpha_5 = 0.2$. A number of query message forwarding for each node is equal to the number of current alpha multiplied by the number of neighbours that the node has (e.g. the number of forwards for the first hop of origin node is $\alpha_1 \cdot L(n_o)$.

There are possibilities that the number of forwards fell below 1 (eg. number of adjacent neighbour is 2 and the current alpha multiplier is 0.4). In order to maintain the continuation of the search, the number of forwards will be reset to 1. If not, the query message may finish earlier than it should have been eliminating the chance to find the resource needed.

Let n_o be the node where the query messages originate from. The number of query message that are forwarded $F_{\alpha-BFS}$ for TTL of 5 and above in $\alpha - BFS$ are as follows

$$F_{\alpha-BFS} = \alpha_5 \cdot L\left(\alpha_4 \cdot L\left(\alpha_3 \cdot L\left(\alpha_2 \cdot L(\alpha_1 \cdot L(n_o)) \right) \right) \right) \quad (4)$$

Using the above mentioned set of alpha multiplier values on networks with 20 neighbours per node, say then the messages sent is reduced from 3.2 million messages to just 122,880 messages only as shown in following calculations:

$$F_{\alpha-BFS} = (1.0 \times 20) \times (0.8 \times 20) \times (0.6 \times 20) \times (0.4 \times 20) \times ($$

$$= 122,880$$

Pseudocode for the F calculation is shown in Fig. 6.

```
01
02 n = number of neighbours;
03 alpha[5] = [1.0,0.8,0.6,0.4,0.2];
04 hops = number of hops;
05
06 F = n * alpha[hops-1];
07 round QF to nearest integer;
08
09 if (F is less than 1) {
10  F is set to 1;
11 }
12
```
Fig. 6 Pseudocode for $F_{\alpha-BFS}$ value

I. RRW with Null Exception

PeerSim has two types of neighbour selection for the purpose of query forwarding [2], they are RW and RRW. Both RW and RRW decide to forward or replicate any query message randomly. The only differences are that RRW uses a method named selectFreeNeighbor that will forward to one of the free neighbours that is neighbour nodes that did not receive the query message earlier.

Nonetheless, if there is no free neighbour available, it will still select and return one non-free neighbour to forward the message. We consider that the query message forwarding to a non-free neighbour is unnecessary and a waste of network resource. Algorithm in Fig. 6 depicts the pseudocode of message forwarding that is being used by RRW.

The message forwarding method that is being used by RRW returns an ID of a neighbour (neighbourID), even though there is no free neighbour available. We have altered the message forwarding selection method to only return a free neighbour's neighbourID. The return value is set to null if there is no free neighbour available. Once the search protocol received the null value, no message forwarding will be done. The query message will stop on that node. The new RRW with the null exception is run after the calculation of $F_{\alpha-BFS}$. Therefore, it will overwrite the outcome of the $F_{\alpha-BFS}$ if there are no free neighbours available.

II. MATERIAL AND METHOD

The experimental setup for the experiments can be divided into two parts namely the topology setup, and the query behaviour setup. The former setup focuses on the topology of the simulated network being experimented on. The setup focuses on the generation of the topology such as the type of topology generator, the direction of connections, variables, and random seed. The latter setup focuses on setup affect the behaviours of the query such as alpha multipliers, query forwarding, and replication, the number of initial query and time-to-live.

A. Topology Setup

The BFS and α-BFS techniques have been tested according to these parameters: one million nodes distributed

and wired using the scale free Barabási-Albert model; undirected connection; k variable of two and is run of 20 cycles. The experimental setup parameters are shown in Table 4.
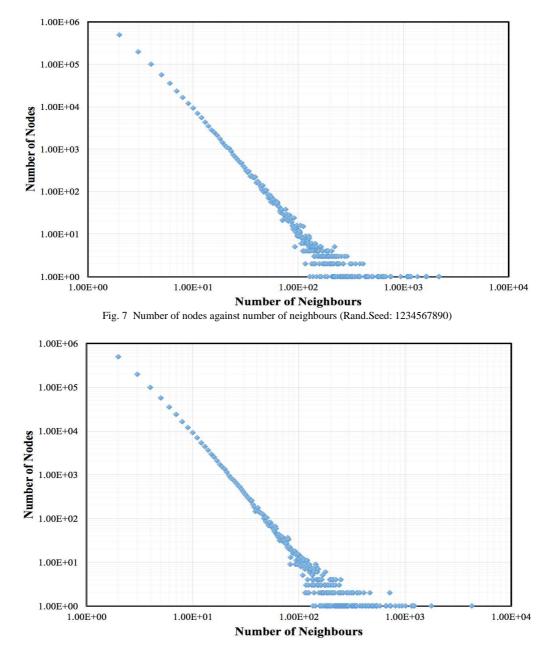
Parameters	Value
Topology generator	Scale free Barabási-Albert
No. of nodes	1,000,000
No. of initial query	1
No. of cycle run	20
k	2
Random seeds	1234567890, 1415926535 and 8979323846
Time To Live (TTL)	5, 10 and 20

The experiments were done using three different random seed in order to get multiple results using the listed sets of alpha multipliers. The first random seed is 1234567890, a standard seed being used in PeerSim simulations. The second and third random seed is the first 10 and the following 10 decimal places of π respectively. The value of 22/7 up until the 20th decimal places is 3.14159265358979323846. Therefore, the value of the second and third random seeds are 1415926535 and 8979323846 respectively.

Fig. 7, Fig. 8 and Fig. 9 show the distribution logarithmic graph of the number of neighbours against the number of nodes in the generated topology using 1234567890, 1415926535 and 8979323846 as the random seed respectively. As expected, the graph obeys the power law and does not show any outliers in the data. A number of neighbours of nodes have a mean ($\bar{x}$) of 3.999992 across all three random seeds. A number of neighbours' standard deviation (σ) for the random seed of 1234567890, 1415926535 and 8979323846 are 8.4006762, 8.6408129 and 8.2416443 respectively.

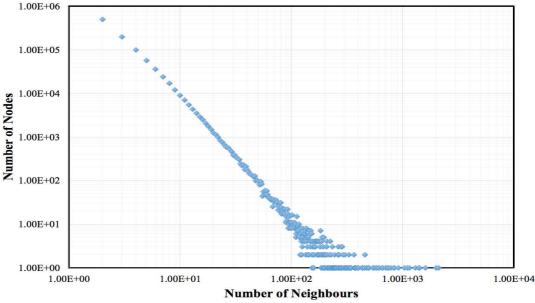Fig. 7 Number of nodes against number of neighbours (Rand.Seed: 1234567890)

1409

Fig. 9 Number of nodes against number of neighbours (Rand.Seed: 8979323846)

B. Query Behaviour Setup

The alpha multipliers are a set of five-tuple numbers that act as multipliers to find the number of query message replication needed on each step of the search. The five-tuple numbers can be of any combination of numbers ranged from zero to one. Let the alpha multipliers be numbers with one decimal place, the five-tuple numbers can have $_{10}C_5$, 252 combinations. Several patterns or orders of alpha multipliers such as fixed numbers, ascending order, descending order, division and logarithmic has been chosen for the experiments. The numbers and their patterns are as shown in Table 5. Fixed numbers pattern is where the numbers are all the same from α_1 to α_5. Ascending is when the value of α_1 keeps on increasing until α_5. Ascending pattern means that smaller number of queries are forwarded nearer to the originator and the query forwarding increases when away from the originator. Descending pattern is the exact opposite of ascending pattern.

| O | Fixed | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |

Division pattern is when the alpha multipliers are half of the previous alpha multipliers. This results with five-tuples that keeps decreasing as the number of hops increases. The division pattern is inspired with the six degrees of separation studies [17]. The alpha multipliers in this pattern can also be summarised as follows

$$\alpha_x = 2^{-x+1} \tag{5}$$

The log10 pattern is when the number of alpha multipliers is decided with the decreasing number of 10 base log ($\log_{10}10 = 1$, $\log_{10}8 = 0.9030$, $\log_{10}6 = 0.7782$, $\log_{10}4 = 0.6021$, $\log_{10}2 = 0.3010$). All experiments started with one initial query.

III. RESULTS AND DISCUSSION

The experiments are repeated three times with the change of the TTL parameter. The TTL is set to 5, 10 and 20. Query efficiency is calculated using Equation (1). The maximum number of successful searches (the number of successful searches when all queries finished their TTL) for each iteration of the experiments are also being recorded.

In order to find the best combination of five-tuple alpha multiplier values for unstructured P2P networks, both query efficiency (η) and the maximum number of successful searches importance are weighted the same. All of the values of query efficiency and the maximum successful searches are converted into a percentage by dividing it by the maximum value of the parameter. For example, the maximum value of query efficiency for the random seed of 1234567890 is 220,228. Therefore, all query efficiency for that set of experiments is divided by 220,228 and multiplied by 100%.

After converting query efficiency and maximum successful searches into a percentage, the values are added

TABLE V
ALPHA MULTIPLIERS' PATTERNS AND VALUES

Set	Pattern	α_1	α_2	α_3	α_4	α_5
A	Fixed (BFS)	1.0	1.0	1.0	1.0	1.0
B	Descending	1.0	0.8	0.6	0.4	0.2
C	$\log_{10}$	1.0	0.90	0.778	0.602	0.301
D	Descending	1.0	0.8	0.4	0.2	0.1
E	Division	1.0	0.5	0.25	0.125	0.062
F	Descending	0.8	0.5	0.3	0.1	0.0
G	Division	0.8	0.4	0.2	0.1	0.05
H	Descending	0.5	0.4	0.3	0.0	0.0
I	Division	0.5	0.25	0.125	0.062	0.031
J	Ascending	0.2	0.4	0.6	0.8	1.0
K	Fixed	0.8	0.8	0.8	0.8	0.8
L	Fixed	0.6	0.6	0.6	0.6	0.6
M	Fixed	0.5	0.5	0.5	0.5	0.5
N	Fixed	0.4	0.4	0.4	0.4	0.4

and the mean is calculated. It is found that all the top results are with the TTL of 20. The top three alpha multipliers that produced the best mean of query efficiency and maximum successful searches are Set B = {1.0, 0.8, 0.6, 0.4, 0.2}, Set M = {0.5, 0.5, 0.5, 0.5, 0.5}, and Set N = {0.4, 0.4, 0.4, 0.4, 0.4} with combined efficiency of 63.15%, 62.78% and 62.23% respectively. The graph for α-BFS for all of the alpha multipliers set (Set A to Set N) with TTL of 20 is shown in Fig. 10.

From the experiments, it is known that the BFS technique that floods the network has the maximum number of successful searches. Nevertheless, the technique has the worst query efficiency compared to any other alpha multipliers combination. BFS (Set A) resource discovery technique has a combined efficiency results of 55.67%. It is observed that the best combination of alpha multipliers, Set B has increased the combined efficiency of BFS by 7.48%.
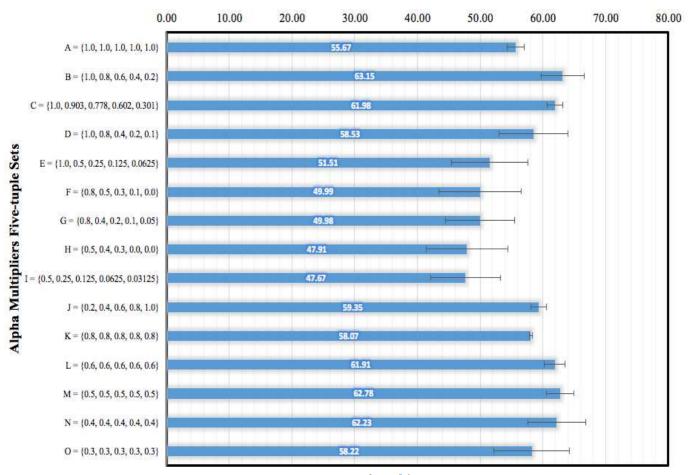

Fig. 10 Combined query efficiency $(\eta + \varpi)/2$ for α-BFS (%)

IV. CONCLUSION

BFS was among the first resource discovery mechanism that was derived from the mathematical model. The model is still being used for resource discovery in unstructured P2P networks. By implementing the alpha multipliers, the α-BFS was able to increase the combined efficiency of BFS from 55.67% to 63.15%. It can be concluded that the α-BFS was able to reduce the query messages being sent in the network and having high successful searches at the same time.

REFERENCES

[1] N. J. Navimipour, A. M. Rahmani, A. H. Navin, and M. Hosseinzadeh, "Resource discovery mechanisms in grid systems: A survey," *Journal of Network and Computer Applications*, vol. 41, pp. 389-410, May 2014.

[2] A. A. Jamal, W. S. W. Awang, M. F. A. Kadir, A. A. Aziz, and W. J. Teahan, "Implementation of resource discovery mechanisms on PeerSim: Enabling up to one million nodes P2P simulation," *International Journal of Digital Information and Wireless Communications*, vol. 5, pp. 14-20, 2015.

[3] N. J. Navimipour and F. S. Milani, "A comprehensive study of the resource discovery techniques in peer-to-peer networks," *Peer-to-Peer Networking and Applications*, vol. 8, pp. 474-492, May 2015.

[4] W. Qu, W. Zhou, and M. Kitsuregawa, "Sharable file searching in unstructured peer-to-peer systems," *Journal of Supercomputing*, vol. 51, pp. 149-166, Feb. 2010.

[5] H. Mashayekhi and J. Habibi, "Combining search and trust models in unstructured peer-to-peer networks,' *Journal of Supercomputing*, vol. 53, pp. 66-85, Jul. 2010.

[6] J. A. Torkestani, "A distributed resource discovery algorithm for P2P grids,' *Network and Computer Applications*, vol. 35, pp. 2028-2036, Nov. 2012.

[7] M. Shojafar, J. H. Abawajy, Z. Delkhah, A. Ahmadi, Z. Pooranian, and A. Abraham, "An efficient and distributed file search in unstructured peer-to-peer networks," *Peer-to-Peer Networking and Applications*, vol. 8, pp. 120-136, Jan. 2015.

[8]   S. L. Mirtaheri and M. Sharifi, "DHMCF: An efficient resource discovery framework for pure unstructured peer-to-peer systems," *Computer Networks*, vol. 59, pp. 213-226, Feb. 2014.

[9]   F. Herzallah and M. Mukhtar, "The impact of perceived usefulness, ease of use and trust on managers' acceptance of e-commerce services in small and medium-sized enterprises (SMEs) in Palestine," *International Journal on Advanced Science Engineering Information Technology*, vol. 6, pp. 922-929, 2016.

[10]   Y. P. Singh, R. Rathi, J. Gajrani, and V. Jain, "Two levels TTL for unstructured P2P network using adaptive probabilistic search," *International Journal of Scientific and Engineering Research*, vol. 3, pp. 1-4, 2012.

[11]   M. T. Jones, *Artificial Intelligence: A Systems Approach*, Massachusetts, USA: Jones and Bartlett Publishers, 2015.

[12]   N. A. Al-Dmour and M. S. Saraireh, "ParCop with new capabilities and efficient scheduling policies," *Leonardo Journal of Sciences*, vol. 18, pp. 11-26, Jan. 2011.

[13]   A. Basu, S. Fleming, J. Stanier, S. Naicken, I. Wakeman, and V. K. Gurbani, "The state of peer-to-peer network simulators," *ACM Computing Surveys,* vol. 45, pp. 1-25, Aug. 2013.

[14]   K. Karaoglanoglou and H. Karatza, "Resource discovery in a grid system: Directing requests to trustworthy virtual organizations based on global trust values," *Journal of Systems and Software*, vol. 84, pp. 465-478, Mar. 2011.

[15]   L. A. Adamic and B. A. Huberman, "Power-law distributions of the World Wide Web," *Science Magazine*, vol. 287, pp. 2115, Mar. 2000.

[16]   T. Lin and H. Wang, "Search performance analysis in peer-to-peer networks," in *Proc. ICPPC'03*, 2003, p. 204.

[17]   J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. ACM STC'00*, 2000, p. 163.

[18]   M. Alqudah, Y. Yusof, S. A. M. Noah, and A. Almabhouh, "Incorporating prioritized user preferences in search system," *International Journal on Advanced Science Engineering and Information Technology*, vol. 2, pp. 61-64, 2012.