Asian Journal of Information Technology 15 (10): 1579-1585, 2016
ISSN: 1682-3915

ISM: Intelligent Slime Mold Optimization Algorithm for Proficient Web Services Discovery

[1]A. Amirthasaravanan, [2]Paul Rodrigues and [3]R. Sudhesh
[1]Department of Information Technology, University College of Engineering, Villupuram,
[2]DMI Engineering College, Chennai,
[3]Department of Mathematics, University College of Engineering, Villupuram, Tamil Nadu,India

Abstract: In recent years, web realm has Service-Oriented Architectures (SOA) as the primary standard for IT infrastructures. Indeed with the commencement of service oriented architecture, web services have gained absurd progression. Web service discovery has become progressively more substantial. Discovering the most suitable web service from huge collection of web services is very crucial for the successful implementation of the web applications. In web service discovery automation, during matching, Quality of Service (QoS) attributes are continuously need to deem. Aiming at this, Bio-inspired algorithms for semantic web services were initiated. Bio-inspired algorithm is a meta heuristics method that imitates the nature in order to unravel optimization problem and evaluates the analysis of some popular bio-inspired optimization algorithm systematically. This paper proposes a new bio inspired algorithm which is inspired from Slime Mold. Slime mold is a fungus. Slime molds are habitually observed when they form large colonies on mulch around trees or shrubs. They may initially appear as a slimy mound or crowd. Slime mold has a unique intelligence to find the pray and it's often unsighted. The character of slime mold is inspired and it solves the problem of complex and composite web services and produces optimized web services.

Key words: Service-oriented architectures, quality of service, bio-inspired algorithm, metaheuristics, slime mold

INTRODUCTION

In recent scenarios, almost all business process functionalities have been implemented with the help of web services. Thus web services have an impressive vision on business process. In industries, several applications are built by accessing various web services that are available on the internet. These applications are highly dependent on discovery and composition of correct and efficient web to the user's request. The discovery of web services represents the process allowing the localization of documents describing a web service. And also it comprises the sematic match between the descriptions of the requested services and the published services. If the web services are not discovered effectively, then they are indecisive. So, web service discovery is the most vital mission in the Web service model. Chaiyakul *et al*. (2006) and Dorigo and Stutzle (2004) Web service discovery has two challenges when facing its realism. They are like-proficient location of the Web service registries which containing the requested Web services and effective retrieval of the requested services from these registries with high Quality of Service (QoS) (Rekaby, 2013). Service composition focuses on models and methods sustaining composition of several existing web services into more complex processes which provide new functionalities. However, with the permanent proliferation in the number of web services and registries, the operations of discovery and composition become difficult (Yang and Gandomi, 2012). In fact, web has many web services, each having its own limited functionality. In several cases, a single service is not ample to respond to the user's request and habitually service composition is required. That is, the web services needs to be combined to succeed a specific goal. For example, if a person wants to travel, it is not ample to book a flight but user should also take care of reserving a hotel, rent a taxi and so on. Such composition is carried out manually today, it means that the user needs to execute all these services one by one and these tasks can be time and effort consuming.

In order to overcome such complex and composite web services we introduce a new bio inspired algorithm called slime mold (Kobayashi *et al*., 2006). It is cellular computing model in the physarum polycephalum, its

Corresponding Author: A. Amirthasaravanan, Department of Information Technology, University College of Engineering, Villupuram, Tamil Nadu, India

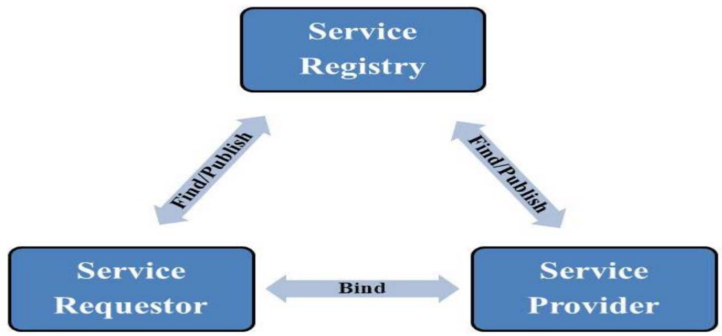

Fig. 1: Web services architecture

biological character inspired and it solves the problem of complex and composite web services and produces optimized web services. In other words from a user perception, this composition and discovery will continue to be deliberated as an effortless service, even though it is composed of numerous web services.

Literature review: Web services are independent software services which are available on the web and it can be retrieved using simple protocols over a network. Web services can do a large array of tasks, ranging from simple request-reply tasks to complete business process interactions (Chaiyakul *et al*., 2006) There is an interface which is described in a machine-process is used to describe the available web services. Based on the web service description using SOAP, other systems can cooperate with the appropriate web service. Typically the communication is done using HTTP and an XML is used to tag the data serialization in union with other Web-related standards. (Chaiyakul *et al*., 2006; Rekaby, 2013) UDDI is used for listing out the services presented. Web services can be executed and used almost in all Programming Languages.

The service provider can provide the web service description and publishes it to a service requestor or service registry (Rajendran and Balasubramanie, 2010). Using service discovery operation, he service requestor can locally retrieve the description of the service or from the service registry. This service description is used to bind with the service provider and work together with the web service implementation. The roles of Service provider and service requestor are logically build and the characteristics of both the service provider and service requestor are disclosed by the service.

The web services architecture isolates the comprehensive elements of the global Web services network that are mandatory in order to ensure interoperability among web services (Mabrouk, 2012). The customary mechanisms of web service discovery and composition are based on the sweeping and binding of all available registries to respond to a client request. With the intensification number of services web, these methods should be optimized for active services response.

To obtain an optimal solution of a given problem, Optimization techniques are used (Wang *et al*., 2006; Rong and Liu, 2010). These optimization techniques minimize time issue or maximize preferred benefit of a given system. Optimization algorithms can be classified into deterministic and stochastic algorithms. Deterministic algorithms do not enclose any operators that cause randomness (Yang and Deb, 2009). These types of algorithms produce similar result as long as their initial conditions remain constant. On the other hand, due to their random nature, stochastic algorithms incline to produce different solutions even when their original conditions remain constant at each run.

The acuity of copying the nature into technology is confirmed to be successful in optimization as described in many literatures. These techniques are being studied and worked in practice at acumulative rate. Their robustness capability of providing from single to multiple solutions of problem and appropriateness of implementation in distributed computing environment makes them ainfluential problem resolving tool (Fig. 1).

MATERIALS AND METHODS

Slime mold (physarum polycephalum): The physarum polycephalum (Slime mold) is a large, single-celled amoeba like organism (Matsumoto *et al*., 2008; Nakagaki *et al*., 2000). The vegetative phase of the life cycle of Physarum polycephalum is the plasmodium, which consists of protoplasmic tube-similar plasmodial veins and many nuclei. Habitually during the search of food, plasmodium spreads out its network of tubes to fill the whole available area that the consent it to

Fig. 2: a) Physarum polycephalum (Slime mold) organism in nature; b) Physarum polycephalum (slime mold)formation of wide network like structure; c) scientist T. Nakagaki on slime mole which determine the shortest path through a maze lab experiment

grow. It sometimes grows to a size of more than a few square meters, while separated segments as small as 1mm can stay alive as individuals. It does not acquire a central information processing mechanism such as a brain to process information intelligently or a refined information communication system such as a nerve system to communicate information throughout its body structure. They process the communication through $_{C}$AMP signals. Therefore, it is indeed remarkable to witness the intelligent behavior.

The slime mould life cycle which alternates between a unicellular feeding stage and a multi cellular reproductive stage is described as follows (Nakagaki *et al*., 2000, 2001, 2004):

- The life cycle begins with spore germination and multiplication
- Haploid amoebae, which consume bacteria, acts as the basic units of slime moulds and they are habitually prowling around on the forest floor
- Once the food supply is exhausted, then individual amoebae starts to move together. And forms the streams of cells calledpseudo plasmodium. It is now known that the process is begun with thecAMPsignaling molecules released by the starving amoebae
- Ultimately the streams come together and form an aggregation which is at the spot with high concentration of cAMP. The mass can number a hundred thousand cells and might reach a size of a few millimeters
- Then they stick together by secreting adhesion molecules and creating a slimesheath (cap) which covers all the cells in the mass. The polarity defined by the anterior (front), posterior (back) ends is established by oxygen gradient. The responsible gene for adhesion can be identified by blocking its expression. The resulting mass becomes loose rubble that is incapable of further development (Fig. 2)

In the times of yore, this physarum has been well-studied from a computational point of view (Matsumoto *et al*., 2008; Nakagaki *et al*., 2000) The Japanese scientist T. Nakagaki and his co-workers found that the physarum was able to determine the shortest path through a maze, as well as connect different arrays of food sources in an efficient manner with low total length yet short average minimum distance between pairs of food sources (Nakagaki *et al*., 2000; Nakagaki, 2001). The maze solving behavior of a physarum is shown in Fig. 2c. Figure 2b shows a thick tube for absorbing nutrients is formed that connects the FSs through the shortest route when two FSs are presented to the physarum in the dark. Because the physarum is photophobic, when the organism is illuminated by an inhomogeneous light field, the tubes connecting the FSs do not follow the simple shortest path, but rather react to the illumination in homogeneity. Figure 2a shows the slime mold formation in nature wood. From experimental observations, researchers also found that the physarum could find the risk-minimum path in the inhomogeneous field of risk.

Contemplation:

Slime mold competence: In a web services the user tends to find the services through request (key words) and the system will find the related services provide to the user as a reply (Rong and Liu, 2010). The whole process is done with the help of WSDL, SOAP, XML and UDDI as discussed earlier. This entire process can be summarized as web services discovery and composition. (Yang, 2010a, b).

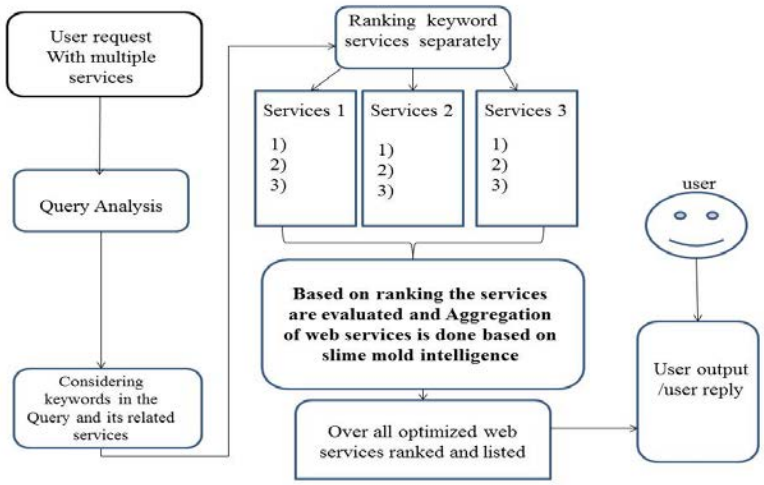

Fig. 3: Work flow of intelligent smile mold algorithm

But, a single service is not sufficient to respond to the user's request and often services should be combined through services composition to achieve a specific goal. In order to have an optimized and effective web services discovery and composition we correlate web services to the cellular computing model in the slime mold (physarum polycephalum). Its biological character inspired and it solves the problem of complex and composite web services. Let us depict the web services discovery and composition inspired by Slime mold pseudo-code:
Pseudo-code of ISM

Alogrithm A; Pseudo-code of inteligent slime mold is given below:

```
Define the objective function
Initialize the slime mold
While (criteria not meet)
{
Search for random key source (KS) (web services)
If (KS0 found)
 Then evaluate KS0 and other nearby KS1 to n randomly based on link conditions;
Evaluate the shortest distance from KS0 to KS 1 to n;
Optimize the paths from KS0 to KS 1to n;
Rank the solution by best path;
End if
If (no other key source available other than KS0) then
Evaluate food source KS0;
Rank the solution;
} Post process result and visualization
End if
```

This Proposed algorithm resolves discovery and composition of multiweb services in a single request problem (Matsumoto *et al*., 2008; Yang, 2009). The slime mold system analyzes the user service request and process the request query randomly. Based on the keywords in user request the first level of ranking is made on services and the service with high priority (high individual ranking) will be the keysource (KS_0) services and from which the system will identify the rest of connective services (KS_1 to KS_n) randomly. For example, a user request service “flight and hotels and taxi in Singapore”. The request contains three different services in single user request (single phrase)i.e., “flight” ”hotels” and “taxi”. If the keyword with high priority ranking is “flights” then the flight services will be the key source from which other services like flight and hotel services are processed randomly (Fig. 3).

Then the slime mold intelligence will identify the shortest path from the key source to other services and then composite all the services into single service and rank them accordingly.

Working of smile mold algorithm: The main objective of the Algorithm is to handle multiple service requests from the user. The user request is processed and the Query is analyzed for related and connecting words like “and or not, in, at” etc., then the main keywords in the Query are coincided and ranked individually (Rothermich, 2002) . By applying slime mold intelligence algorithm, the keyword with highest rank is coincided as the source (service) of the query and from which other keyword (services) are mapped and all the keywords are integrated to a single request and again ranking is made for optimum solution.

RESULTS AND DISCUSSION

Principle model of slime mold for web services: Figure 4 above shows the working principle of intelligent slime mold inspired web services system. In this example, the

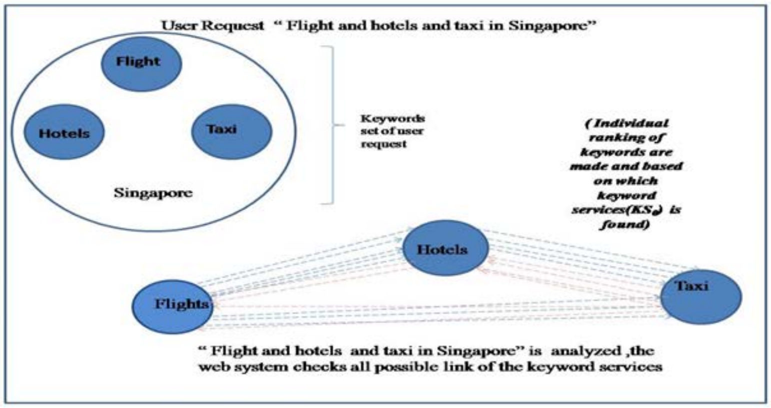

Fig. 4: Analysis diagram for user request web services

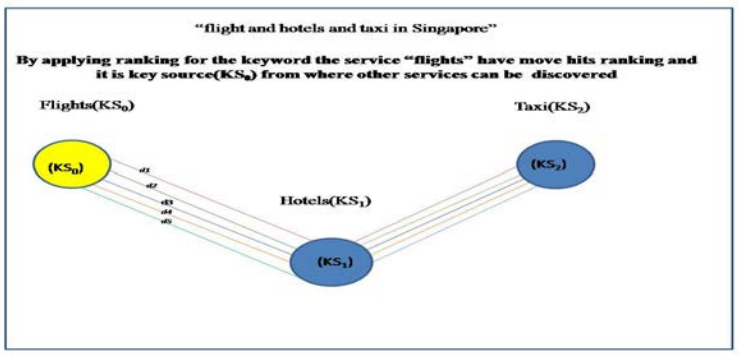

Fig. 5: Ranking the keywords in the user request and there services

user have requested for "flight and hotel and taxi in singapore". Based on the users query request the keyword analysis process is carried out i.e., individual ranking is made for keywordsand all possible links to the keywords are evaluated.

Figure 5 depicts the ranking progression, among the keyword "flights" have high number of (hits) ranking and it is key source service (KS_0) for the users query request from where other services can be discovered and then aggregation process of web services is done based on slime mold intelligence. Since, there may exist many web services for the request, the distance ((KS_0) (KS_1 (KS_2)) association are evaluated and then web services distance are listed (d1, d2, d3, d4 and d5) for further optimization of web services. Let us consider:

$$\left.\begin{array}{l}(KS0) = Flights \\ (KS1) = Hotels \\ (KS3) = Taxi\end{array}\right\} \text{Services}$$

(d_1, d_2, d_3, d_4, d_5...d_n) = Total Distance of Σ(KS) (Response Time) WS = Web Services:

$$\begin{array}{l}((KS_0)(KS_1)(KS_2)) = d_1 \\ ((KS_0)(KS_1)(KS_2)) = d_2 \\ ((KS_0)(KS_1)(KS_2)) = d_3 \\ ((KS_0)(KS_1)(KS_2)) = d_4 \\ ((KS_0)(KS_1)(KS_2)) = d_5 \\ ((KS_0)(KS_1)(KS_2)) = d_6\end{array} \qquad (1)$$

$$\sum((KS_0)(KS_1)(KS_2)) = WS \qquad (2)$$

$$\left.\begin{array}{l}(WS)_1 = d_1 \\ (WS)_2 = d_2 \\ (WS)_3 = d_3 \\ (WS)_4 = d_4 \\ (WS)_5 = d_5\end{array}\right\} \text{List of web services for the request} \qquad (3)$$

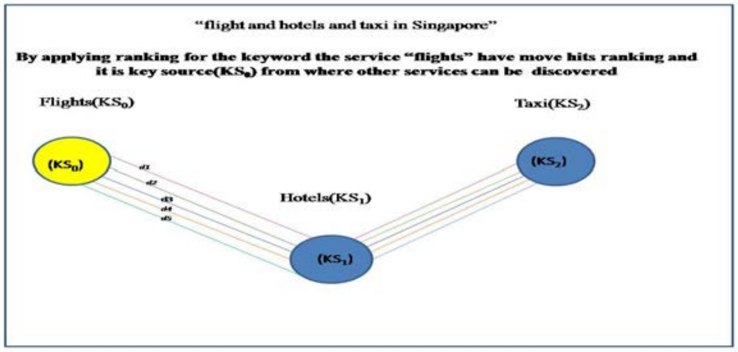

Fig. 6: Finding the optimized solution for the user request by ISM algorithm

$$\left.\begin{aligned}(WS)_1 &= d_1\\(WS)_2 &= d_2\\(WS)_3 &= d_3\\(WS)_4 &= d_4\end{aligned}\right\}\text{Optimized list of web services by applying ISM Algorithm} \tag{4}$$

Figure 6 show the optimized Result for user request service "flights and hotels and taxi in Singapore". The best Web services with less distance i.e. less response time will the best Web service to the user expectation.

Advantages of ISM algorithm: The main contributions of our works lie in three aspects: We bring two desirable characteristics-low- complexity and high-parallelism-into the original physarum model and effective optimization for web services. Compare with the conventional model, our proposed model contains lower complexity and is capable to be implemented in parallel. Therefore, our model is more suitable for designing practical biology-inspired algorithms.

CONCLUSION

In this research, implementation of ISM inspired optimization algorithm was presented. The intelligent slime mold algorithm performs well when searching for optima web services and this algorithm is introduced to handle multiple service requests from the user at single instance. This algorithm minimizes the response time of the system and complexity of the Query request and also accomplishes parallelism. Thus proposed system is capable of providing proficient web service optimization. Our proposed model is more suitable for designing practical biology-inspired algorithms. Future work will test other types of QoS related problems and compare the results of the slime mold algorithm to other optimization algorithms.

REFERENCES

Chaiyakul, S., K. Limapichat, A. Dixit and E. Nantajeewarawat, 2006. A framework for semantic web service discovery and planning. Proceedings of the 2006 IEEE Conference on Cybernetics and Intelligent Systems, June 7-9, 2016, IEEE, Bangkok, Thailand, ISBN: 1-4244-0023-6, pp: 1-5.

Dorigo, M. and T. Stutzle, 2004. Ant Colony Optimization. MIT Press, Cambridge, MA., USA., ISBN-13: 9780262042192, Pages: 305.

Kobayashi, R., A. Tero and T. Nakagaki, 2006. Mathematical model for rhythmic protoplasmic movement in the true slime mold. J. Math. Biol., 53: 273-286.

Mabrouk, N.B., 2012. QoS-aware service-oriented middleware for pervasive environments. Ph.D Thesis, University Pierre and Marie Curie, Paris, France.

Matsumoto, K., S. Takagi and T. Nakagaki, 2008. Locomotive mchanism of Physarum plasmodia based on spatiotemporal analysis of protoplasmic streaming. Biophys. J., 94: 2492-2504.

Nakagaki, T., 2001. Smart behavior of true slime mold in a labyrinth. Res. Microbiol., 152: 767-770.

Nakagaki, T., H. Yamada and A. Toth, 2000. Intelligence: Maze-solving by an amoeboid organism. Nat., 407: 470-470.

Nakagaki, T., H. Yamada and A. Toth, 2001. Path finding by tube morphogenesis in an amoeboid organism. Biophys. Chemistry, 92: 47-52.

Nakagaki, T., H. Yamada and M. Hara, 2004. Smart network solutions in an amoeboid organism. Biophys. Chemistry, 107: 1-5.

Rajendran, T. and P. Balasubramanie, 2010. An optimal agent-based architecture for dynamic web service discovery with QoS. Proceedings of the International Conference on Computing Communication and Networking Tech-nologies, July 29-31, 2010, Karur, Indian, pp: 1-7.

Rekaby, A., 2013. Directed artificial bat algorithm (DABA)-A new bio-inspired algorithm. Proceedings of the International Conference on Advances in Computing, Communications and Informatics, August 22-25, 2013, IEEE, Mysore, India, ISBN: 978-1-4799-2432-5, pp: 1241-1246.

Rong, W. and K. Liu, 2010. A survey of context aware web service discovery: from user's perspective. Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering (SOSE) 2010, June 4-5, 2010, IEEE, Nanjing, China, ISBN: 978-1-4244-7327-4, pp: 15-22.

Rothermich, J.A., 2002. From multicellularity to cell based optimization. Master's Thesis, University of Birmingham, Birmingham, England.

Wang, W., B. Wu, Y. Zhao and D. Feng, 2006. Particle Swarm Optimization for Open Vehicle Routing Problem. In: Computational Intelligence. Huang, D.S. and K. Li (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-37275-2, pp: 999-1007.

Yang, X.S. and H.A. Gandomi, 2012. Bat algorithm: A novel approach for global engineering optimization. Eng. Comput., 29: 464-483.

Yang, X.S. and S. Deb, 2009. Cuckoo search via Levy flights. Proceedings of the World Congress on Nature and Biologically Inspired Computing, December 9-11, 2009, Coimbatore, India, pp: 210-214.

Yang, X.S., 2010a. A New Metaheuristic Bat-Inspired Algorithm. In: Nature Inspired Cooperative Strategies for Optimization, Gonzalez, J.R., D.A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor (Eds.). Springer, Berlin, Germany, ISBN: 9783642125379, pp: 65-74.

Yang, X.S., 2010b. Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio Inspired Comput., 2: 78-84.