

A Novel Bi-Velocity Particle Swarm Optimization Scheme for Multicast Routing Problem

Dr. Subarna Shakya
Professor, Department of Electronics and Computer Engineering,
Central Campus, Institute of Engineering, Pulchowk,
Tribhuvan University,
Pulchowk, Lalitpur Nepal.
Email: drss@ioe.edu.np.

Abstract: A nondeterministic polynomial (NP) with complete Multicast routing problem is defined using a bi-velocity particle swarm optimization (BVDPSO) is proposed in this paper. The shift of particle swarm optimization to the discrete or binary domain, stepping away from the continuous domain is the major impact of the work. Initially a bi-velocity strategy is built such that it characterizes each dimension in terms of 0 and 1. The basic function of this strategy is to describe the MRP's binary characteristics such that 0 stands for the node not being selected while 1 stands for selection. Based on the location and velocity of the original PSO in the continuous domain, the BVDPSO is updated. This will preserve the global search ability and fast convergence speed of the original PSO. 58 instances of large, medium and small scales are used for experimentation in the OR-Library. Based on the results, it is identified that it is possible to get near-optimal or optimal solutions for BVDPSO as it requires generation of limited multicast trees. This approach is found to be optimal over its peers and outperforms recent heuristic algorithms and many advanced techniques used for the MRP problem. They also outperform several PSO, ant colony optimization and genetic algorithms.

Keywords: Steiner tree; Particle swarm optimization; multicast routing problem; communication networks; nondeterministic polynomial;

1. Introduction

Over the past many decades, Multicast routing problem has drawn the attention of many researchers due to its impact and significance in networking and communications system [1]. The MRP is a problem which establishes connection to the destination using multicasting to transmit data provided, the communication network comprises of a set of intermediate, set of destination nodes and set of edges and the source node which are enabled with network connection to find the right node. Over the past few years, much research has been done leading to understanding the importance of MRP is communication networks, giving way to services such as video broadcast, multimedia conferencing, internet telephone, distributed data process and internet protocol television [2]. In this regard, MRP is considered to be an optimization problem which can be built for a specific objective like reducing the delay between the source and destination as it is important for multicasting which is sensitive to latency. Hence, taking into consideration the delay caused at every edge to be the cost and with minimal cost consideration, optimize the multicast tree.

Moreover, other properties such as utility price and reserved bandwidth need to be considered for the edge cost estimation [3]. In graph theory, the optimization of MRP cost can also be called Steiner Tree Problem (STP). However, the issue with STP and MRP is that they are also called a complete nondeterministic polynomial (NP) problem. In order to solve MRP, nondeterministic and approximation algorithms are preferred over deterministic algorithms. In the beginning, a number of theories were proposed for STP like average distance heuristic (ADH), distance network heuristic (DNA) and shortest path heuristic (SPH). However, recent advancement in technology has paved way to modern heuristics like Greedy Randomized Adaptive Search Procedure and Directed Convergence Heuristic (DCH). The drawback is that these algorithms are not optimal when it comes to large networks with multiple edges and nodes as they are based on greedy heuristic algorithm which is not competent enough to optimize the solution in complex environment. Evolutionary computation has led to several algorithms like particle swarm optimization and colony optimization which seem to give acceptable solutions to many complex problems.

The drawback with ACo and GA based algorithms is that they may have low convergence speed, thereby resulting in inefficiency of the system [4]. On the other hand, a PSA is found to have quicker convergence rate and simpler implementation, adapting best to the requirement. Hence the proposed work focuses on using PSO to solve STP which is an MRP lacking QoS constraints. Likewise, an intriguing method of jumping PSO was proposed by Qu et al in [10] and another used discrete PSO to solve STP. However, these methods lacked in many aspects and were not able to offer the standard advantages of the original PSO. To overcome these difficulties, we have proposed to use solve STP, considering it to be an optimization problem, thereby optimizing MRP along with the original PSO in the binary domain. The output of this solution is saved in terms of 0/1 strings where 0 represents that the node isn't selected while 1 represents selection of the node.

Depending on this scheme of coding a bi-velocity DPSO (BVDPSO) which can match binary characteristics of MRP as well as maintain PSO search advantages is proposed. Hence this strategy is distinguished from the other existing methodologies which use set-based strategy, sigmoid function strategy and random jumping strategy. It uses a very straightforward strategy to represent 0 and 1 using two vectors. Further, in order to maintain the original PSO's learning mechanism in the continuous domain, the position and velocity of BVDPSO has to modify position update and velocity. The proposed work gives a comprehensive analysis of BVDPSO in local networks is made for testing small, medium and large scale networks which have a huge amount of destination nodes, edges and other nodes. There are two major aspects concentrated in this work which includes finding the solution for binary optimization problem and to overcome the drawbacks of other heuristic algorithms to find the multicast solution at a quicker rate.

2. Related Works

2.1. PSO in Continuous Domain

In general, one commonly used EC algorithm is PSO which emulates the swarm behaviour of birds gathering together [4]. According to this algorithm, every particle 'i' in an dimensional continuous problem will have a position vector Y_i and velocity V_i which will represent the present location of the particle. The neighbourhood local best position and best position vector are represented by L_i and P_i such that they are used to update X_i and V_i during every generation. Enhanced PSOs are explained in detail in [5], [6] and [7] for PSO industrial applications.

2.2. Formulation of MRP

If $Netw=\{B,A\}$ is a weighted, connected and unidirectional network where A is edge set and B is the node set, then the cost of edge e in A can be expressed as $c(e)$. Set R comprises of all the multinodes along with the source node s such that $R= \{s\} \cup D$. The other nodes are known as Steiner nodes or intermediate nodes [8]. The basic function of this algorithm is to transmit data to the destination node in the set D from the source node with the objective of utilizing minimal cost tree T between the two nodes [9].

3. Proposed Work

3.1 Particle Code

An optimal multicast tree is built using the MRP, considering that it is possible to find a set of nodes (inclusive of all destination nodes, source node and other intermediate nodes). Hence it is easier to build a code of solution which uses the binary string that uses a length which is equal to the nodes in the tree such that it is always assigned bit 1 showing the presence of tree [11]. Further, it should also be noted that if there any intermediate nodes with the bit value of 1, it means that, that particular node is used in the multicast tree construction. If the node is assigned the value of 0, it means that the node is not used. Hence, if there are N nodes in total, then the position 'i' of every particle can be expressed as

$$Y_i = [y_{i1}, y_{i2}, y_{i3}, \dots \dots y_{iN}] \text{ where } y_{ij} = 0 \text{ or } 1$$

Similarly, velocity is represented as given below

$$V_i = \begin{bmatrix} v_{i1}^0, v_{i2}^0, \dots, v_{iN}^0 \\ v_{i1}^1, v_{i2}^1, \dots, v_{iN}^1 \end{bmatrix}$$

Here the values of v_{ij} will vary between 0 and 1.

The above equation represents the possibility of y_{ij} being 1 for v_{ij}^1 and y_{ij} being 0 for v_{ij}^0 wherein the j th dimension represents the bivalued for V_i . It should be observed that the values of v_{ij}^1 and v_{ij}^0 represent the difference between the two positions and are not dependent on the values of x_{ij} assignment to 1 and 0 leading to the sum the components to be equal to 1.

3.2 Velocity update

It is vital to follow the original PSO learning concept [11] when updating velocity of particle in BVDPSO. The proposed work implements the following equation with minor changes:

$$v_{ij} = \omega * v_{ij} + c_1 * r_{1j} * (p_{ij} - x_{ij}) + c_2 * r_{2j} * (l_{ij} - x_{ij})$$

There are three modifications made in order to adapt to bi-velocity code and to fit 0 and 1 position in the proposed BVDPSO. They are as given below:

- Velocity can be defined as the sum of velocities.
 If there are two velocities V_1 and V_2 involved, then the final velocity can be found using the formula $V_i = V_1 + V_2$. Let us take an example where $V_1 = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.5 & 0.6 & 0.7 & 0.8 \end{bmatrix}$ and $V_2 = \begin{bmatrix} 0.3 & 0.5 & 0.2 & 0.1 \\ 0.4 & 0.4 & 0.2 & 0.1 \end{bmatrix}$. Then $V_i = \begin{bmatrix} 0.4 & 0.7 & 0.5 & 0.5 \\ 0.9 & 1 & 0.9 & 0.9 \end{bmatrix}$
- Velocity can be defined as the difference in positions.
 Velocity = $P_1 - P_2$, where P_1 and P_2 are the positions
 Based on the positions of Y_1 and Y_2 , velocity can be determined using the formula $V_i = Y_1 - Y_2$ keeping the original PSO learning concept. Let us consider that y_{1j} is 'a' and y_{2j} is 'not a' (where 'a' may be 1 or 0) then at the j th dimension, Y_2 and Y_1 are different which means that Y_2 should learn from Y_1 since it is in a superior position. Let us take an example where $Y_1 = [1, 1, 0, 1, 0, 1, 1, 0]$ and $Y_2 = [1, 1, 1, 0, 1, 0, 1, 1, 0]$. Then the output is

$$V_i = Y_1 - Y_2 = \begin{bmatrix} 0, 0, 0, 1, 0, 0, 0, 0 \\ 0, 0, 1, 0, 0, 0, 0, 0 \end{bmatrix}$$

- Velocity can be defined as the product of velocity and coefficient. Every element of the current velocity is multiplied with $c*r$ which is equal to w , the Coefficient. In such a case, the value of elements higher than 1 will be prefixed to be equal to 1. Let us take an example where $c*r = [1.5, 1.6, 0.4, 1.1, 1.9, 0.3, 1.3, 1.5]$ and $V = \begin{bmatrix} 0, 0, 0, 1, 0, 0, 0, 0 \\ 0, 0, 1, 0, 0, 0, 0, 0 \end{bmatrix}$. Then

$$V = (c*r) * V = \begin{bmatrix} 0, 0, 0, 1.1, 0, 0, 0, 0 \\ 0, 0, 0.4, 0, 0, 0, 0, 0 \end{bmatrix} = \begin{bmatrix} 0, 0, 0, 1, 0, 0, 0, 0 \\ 0, 0, 0.4, 0, 0, 0, 0, 0 \end{bmatrix}$$

3.3 Position update

A position update includes addition of the velocity to the position Y_i . However, due to the difference in domain, it is not possible to add them. Hence a new position is devised following a strategy such that it is possible to continue learning the original PSO concept. The strategy is defined as follows:

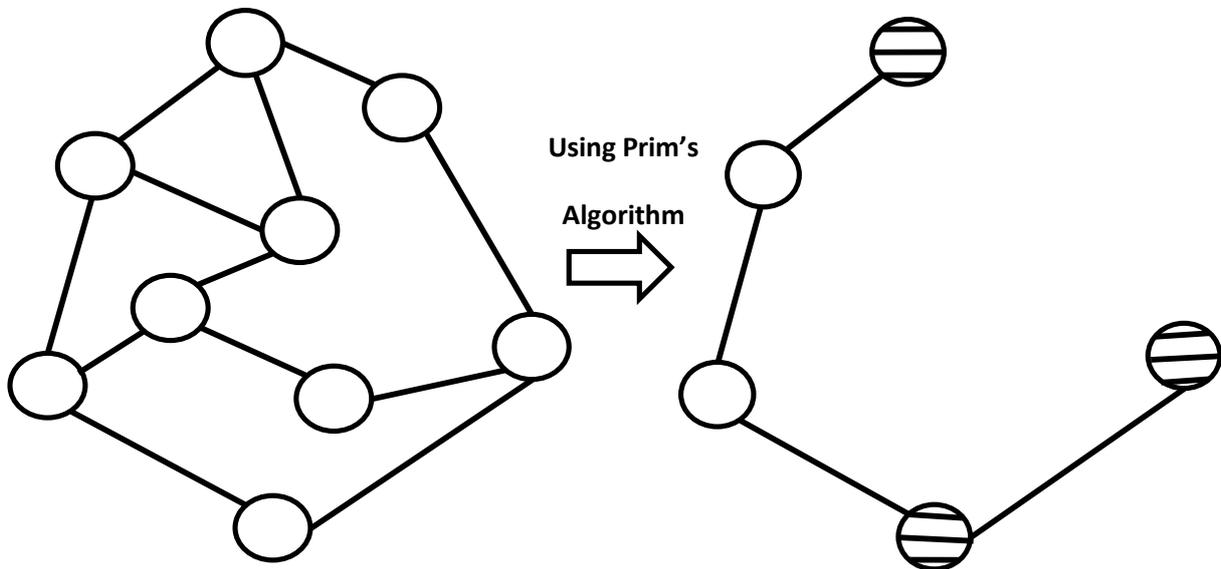
$$y_{ij} = \begin{cases} 1, & \text{when } (v_{ij}^1 > a \text{ and } v_{ij}^0 \leq a) \\ 0, & \text{when } (v_{ij}^1 \leq a \text{ and } v_{ij}^0 > a) \\ y_{ij}, & \text{when } (v_{ij}^1 > a \text{ and } v_{ij}^0 \leq a) \\ rand\{0,1\}, & \text{when } (v_{ij}^1 \leq a \text{ and } v_{ij}^0 > a) \end{cases}$$

This will ensure that the new velocity is given the right attention, thereby enabling exemplars for the particle to learn the original PSO concept.

3.4 Fitness Evaluation

Fitness Evaluation is one of the biggest problem when using BVDPDSO for MRP optimization. A 0/1 string is attached to the nodes indicating whether or not they are used in the construction of multicast tree. Based on the string, the fitness evaluation is made and the cost of the tree is calculated. Using Floyd's algorithm [12], a cost complete graph (CCG) is built transforming the MRP network that is of low cost. Depending on the CCG, Prim's minimum spanning tree algorithm is modified and pruned such that leaves which aren't destination nodes are deleted and then multicast tree is constructed. A position update includes addition of the velocity to the position Y_i . However, due to the difference in domain, it is not possible to add them. Hence a new position is devised following a strategy such that it is possible to continue learning the original PSO concept.

- **CCG procedure:** According to this procedure, two nodes are connected by either direct or indirect paths depending on whichever is the shortest. Hence a proper record of every node travelled is to be maintained for ease of path restoration. Let the two nodes to 3 and 4, then then $C_{(3,4)} = -1$ which means they are not directly connected to each other. Hence the shortest path travelled can be recording using the data structure $S_{(3,4)}$ representing the length of the path between 3 and 4.
- **Modified Prim's Algorithm:** This algorithm prefers the use of direct path instead of the indirect one which is why the multicast is built using solution string with node value of 1. Hence this algorithm leans towards choosing the shortest direct path even if it is actually longer than the indirect paths. The following example showcases the output using Prim's Algorithm for $X=[1, 0, 1, 0, 0, 0, 1, 1, 1]$



4. Results and Discussion

Results of the proposed work are represented in **Fig.1**. Based on the observation it is noted that the maximum generation G is found to be 500, 1250 and 2500 for the population size M with values 20, 50 and 100 resulting in 9 combinations.

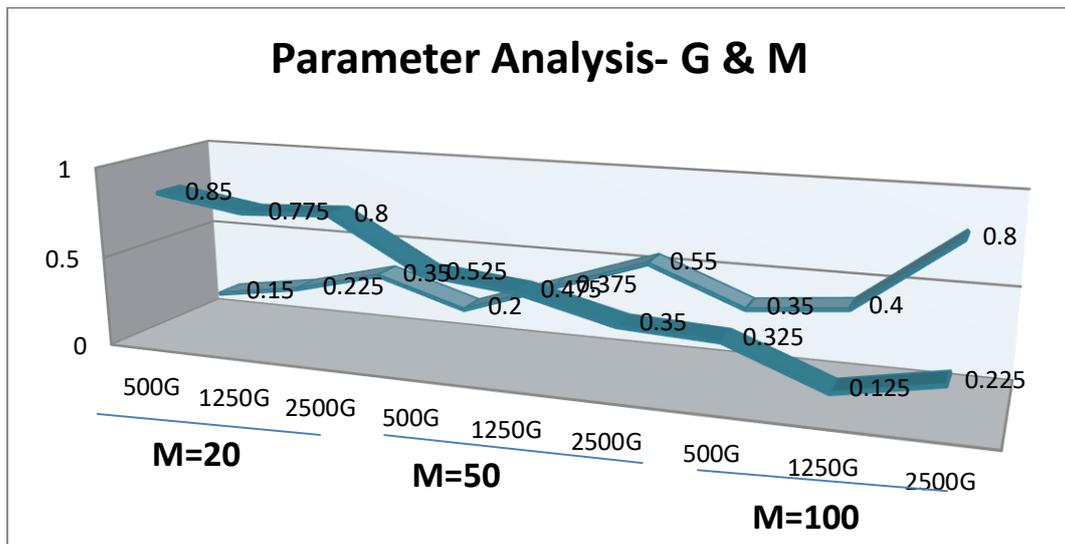


Fig.4.1 Parameter Analysis of G and M for C09

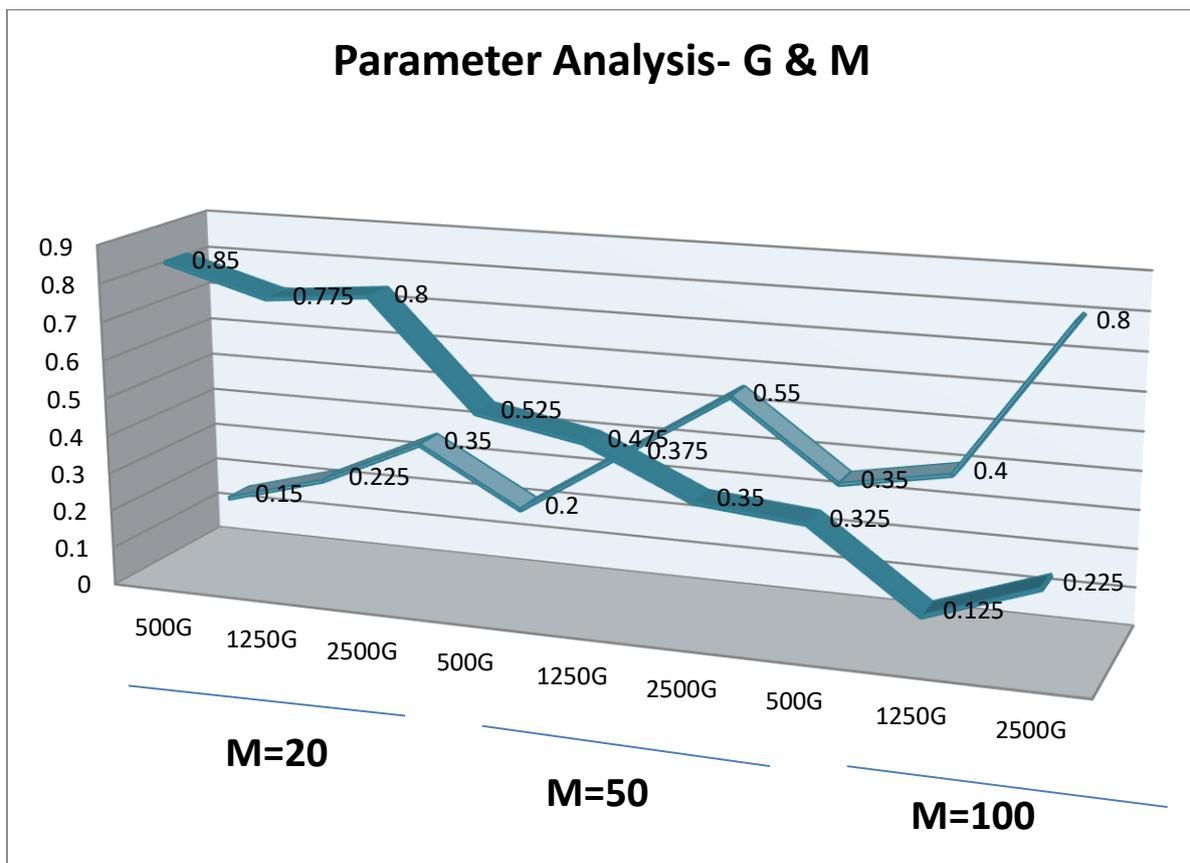


Fig.4.2 Parameter Analysis of G and M for C14

Here the value of CPU time and R are found to be 32 independent runs at an average for individual combinations. Based on the observations, it is noticed that as G or M increases, the accuracy level also improves significantly. But this will correspondingly increase the CPU time. Hence a trade off between CPU time and accuracy should be established for proper functioning of the algorithm.

5. Conclusion

In this algorithm, BVDPSO approach has been followed in order to optimize MRP. The proposed work portrays the improved efficiency in solving MRP, resulting in higher accuracy when compared with the other existing approaches. The result also shows faster convergence speed and provides a novel approach to solve multicasting problems. As this approach has good future scope in finding the solution for constrained problems, it is possible to extend it further to be suitable for multi-objective MRP problems.

References

- [1] A. Sabbah, A. EI-Mougy, and M. Ibnkahla, "A survey of networking challenges and routing protocols in smart grids," *IEEE Trans. Ind. Informt.*, vol. 10, no. 1, pp. 210–221, Feb. 2014.
- [2] G. Kandavanam, D. Botvich, S. Balasubramaniam, and C. Kulatunga, "PaCRAM: Path aware content replication approach with multicast for IPTV networks," in *Proc. IEEE Globecom*, 2010, pp. 1–6.
- [3] G. Kandavanam, D. Botvich, S. Balasubramaniam, and B. Jennings, "A hybrid genetic algorithm/variable neighborhood search approach to maximizing residual bandwidth of links for route planning," in *Proc. 9th Int. Conf. Artif. Evol.*, 2009, pp. 49–60.
- [4] R. J. Wai, J. D. Lee, and K. L. Chuang, "Real-time PID control strategy for maglev transportation system via particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 629–646, Feb. 2011.
- [5] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.
- [6] W. N. Chen, J. Zhang, Y. Lin, N. Chen, Z. H. Zhan, H. Chung, Y. Li, and Y. H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. Apr. 2013.
- [7] K. Chan, T. Dillon, and E. Chang, "An intelligent particle swarm optimization for short-term traffic flow forecasting using on-road sensor systems," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4714–4725, Oct. 2013.
- [8] H. Wang, X. X. Meng, S. Li, and H. Xu, "A tree-based particle swarm optimization for multicast routing," *Comput. Netw.*, vol. 54, no. 15, pp. 2775–2786, Oct. 2010.
- [9] R. Qu, Y. Xu, J. P. Castro, and D. Landa-Silva, "Particle swarm optimization for the Steiner tree in graph and delay-constrained multicast routing problems," *J. Heuristics*, vol. 19, no. 2, pp. 317–342, Apr. 2013.

- [10] R. J. Wai, J. D. Lee, and K. L. Chuang, "Real-time PID control strategy for maglev transportation system via particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 629–646, Feb. 2011.
- [11] H. P. Li and Y. Shi, "Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574–1582, Mar. 2014.
- [12] A. F. Zobaa, "Optimal multiobjective design of hybrid active power filters considering a distorted environment," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 107–114, Jan. 2014.