QoS-VNS-CS: QoS constraints Core Selection Algorithm based on Variable Neighborhood Search Algorithm

Youssef Baddi and Mohamed DafirEch-Cherif El Kettani

Information Security Research Team – ISeRT,
ENSIAS Mohammed V-Souissi University,Rabat, Morocco
Youssef.baddi@um5s.net.ma, dafir@um5s.net.ma

***Abstract*:**Within the development of network multimedia technology, more and more real-time multimedia applications arrive with the need to transmit information using multicast communication. Multicast IP routing is an important topic, covering both theoretical and practical interest in different networks layers. In network layer, there are several multicast routing protocols using multicast routing trees different in the literature. However PIM-SM and CBT protocols remain the most used multicast routing protocols; they propose to use a shared Core-based Tree CBT. This kind of tree provides efficient management of multicast path in changing group memberships, scalability and performance. The main problem concerning the construction of a shared tree is to determine the best position of the core. QoS-CS's problem (QoS constraints core Selection) consists in choosing an optimal multicast router in the network as core of the Shared multicast Tree (CBT), within specified associated QoS constraints. The choice of this specific router, called RP in PIM-SM protocol and core in CBT protocol, affects the structure of multicast routing tree, and therefore influences performances of both multicast session and routing scheme. QoS-CS is an NP complete problem which needs to be solved through a heuristic algorithm: in this paper, we propose a new core Selection algorithm based on Variable Neighborhood Search algorithm and a new CMP fitness function. Simulation results show that good performance is achieved in multicast cost, end-to-end delay, tree construction delay and others metrics.

***Keywords*:** Core, QoS-VNS-CS, QoS, PIM-SM, CBT, Multicast routing, QoS-CS**.**

1. Introduction

Many distributed real-time applications, such as audio- and video-conferencing, collaborative environments and distributed interactive simulation require simultaneous communication between groups of computers with quality of service (QoS) guarantee; these applications involve a source in sending messages to a selected group of receptors. Classic unicast and broadcast network communication is not optimal; therefore, Deering[1] proposed a technique called IP multicast routing for one-to-multiple and multiples-to-multiple communication, which entrusts the task of data duplication to the network: applications can send one copy of each packet and address it to the group of involved computers; the network takes care of message duplication to the receivers of the group.

Multicast technology has become a research hotspot. Group communications on Internet are increasingly pervasive parallel to the wider acceptance of group communication applications over the Internet. Multicast IP is a bandwidth conserving technology that reduces traffic in the network, and by the many, the bandwidth consumption.

Multicast communication is based on a multicast tree for data routing; multicast routing protocols are built using two kinds of multicast trees: Source Based Tree SBT and shared Core-Based Tree CBT. With Source Based Tree, a separate tree is built for each source. With a shared Core-Based Tree, one tree is built for the entire group and shared among all senders; core based trees have a significant advantage in terms of routing resources more than source-based trees in that only one routing table entry is needed for the group [2].

PIM-SM [3] and CBT [4] protocols are multicast routing protocols based in shared Core-Based multicast Tree to forward multicast packets. Construction of this kind of tree requires the selection of a center router called "Rendezvous point" RP in PIM-SM [3] protocol and Core in CBT [4] protocol; With the proliferation of existing multimedia group applications, the construction of Shared multicast tree satisfying the quality of service (QoS) requirements is becoming a problem of prime importance.

Find out an optimal router as a core with Qos guarantee is known by QoS Constraints Core selection Problem. This problem is known to be an NP complete problem [1], [5], [6], which needs to be solved with a heuristic algorithm. Current implementations of the PIM-SM [3] and CBT [4] protocols decide on the Core router administratively [7], based only in priority and IP address of each candidate. This leads to high cost, high delay, and high congestion, by the many, directly impact tree structure and performance of the routing scheme of multicast.

In this paper, we introduce newQoS constraints Core Selection Algorithm QoS-VNS-CS that can improve the delay and delay variation constraint in a multi-source shared Core-Based Tree. In this work we adopted a heuristic search algorithm [7] named Variable Neighborhood Search VNS [8].

This paper is organized as follows. In the next section, we describe the core selection problem. Section 3 is devoted to the description of a mathematical modeling of core selection problem. Section 4 presents the state of research of the core selection problem in the literature. Section 5 describes the proposed QoS-VNS-CS algorithm for the core selection problem. Simulation results are reported in Section 6. Finally, Section 7 provides concluding remarks.

2. Background

The main role of a multicast routing protocol consists in managing multicast groups and routing multicast messages through constructed optimal multicast tree in order to reach all group nodes, which facilitates the operation of multicast packet duplication. Constructing an optimal unique multicast tree covering all multicast groups members (receivers and sources) at the same time is known by the minimum Steiner tree problem (MST) [9]; this problem is NP complete [2], [6], [10], since it seeks to find a low-cost tree spanning all members of the group at a time by minimizing cost and transmission delay. Because of the difficulties in obtaining SMT, especially in larger graphs such as Internet Network, it is often estimated acceptable to use other optimal trees to replace SMTs through a heuristic algorithm. Multicast routing protocols are classified in many categories [11], we mention in this work the most used and implemented two categories [12]: Source Based Tree SBT and shared Core-Based Tree CBT.

Source based tree SBT or Shortest Path Tree SPT is composed of the shortest paths between the source as root and each receivers of multicast group. The main motivations behind using a source based tree SBT are the simplicity of building in a distributed manner using only the unicast routing information [13], [14], and optimization of transmission delay between source and each receiver [6]. The main drawbacks of SBT are: additional costs for maintaining SBT trees, and the number of statements to be stored in the nodes. Complexity is O(S * G) (S is the number of sources and G is the number of groups) [6]. The shortest path tree SPT is used by several multicast routing protocols such as DVMRP [15], MOSPF [16], and PIM-DM [17].

Generally, Source-Based Trees are mostly suitable for small-scale, local-area applications. The main motivation for their use is delay optimization during multicast forwarding. They are not adapted to sparse mode situation because of the additional overhead of tree maintenance; this leads to an overhead in terms of total reduction of the traffic; also the scalability of source-based protocols tends to degradation in terms of network resource consumption [6].

Shared tree can be constructed using a shared core tree: It requires the selection of a central router called "Rendezvous point" RP in PIM-SM [3] protocol and “Core” in CBT [4] protocol.

Shared core trees are more appropriate when there are multiple sources in the multicast group. Under this approach, Shared trees separate the concept of source from that of the tree root. One node in the network is chosen as the center, and the sources forward messages to the center. Like SBT tree, a shortest path multicast tree is constructed rooted at the selected Core, offering better flexibility and extensibility. Also only routers on the tree need to maintain information related to group members. It gives good performance in terms of the quantity of state information to be stored in the routers and the entire cost of routing tree [15]–[17].

Joining and leaving a group member is achieves explicitly in a hop-to-hop way along the shortest path from the local router to core router resulting in less control overhead, efficient management of multicast path in changing group memberships, scalability and performance [1], [18].

Several multicast routing protocols in the literature use Shared Core-Based Tree: Protocol Independent Multicasting-Sparse mode PIM-SM [3] and Core-Based Tree (CBT) [4]. With an advantage for PIM-SM, which provides the advantage to use both CBT tree [4] as default and source based tree when a customer request.

Current implementation of PIM-SM [3] and CBT [4] protocols divides the tree construction problem into two sub-problems: the first is center selection problem and the second is routing problem. PIM-SM [3] (and CBT [4]) uses for center selection a special router called Bootstrap router (BSR) defined in RFC 5059 [19], which notifies a set of candidate cores. Every node uses a Hash function to map to one core, according to the address of the group; this hash function is based on router priority and his IP address. Both of these parameters do not guarantee the selection of an optimal core with any delay and delay variation guaranties. This leads to high cost, high delay, and high congestion. This problem first proposed by G. N. Rouskas and I. Baldine[5], is an NP complete problem[2], [6], [10], which needs to be solved through a heuristic algorithm.

In this paper, we propose a new Core Selection Algorithm QoS-VNS-CS based on a “Variable Neighborhood Search”. VNS algorithm has already been applied successfully to resolve a wide variety of NP-hard problems[20]–[24] to select a global optimal solution using several neighborhoods structures systematically, but not yet in Core selection problem with QoS guarantee. QoS-VNS-CS can simultaneously minimize the delay, delay variation and cost of the multicast tree. It attempts to find the best Core using a fitness function.

3. Mathematical Modeling

In this section, we describe network topology Mathematical Modeling and cost function used to evaluate our solution. Many cost functions are a heady proposed in the literature [25] and compared in many works [12].

A computer network is modeled as a simple directed and connected graph $G = (N, E)$, where N is a finite set of nodes and$E$ is the set of edges (or links) connecting the nodes. Let $|N|$ be the number of network nodes and $|E|$ the number of network links. An edge$e \epsilon E$connecting two adjacent nodes $u \epsilon N$ and $v \epsilon N$ will be denoted by $e(u, v)$, the fact that the graph is directional, implies the existence of a link $e(u, v)$ between $v$and u. Each edge is associated to two positive real values: a cost function $C(e) = C(e(u, v))$ represents link utilization (may be either monetary cost or any measure of resource utilization), and a delay function $D(e) = D(e(u, v))$ represents the delay that the packet experiences through passing that link including switching, queuing, transmission and propagation delays. We associate for each path $P(v_0, v_n) = \big(e(v_0, v_1), e(v_1, v_2), \dots, e(v_{n-1}, v_n)\big)$in the network two metrics:

$$C\big(P(v_0, v_n)\big) = \sum_{0}^{n-1} C\big(e(v_i, v_{i+1})\big) \; (1)$$

And

$$D\big(P(v_0, v_n)\big) = \sum_{0}^{n-1} D\big(e(v_i, v_{i+1})\big) \; (2)$$

A multicast tree$T_M(S, C, D)$ is a sub-graph of G spanning the set of sourcesnode$S \subset N$ and the set of destination nodes $D \subset N$ with a selected core C.Let$|S|$ be the number of multicast destination nodes and $|D|$ is the number ofmulticast destination nodes.

In Protocols using Core-based tree, all sources node need to transmit themulticast information to the selected core via unicast routing, then itwill be forwarded to all receivers in the shared tree.In order to model the existence of these twoparts separated by core, we use both followingcost and delayfunctions:

$$C(T_{\mathrm{M}}(S,C,D)) = \sum_{s\in S} C\big(P(s,C)\big) \quad + \sum_{d\in D} C\big(P(C,d)\big) \quad (3)$$

And

$$D(T_{\mathrm{M}}(S,C,D)) = \sum_{s\in S} D\big(P(s,C)\big)$$

$$+ \sum_{d\in D} D\big(P(C,d)\big) \qquad (4)$$

We also introduce a Delay Variation (7) function defined as the difference between the Maximum (5) and Minimum (6) end-to-end delays along the multicast tree from the source to all destination nodes and is calculated as follows:

$$Max_{Delay} = Max(D\big(T_{\mathrm{M}}(S,C,D)\big)(5)$$

$$Min_{Delay} = Min(D\big(T_{\mathrm{M}}(S,C,D)\big)(6)$$

$$DelayVariation = Max_{Delay} - \quad Min_{Delay}(7)$$

Consequently, based upon the above definition, we can now state mathematically the multicasting routing problem in our paper as: for a given weighted graph$G = (N,E)$, including a set of multicast sources node $S \subset N$, a destination node set $D \subseteq N$, α and βconstant, the problem consist in finding a core router to construct an optimal multicast tree $T_M(S,C,D)$:

$$\begin{gathered} Min\ C\big(T_{\mathrm{M}}(S,C,D)\big) \\ Delay < \alpha \qquad (8) \\ DelayVariation < \beta \end{gathered}$$

4. Literature Review

Bootstrap RP [19] was added to PIM version 2 [3] as a standard mechanism for dynamic allocation of a Rendezvous Point RP, In Bootstrap RP [19] mechanism, the Rendezvous Point RP selection based on a list of candidate routers according to the priority of each one. Bootstrap RP uses for center selection a special router called Bootstrap router BSR, which notifies a set of candidate RPs. BSR router uses Hash function to select one Rendezvous Point RP; this hash function is based on router priority declared by each Rendezvous Points candidates and his IP address. This method of selection does not assume selecting best router location and does not take into account the distribution of multicast group members, which affects the performance of the multicast session.

There are several others proposals, algorithms and mechanisms for core selection problem in the literature. A variety of these algorithms are compared in [11]. Among proposed selection algorithms, we find the Random Selection, in which, the center is chosen randomly among the network. It is comparable to selecting the first source or the initiator of the multicast group, as proposed in PIM-SM [3] and CBT [4] protocols.

4.1 Core selection based on the QoS constraints

Some Proposed algorithms select Core on the basis of basicheuristics and do not consider QoS constraints. This kind of Core selection can provide every member of the group with a cost function guarantee to the Core iteratively selected, with this set of algorithms is hard to guarantee QoS for all group members.

Shields and Garcia-Luna-Aceves[26] have proposed OCBT to avoid looping problem, present in the initial design of CBT [4]. Theoretically, Optimal Center-Based Tree OCBT [26] selection algorithm is the best. It considers all nodes as a list of candidates' cores: From this list, the best node is selected. But practically, this process requires more processing time because it is required to calculate the actual cost of the tree rooted at each node in the network each time, and pick the one which gives the lowest cost. Several other algorithms which operate on all nodes in the network, as OCBT [26] with change of the loss function; was proposed by Wall [27], especially Maximum-Centered Tree (MCT), Average-Centered Tree (ACT) and Diameter-Centered Tree (DCT). To reduce the area of research and the execution time, Minimum Shortest Path Tree (MSPT) is suggested. This approach requires calculating the actual costs for the trees rooted at each group member, and chooses the member with the lowest cost.

Topology-Based Algorithm [10] uses the domain topology and sub-graph constructed from the multicast group to select a single core closest to topology center. This selection method requires knowledge of more information than random selection method; this information should quickly include characteristics that was not altered. All selected cores by this algorithm for all multicast groups are supposed to be in the proximity of the center of network. Therefore, the multicast traffic will converge at this region and the increase of multicast groups will cause overload at cores. Topology-Based Algorithm will use an excessive number of nodes to calculate optimal core: the execution time becomes important, especially in a large topology internet such as. Time complexity is equal to $O(N)$.

To reduce the search area used by the Topology-Based Algorithm, and select a distributed cores for all multicast groups in the network domain, [10] proposed group-based algorithm: it takes as parameter location information of all group multicast members (recipients and sources) in addition to information about the network topology, which makes it more complex and requires more information. Unlike the topology-based algorithm which selects a core for the entire network, group-based algorithm selects a core for each group; the core selected is close to the group members, thus avoiding convergence of traffic, this selection process needs a list of all multicast group members and its time complexity is equal to$O(D)$. Group-based algorithm[10] is more efficient when the multicast group is located in the same topological area, the practical implementation of this algorithm depends not only on the availability of information about the multicast group members and their locations, but also requires an effective and reliable mechanism for the core migration to other cores during a change of the multicast group members distribution.

Tournament-based algorithm proposed by Shukla, Boyer, and Klinke [18] executes a Distributed tournament between nodes to determine a center. This tournament-based algorithm needs a list of all sources and members at each source and receivers group. Algorithm execution starts with matching sources with group members in declining order of hop-count metric. If the number of sources and members is odd, the remaining nodes are matched randomly. The winner of each pair is determined by finding the node in the proximity to the middle of the shortest path connecting the pair. Algorithm execution finishes when one winner remains. Tournament-based algorithm involves cooperation between nodes, and requires knowledge of the network topology. Finding middle of path between each pair requires an exchange of route tracing messages and takes$O(|D|^2)$ as complexity time.

Tabu Search algorithm for RP selection (TRPSA) [28] is a distributed core selection algorithm based on dynamic meta-heuristic Tabu Search TS algorithms proposed first by Glover [29] to solve combinatorial optimization problems in PIM-SM protocol [3]. TRPSA [28] tries to find a local solution after a certain finite number of iterations by using memory structures that describe the visited solutions. The basic idea of the TRPSA [28] algorithm is to mark the best local solution obtained in order to prevent the research process to return back to the same solution in subsequent iterations using a data structure to store the solutions already visited, this structure is called tabu list. However, the method requires a better definition of stopping criterion and effective management of the tabu list, since the choice of stopping criterion and tabu list size is critical and influences the performance of the algorithm. According to [26], TRPSA has$O(|E| + (|E| + (|S| + |D|) * N^2)$ complexity.

We cite also our proposed algorithms VNS-RP [30], VND-CS [31] and GRAS-RP [32] based in VNS [8], VND [33] and GRAS [34] heuristics successively.

4.2 Core selection based on QoS constraints

There are also many well-known approaches to select core router satisfying QoS constraints.

Delay Variation Multicast Algorithm (DVMA) was proposed by G. N Rouskas, I. Baldine[5] to resolve the Delay and Delay Variation Bounded Multicasting Network (DVBMN) problem. DVMA [5] tries to find a sub-network given a source and a set of destinations that satisfies the QoS (Quality of Service) requirements on the maximum delay from the source to any of the destinations and on the maximum inter-destination delay variance: it starts with a source-based tree spanning some and not always all multicast members satisfying the delay constraint only. Then the algorithm searches through the candidate paths satisfying the delay and delay-variation constraints from a non-tree member node to any of the tree nodes. DVMA [5] is most classed in source-based tree then shared tree, and it assumes that the complete topology is available at each node. The computer simulation shows that the performance of DVMA [5] is good in terms of multicast delay-variation. However, it shows a high complexity ($O(kldn^4)$) where k and l are the number of paths satisfying the delay bound between any two nodes; $|D| = d$ and$|N| = n$ represents number of multicast receptors node and total number of nodes in the topology network respectively.

Delay and Delay Variation Constraint Algorithm (DDVCA) was proposed by Sheu and Chen [6] based on the Core Based Tree (CBT) [4]: the main objective of DDVCA [6]is to find as much as possible core router spanning a multicast tree with a smaller multicast delay variation under the multicast end-to-end delay constraint. To do that, DDVCA [6] first calculates the delay of the least delay path from the destination nodes to all the nodes. The node that has the minimum delay-variation is selected as the core node. In comparison with the DVMA [5], DDVCA [6] Algorithm shows a significant lower complexity i.e. $O(dn^2)$where d is the number of destination nodes and n is the total number of nodes in the computer network.

KIM et.al [35] has proposed another efficient core selection algorithm based also on CBT like DDVCA [6] to build a core based multicast tree under delay and delay-variation bound. First, AKBC [35] finds a set of candidate core nodes that have the same associated multicast delay-variation for each destination node. Then, it selects a final core node from this set of candidate core nodes that has the minimum potential delay-variation. AKBC [35] algorithm investigates candidate nodes to select the better node with the same complexity as DDVCA [6] i.e. $O(dn^2)$.

All these algorithms (DDVCA [6], DVMA [5] and AKBC [35]) are only applied in the symmetric network environment that has no direction. To overcome this limitation, Ahn, Kim and Choo[36] proposed AKC (Ahn Kim Choo) to build a multicast tree with low delay-variation in a realistic network environment that has two-way directions. This algorithm works efficiently in the asymmetric network with the same complexity as DDVCA [6] i.e. $O(dn^2)$.

Sahoo and. al [37] proposed TRPSA Algorithm based on dynamic meta-heuristic Tabu Search TS algorithms, proposed first by Glover [29], to solve combinatorial optimization problems. Tabu Search algorithm for RP selection (TRPSA) [37] is a distributed core selection algorithm to find a local solution after a certain finite number of iterations by using memory structures that describe the visited solutions. The basic idea of the TRPSA [37]algorithm is to mark the best local solution obtained in order to prevent the research process to return back to the same solution in subsequent iterations using a data structure to store the solutions already visited, this structure is called tabu list. However, the method requires a better definition of stopping criterion and effective management of the tabu list, since the choice of stopping criterion and tabu list size is critical and influences the algorithm performances. According to [37], TRPSA has$O(|E| + (|E| + (|S| + |D|) * N^2)$ complexity.

However, these algorithms DDVCA [6], DVMA [5] and AKBC [35]) select the best core node out of a set of candidate core nodes that have the same associated delay-variation. Therefore, these algorithms are restricted only to selecting the best core node, which may not generate an optimal delay-variation-based multicast tree in many cases. Also TRSPA [35] doesn't overcome this limitation because it just selects a local optimal node which may not generate an optimal delay and delay-variation-based multicast tree in all topology networks.

The last core selection algorithm proposed by Baddi and El Kettani[38] is D2V-VNS-RPS: this algorithm, proposed as an extension to BootStrap RP in PIM-SM [3] protocol, uses VNS algorithm with a simple cost function to select one router as Rendezvous Point RP; this function tries to select the best router in terms of delay and delay variation using VNS algorithm to evaluate the solution after each iteration. Therefore, this algorithm is restricted only to selecting the best Rendezvous Point node whiteout caring in the Rendezvous Point position, which may not generate an optimal delay and delay-variation based multicast tree in many cases. To overcome this limitation, we propose in this work a QoS-VNS-CS algorithm compatible with all Core-Based Tree protocols using a MODE function [36] to manage the core location.

5. QoS-VNS-CS Algorithm

5.1 Basic variable neighborhood search algorithm

Contrary to all others kind of meta-heuristics based on local search methods, Mladenović and Hansen [8] proposed a recent meta-heuristic Variable Neighborhood Search VNS Algorithm based on the simple idea of a systematic neighborhood changing arbitrarily, which varies in size, but usually with increasing cardinality, within a local search algorithm (Hill Climbing, Simulated annealing, tabu ...).
VNS has been applied successfully to a wide variety of NP-hard problems to select a global optimal solution such as the travelling salesman problem [20], Job Shop Scheduling Problems [21], the clustering problem [22], arc routing problems [20], and nurse rostering[23].
The use of more than one neighborhood provides a very effective method that allows escaping from a local optimum. In fact, it is often the case that the current solution, which is a local optimum in one neighborhood, is no longer a local optimum in a different neighborhood; therefore, it can be further improved using a simple descent approach.
As defined by Mladenović and Hansen [8] and presented in figure 1, in the VNS paradigm, a finite set of neighborhoods structures$N_k(k = 1, \dots, k_{max})$and an initial solution S are generated, starting from this initial solution, a so-called shaking step is performed by randomly selecting a solution S' from the first neighborhood, This is followed by applying an iterative improvement local search algorithm to get a S" solution. If this solution(S'') improves the weight function presented in formula (8) one starts with the first neighborhood of this new solution ($S' \leftarrow S''$); otherwise one proceeds with the next neighborhood. This procedure is repeated as long as a neighborhood structure allows such iteration.

Figure 1. QoS-VNS-CS algorithm execution

5.2 A variable neighborhood descent for core selection problem

The main motivation behind the use of the VNS search algorithm to solve core selection problem is the use of several neighborhoods to explore different neighborhood structures systematically. Our goal is to break away from a local minima, this use is based on three facts:

- If node N_1 is a local minimum for one neighborhood structure N_K is not necessary so with another one$N_{K'}$.
- A global minimum solution S is a local minimum for all possible neighborhood structures.
- For the core selection problem local minima to all neighborhood structures is relatively close and localized in the same place.

In this section, we provide a detailed description of the Variable Neighborhood Search algorithm for core selection Problem with delay and delay variation guarantee QoS-VNS-CS, and his three process phases: the initialization process, Stopping conditions phase and the shaking step.

5.3 Fitness function

The goal of this paper is to propose an algorithm which produces multicast trees with low cost, multicast delay and delay variation. The proposed algorithm consists of a core node selection as a first part of the multicast tree construction. Core selection problem in multicast routing consist in finding an optimal best position of core such that the tree cost, multicast delay and delay variation can be optimal. These parameters must be optimized from all sources to each multicast member. Unlike the first work (D2V-VNS-RPS [38]) we introduce in this work the MODE and fitness functions: MODE function [36] is used to know the location of core and a fitness function to measure the potential cost, delay and delay variation, this function is an enhanced version of the CMP function [36], [37].
The MODE function is presented in formula (9) and can take 5 values: I) if the candidate core is one source, II) if one destination node exist in the path between the candidate core and one source, III) if one source exist in the path between the candidate core and one destination node, IV) if II and III, and V) otherwise, where $s \in S$ is a source node, $d \in D$ is a destination node and $P_{min}(u, v)$is the least delay path from node u to v.

$$\text{MODE (C)} = \begin{cases} I & f \text{ core } = \text{ s } \forall s \in S \\ II & \text{if } \exists \text{d} \in \text{D} \setminus \text{d} \in P_{min}(\text{s, C}) \ \forall s \in S \\ III & \text{if } \exists \text{s} \in \text{S} \setminus \text{s} \in P_{min}(\text{d, C}) \ \forall d \in D \\ IV & \text{if II and III} \\ V & \text{otherwise} \end{cases} \quad (9)$$

The fitness Function is presented in formula (10), where $max^* = \max\{D(P_{min}(C, d))\} \mid d \in D \setminus \{d^* \in M \mid$

$$\exists d^* \in P_{min}(s, C) \quad \forall s \in S \quad V \quad \exists s \in P_{min}(d^*, C)\}\}$$

And $min^* = \min\{D(P_{min}(C, d))\} \mid d \in D \setminus \{d^* \in M \mid$

$$\exists d^* \in P_{min}(s, C) \quad \forall s \in S \quad V \quad \exists s \in P_{min}(d^*, C)\}\}$$

$$\begin{cases} \left| \sum_{s \in S} D(s, C) + max^* - min^* \right| \\ if\ MODE(c) = II\ or\ III\ or\ IV \\ \max\{D(P_{min}(C, d))\} - \min\{ D(P_{min}(C, d))\} \\ \forall d \in M \quad Otherwise \end{cases} \quad (10)$$

5.4 Initial solution

The first step of variable neighborhood search is to define an initial solution. Many methods can be used to generate this solution; the simplest is to select randomly one node in the network as initial solution. There are other methods that try to reduce the selection area and generate an initial solution from an ordered set of multicast group members. In this paper we use the selected node as a RP by the bootstrap Protocol as an initial solution.

Neighborhood structuresQoS-VNS-CS uses neighbor nodes concept to generate neighborhood structures: a node u is neighbor of another node v if an edge $e(u, v)$between u and v exists. We propose to compute a neighborhood structure Nj through the following formula (with neighbor(S) a set of neighbor nodes of S):

$$N_j(S) = \begin{cases} neighbor(S) & if \quad j = 1 \\ x \in N_{j-1}(S)\ N_1(x) & else \end{cases}$$

5.5 Shaking

From an initial solutionS, the shaking function varies and explore a new parts of the search space in random manner. It chose other solution $S'$from the k^{th}neighborhood structure$N_k(S)$. For this reason, Neighborhood structures are ranked in such a way that VNS algorithm explores increasingly further away from S'. After exploring search space thoroughly with a local search algorithm, the best local solution, S" is compared with S. If S" is better than S, it replaces$S$ ($S \leftarrow S$")and the algorithm starts all over again with $k = 1$. Otherwise, kis incremented and algorithm continues from shaking phase with next neighborhood structure. This function of shaking is terminated after all neighborhood structures are exhausted. The result of each Shake, S' is used as the starting point for the next Local Search.

5.6 Local search phase

In recent years, several local search algorithms have been proposed; they proceed from an initial solution S' generated randomly from a neighborhood structure and trays to find an optimum local solution S'' by a sequence of local changes and an iterative fashion by successively replacing the current solution by a better solution in the neighborhood of the current solution, which improve each time the value of the objective function.

The QoS-VNS-CS is independent of the local search algorithm used; it can work with hill climbing, adaptive multi-start, variable depth search, simulated annealing, Tabu search (TS), GRASP and others such as genetic search.

In this paper we adapt the GRASP-RP algorithm [32] to act as a local search algorithm. First proposed by Feo and Resende, the basic idea of this meta-heuristics, as presented in Algorithm 1, is to create a new solution iteratively from an initial solution generated by QoS-VNS-CS, independent of previous ones, where each iteration consists of two phases: first one is a construction phase using a randomized greedy algorithm, the second phase is a local search phase act just on neighbor node set, which improves eachtime the value of the cost function defined in formula (6).

The construction phase is a non-deterministic phase allows to diversify the search and to produce a feasible solution that is used as the starting point for local search. This first phase leads to the creation of a restricted candidate list (RCL) formed by the best starting solutions.

Algorithm 1: GRASP Pseudo code

```
GRAS(max_ iterations)
for(I ← 0; i< max_ iterations; i++)
Solution ← Greedy Randomized Construction(Seed);
Best_Solution← solution;
Node[] == Neigboor(Solution);
for(inti=0; I Node.legth(), i++)
    BestSolution←min(opt_F(Node[i]),opt_F(Best_
    Solution));
end;
return Best Solution;
end GRAS.
```

5.7 QoS-VNS-CS algorithm and pseudo code

In this section, a step by step QoS-VNS-CS-search-based algorithm for Core selection problem is presented, also we present a pseudo code in Algorithm 2.

Step 1: Declaration of provided information according to the network graph G.

Step 2: Set maximum iteration number of VNS, maximum number of iteration of stable solution, maximum iteration number of local search method...

Step 3: Select initial solution$S$.

Step 4: Choose the K_{max} scalar, select the set of neighborhood structuresN_k, for $k = 1, \cdots, k_{max}$, that will be used in the search; choose a stopping condition.

Step 5: Shaking phase: Take at random a solution S from$N_K(S)$.

Step 6: Local search phase: execute a local search algorithm, such as tabu search [29], GRASP [34] ..., to produce a local optimal solution S.

Step 7: Check if objective function value of solution S'' is less than objective function value of solution S, then move to S'' solution and continue the search with N_K $(K \leftarrow 1)$ from step 4; otherwise, set $K \leftarrow K + 1$ and also continue the search from step 4;

Step 8: Output the best solution core selected.

Step 9: Waiting for a recovery event

Algorithm 2: QoS-VNS-CS Pseudo code

```
Input: i = 0
Input: totalIt= 0
/*number of iteration of stable solution */
Input: maxItWithoutImprovement
Input: initialSolution
Input: Solutionsheking
Input: Solutionlocal
Best_Solution←initial_Solution;
whilei<maxItWithoutImprovement&&totalIt< max it
do
    lastCost← fitnessFunction(Best_Solution);k ← 0 ;
    while current it < max it &&k <k_max do
      if totalIt> max it then
          break ;
```

```
7        end
8        getN_K(s) ;
9        Solutionsheking ← getRandomNk(s) ;
10       Solutionlocal←localSearchMethod(Solutionsheking);
11       if fitnessFunction(Solutio local)
>fitnessFunctionFunctions(s)
then
12         k ← k+ 1 ;
13        else
14         s ← Solution local;
15         k ←0 ;
16        end
17        totalIt← totalIt+ 1;
18        current it ←current it + 1;
19     end
20     if lastSCost>fitnessFunction(s)) then
21        i ←0;
22     else
23        i ←i+ 1;
24     end if
25   end while
26   return s ;
27 end
```

5.8 Complexity study

The complexity of QOS-VNS-CS algorithm is explained line by line in the following. Line 3 is initialization statement. Their complexity is $O(1)$. Line 2 and 4 are a judgment statements of the while loop, and its complexity is $O(1)$. Line 3 is initialization statement. Their complexity is $O(1)$. Lines 5 - 7 are judgmentstatements and their complexity is $O(1)$. Line 8 generate the kth neighborhood structures, the complexity is $O(|\mathrm{neighborhoodstructures}| * \mathrm{k}$, the average value of $|\mathrm{neighborhoodstructures}|$ is $(2|\mathrm{E}|/|\mathrm{N}),$ then the total complexity is $O((2|\mathrm{E}|/|\mathrm{N}|)\mathrm{k}) < \mathrm{O}(|\mathrm{E}|)$. Line 9 select randomly one solution from$N_k(S)$, their complexity is $O(1)$. Like 10 is affectation statement their complexity is $O(1)$. Line 11 computes and compare weight function and their complexity is $O(|\mathrm{S}| + |\mathrm{D}|))$. Lines 12 - 19 correspond to assignment, their complexity is $O(1)$. Then the complexity is$O(|N|\big(O(1) + O(|E|) + O(1) + \ O(|\mathrm{S}| + |\mathrm{D}|) + O(1)\big) = O\big(3|N| + |N||E| + |N|(|S| + |D|)\big)$. Line 26 is a return statement, and its complexity is $O(1)$. Therefore, the complexity of the algorithm is$O(1) + O\big(3|N| + |N||E| + |N|(|S| + |D|)\big)$, that is $O\big(3|N| + |N||E| + |N|(|S| + |D|)\big)$.

6. Simulation Results

In this section, we use simulation results to demonstrate the effectiveness of the proposed algorithm described above. To study the performance of our selection algorithm QoS-VNS-CS, we implement it in a simulation environment; we use the network simulator NS2 [39]. The random graph generator GT-ITM [40] is used to generate a random different 100 networks, and we adopt Waxman [41] as the graph model. Our simulation studies were performed with a 100 runs. The values of $\alpha = 0.2$ and $\beta = 0.2$ were used to generate networks with an average degree between 3 and 4 in the mathematical model of Waxman.

To demonstrate the performance of this algorithm (QoS-VNS-CS), we compare it with the following algorithms, including, Tabu RP selection (TRPS) [37], DDVCA [6], AKC [36] and D2V-VNS-RPS[38].

The main objective of our algorithm is to reduce delay and delay variation; therefore, we start the simulation results by comparing these two metrics. And then, we compare tree cost and construction tree delay.

End-To-End delay is the time delay from the source node to the furthest receiver node in the multicast group. In Figure 2 the end-to-end Delay is plotted as a function of the number of nodes in the network topology, for this we use a topology network with multicast groupmember's size is 10 % of the overall network nodes. Simulation results show that QoS-VNS-CS is the best among all the algorithms on average delay, it decreases more the end-to-end delay to transmit multicast packet, with D2V-VNS-RPS [38], TRPS [37] and AKC [36] following it, and DDVCA [6] is the worst.

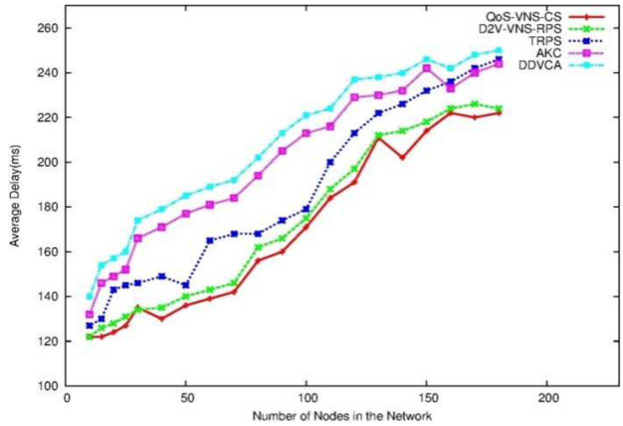

Figure 2. Comparison of Delay VS Network Size

Delay Variation is the difference between the first time of the reception of a multicast packet by a receiver of the multicast group and the last reception of the same multicast packet by another receiver of the multicast group. In Figure 3 and Figure 4 the Delay Variation is plotted as a function of the number of nodes in the network topology with a multicast group member's size are 10 % and 20% of the overall network nodes respectively. Simulation results show that multicast trees build by our proposed algorithm have an average multicast delay variation better than D2V-VNS-RPS [38], TRPS [37], AKC [36], and DDVCA [6]algorithms and support more multicast members.

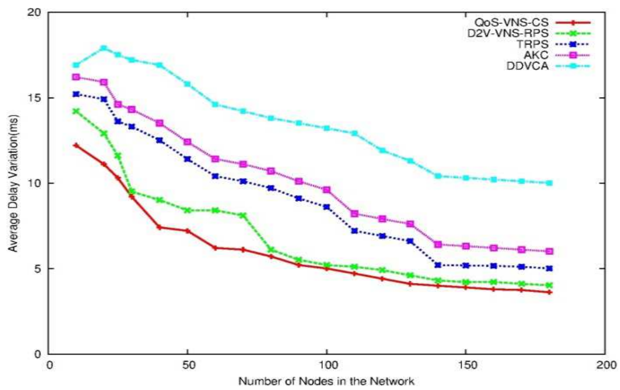

Figure 3. Comparison of Delay Variation VS Network Size with 10% nodes as destinations.

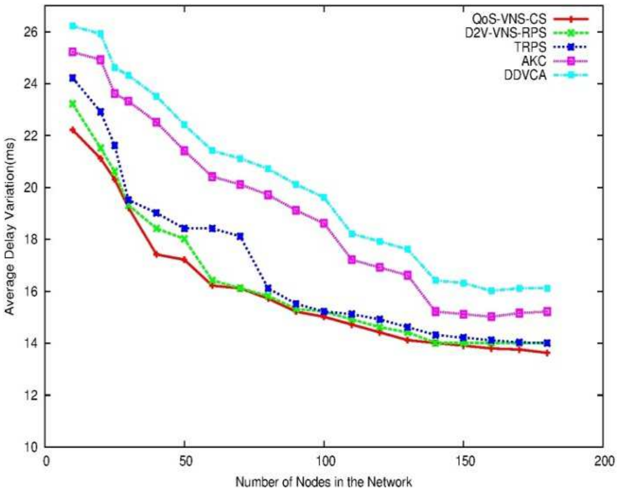

Figure 4. Comparison of Delay Variation VS Network Size with 20% nodes as destinations.

Based on the cost function in the formula (3), Figure 5 presents a comparison study of multicast tree Cost generated by each algorithm, in this simulation, we use a topology network with multicast group member's size as 10 % of the overall network nodes, the performance of DDVCA [6] selection is the worst, followed by AKC [36], TRPS [37] and D2V-VNS-RPS [38], QoS-VNS-CS shows better performances, and it has the minimal cost.

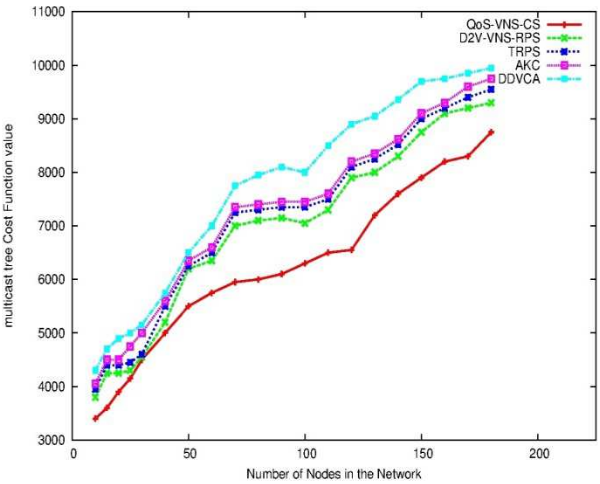

Figure 5. Comparison of Multicast Tree Cost VS network size

With a simulation topology generated containing 10% node as group members, we compare construction tree delay. This metric is computed as the time required to build all branches of the multicast tree after receiving all membership requests sent by the receivers. Simulation results presented in Figure 6 shows that QoS-VNS-CS outperforms all others algorithms in construction tree delay constraints when multicast group are widely localized.

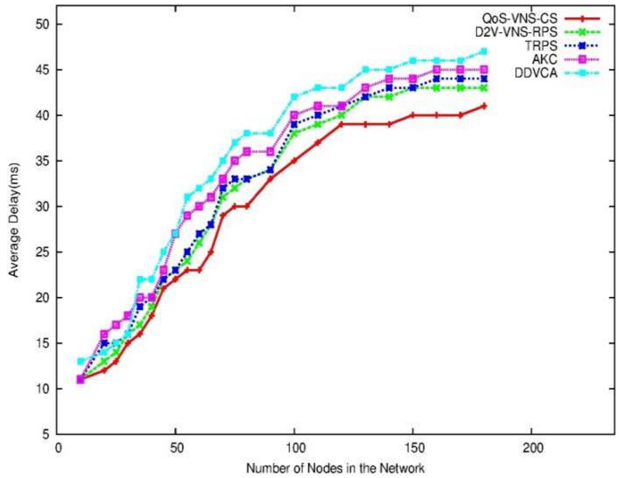

Figure 6. Comparison of Delay Tree Construction VS network Size

Figure 7 shows a study of scalability relative to the number of supported group by each algorithm, the networks used for this study consist of a random network of 100 nodes in about 10 % of multicast group members (5 % of sources and 5 % of receivers), we note that all algorithms gives same results in small topology, but our algorithm is more scalable when the number of groups is important.

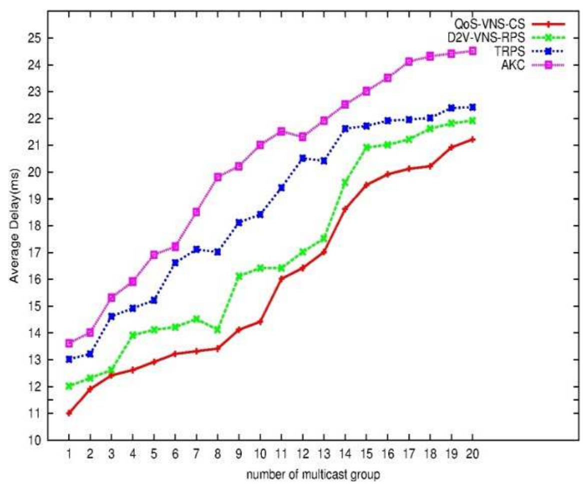

Figure 7. Comparison of Delay VS number of multicast group

Figure 8 shows a study of scalability relative to the number of source supported by each algorithm, the networks used for this study consist of a random network of 100 nodes in about 10 % member of the multicast group and one multicast group, our algorithm shows more scalability of number of multicast sources supported, followed by the D2V-VNS-RPS[38] algorithm, and in lastplace we find AKC [36].

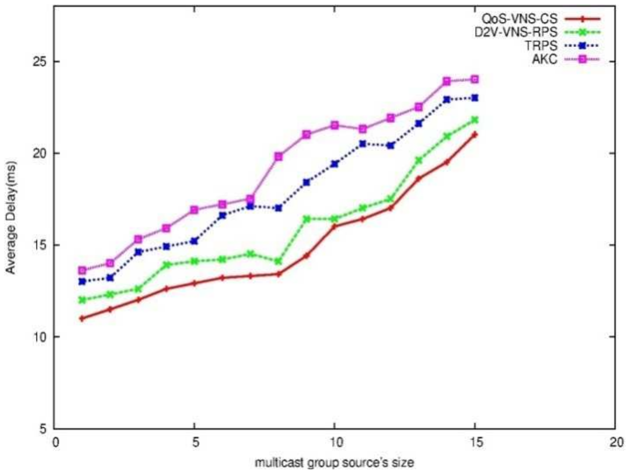

Figure 8. Comparison of Delay VS number of multicast source

7. Conclusion

Core selection problem in multicast routing protocol, using core-based tree to forward multicast data, affects directly the structure of the tree and the performance of the routing scheme of multicast accordingly. Current algorithms (PIM-SM [3] and CBT [4]) decide on core router administratively, which leads to high cost, high delay, and high congestion. Supporting core selection algorithm with QoS guaranties is a serious enough issue to more merit attention. To solve these problems, QoS-VNS-CS algorithm is proposed based on VNS [8] heuristic algorithm. To present our proposition, we started with a brief overview of multicast routing protocols and two types of multicast trees SBT and CBT. We reviewed and analyzed the cost and delay function for rendezvous selection algorithms. We reviewed the core selection algorithms studied in literature for their algorithmic structures. To test the effectiveness of our algorithm we compared it with a number of other commonly used core selection algorithms. Simulation results show that our algorithm presents good performances in multicast cost, delay, delay variation and other aspects. Our future work is focused on extending this algorithm to support others multiple QoS criteria imposed by receivers across the network.

References

[1] S. E. Deering, “Multicast Routing in Internetworks and Extended LANs,” Stanford University, Stanford, CA, USA, août 1988.

[2] D. Zappala and A. Fabbri, “An Evaluation of Shared Multicast Trees with Multiple Active Cores,” in ICN ’01: Proceedings of the First International Conference on Networking-Part 1, London, UK, 2001, pp. 620–629.

[3] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised),”IETF RFC Editor, United States, 2006.

[4] A. Ballardie, “Core Based Trees (CBT version 2) Multicast Routing – Protocol Specification –,” IETF RFC Editor, United States, 1997.

[5] G. N. Rouskas and I. Baldine, “Multicast Routing with End-to-End Delay and Delay Variation Constraints,” North Carolina State University at Raleigh, Raleigh, NC, USA, 1995.

[6] P.R. Sheu and S.T. Chen, “A Fast and Efficient Heuristic Algorithm for the Delay- and Delay Variation Bound Multicast Tree Problem,” in Proceedings of the 15th International Conference on Information Networking, Washington, DC, USA, 2001.

[7] M. Ramalho, “Intra- and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy,”IEEE Commun. Surv. Tutor., Vol. 3, No 1, pp. 2–25, 2000.

[8] P. Hansen, N. Mladenović, and L. C. D. Gerad, “Variable neighborhood search: Methods and recent applications,” in Proceedings of MIC’99, 1999, pp. 275–280.

[9] K. Mehlhorn, «A faster approximation algorithm for the Steiner problem in graphs,”Inf Process Lett, Vol. 27, No 3, pp. 125–128, 1988.

[10] K. L. Calvert, E. W. Zegura, and M. J. Donahoo, “Core Selection Methods for Multicast Routing,”ICCCN, pp. 638–642,1995.

[11] N. Meghanathan, “Survey of topology-based multicast routing protocols for mobile ad hoc networks,”International Journal of Communication Networks and Information Security (IJCNIS), Vol. 3, No 2, pp 124-137, 2011.

[12] A. Karaman and H. Hassanein, “Core-selection algorithms in multicast routing - comparative and complexity analysis,”Comput Commun, Vol. 29, No 8, pp 998–1014, 2006.

[13] L. Wei and D. Estrin, “A Comparison of Multicast Trees and Algorithms,” 1994.

[14] A. Karaman and H. Hassanein, “QoS-constrained core selection for group communication,”Comput Commun, Vol. 30, No 7, pp 1600–1612, 2007.

[15] D. Waitzman, C. Partridge, and S. E. Deering, “RFC 1075: Distance Vector Multicast Routing Protocol,”IETF RFC Editor, United States,1988.

[16] J. Moy, “MOSPF: Analysis and Experience,”IETF RFC Editor, United States, 1994.

[17] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised),” IETF RFC Editor, United States, 2005.

[18] S. B. Shukla, E. B. Boyer, and J. E. Klinker, “Multicast Tree Construction in Network Topologies with Asymmetric Link Loads,” 1994.

[19] N. Bhaskar, A. Gall, J. Lingard, and S. Venaas, “Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM),”IETF RFC Editor, United States, 2008.

[20] F. Carrabs, G. Laporte, J. F. Cordeau, and Q. C. for R. on T. (Montréal, “Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading,”Centre for Research on Transportation (C.R.T.), 2005.

[21] P. Hansen and N. Mladenovic, “Variable neighborhood search: Principles and applications,”Eur. J. Oper. Res., Vol. 130, No 3, pp. 449–467, 2001.

[22] P. Hansen, N. Mladenovic, and J. A. Moreno-Pérez, “Variable neighbourhood search: methods and applications,”Ann. OR, Vol. 175, No 1, pp 367–407, 2010.

[23] E. Burke, P. D. Causmaecker, S. Petrovic, and G. V. Berghe, “Variable neighborhood search for nurse

rostering problems," Norwell, MA, USA: Kluwer Academic Publishers, 2004, pp. 153–172.

[24] M. Sevkli and M. E. Aydin, "Variable Neighbourhood Search for Job Shop Scheduling Problems,"JSW, Vol. 1, No 2, pp. 34–39, 2006.

[25] J. Akpojaro, P. Aigbe, and D. Oyemade, "A cost-based approach for analysing the overheads of multicast protocols in non-strictly hierarchical networks,"International Journal of Communication Networks and Information Security (IJCNIS), Vol. 3, No 1, pp. 67-73, 2011.

[26] C. Shields and J. J. Garcia-Luna-Aceves, "The Ordered Core Based Tree Protocol," in Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, Washington, DC, USA, 1997.

[27] D. W. Wall, "Mechanisms for Broadcast and Selective Broadcast," Stanford University, Stanford, CA, USA, 1980.

[28] W. Hua, M. Xiangxu, Z. Min, L. Yanlong, and others, "Tabu search algorithm for RP selection in PIM-SM multicast routing,"Comput Commun, Vol. 33, No 1, pp. 35–42, 2010.

[29] F. Glover, "Tabu Search - Part II,"Inf. J. Comput., Vol. 2, No 1, pp. 4–32, 1990.

[30] Y. Baddi and M. D. E. Kettani, "VNS-RP algorithm for RP selection in multicast routing protocol PIM-SM," Tangier, Morocco, 2012.

[31] Y. Baddi and M. D. E. Kettani, "VND-CS: A Variable Neighborhood Descent Algorithm for Core selection Problem in multicast routing protocol," Dubai, India, 2012.

[32] Y. Baddi and M. D. E. Kettani, "GRAS-RP: Greedy Randomized Adaptive Search algorithm for RP selection in PIM-SM multicast routing," in Seventh International Conference on InTelLigent Systems: Theories and Applications (SITA'12), Mohammedia, Morocco, 2012.

[33] M. Tagmouti, M. Gendreau, and J.-Y. Potvin, "A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs,"Comput Ind Eng, Vol. 59, No 4, pp. 954–963, 2010.

[34] T. A. Feo and M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures,"J. Glob. Optim., Vol. 6, pp. 109–133, 1995.

[35] M. Kim, Y.-C. Bang, H.-J. Lim, and H. Choo, "On Efficient Core Selection for Reducing Multicast Delay Variation under Delay Constraints,"IEICE Trans., Vol. 89-B, No 9, pp. 2385-2393, 2006.

[36] S. Ahn, M. Kim, and H. Choo, "Efficient Algorithm for Reducing Delay Variation on Delay-Bounded Multicast Trees in Heterogeneous Networks," in WCNC 2008, IEEE Wireless Communications and Networking Conference, March 31 2008 - April 3 2008, Las Vegas, Nevada, USA, Conference Proceedings, 2008, pp. 2741–2746.

[37] S. P. Sahoo and M. R. Kabat, "Tabu Search Algorithm for Core Selection in Multicast Routing," in Proceedings of the 2011 International Conference on Communication Systems and Network Technologies, Washington, DC, USA, 2011, pp. 17–21.

[38] Y. Baddi and M. D. E.-C. El Kettani, "D2V-VNS-RPS: Delay and delay variation constrained algorithm based on Variable Neighborhood Search algorithm for RP Selection problem in PIM-SM protocol," in 2012 International Conference on Complex Systems (ICCS), 2012, pp. 1-6.

[39] T. Issariyakul and E. Hossain, "Introduction to Network Simulator NS2," Springer Publishing Company, Incorporated, 2008.

[40] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network Topologies, Power Laws, and Hierarchy", 2001.

[41] B. M. Waxman, "Routing of multipoint connections,"Sel. Areas Commun. IEEE J. On, Vol. 6, No 9, pp. 1617–1622, 2002.