ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF INFORMATICS

VIRTUAL TOPOLOGY DESIGN FOR OPTICAL NETWORKS USING THE MANHATTAN STREET NETWORK AND THE SHUFFLENET

M.S. Thesis by

Burak USLU

Department : Advanced Technologies

Programme : Computer Science

Supervisor: Assoc. Prof. Sema OKTUĞ

MAY 2004

PREFACE

First of all, I would like to thank my supervisor Assoc. Prof. Sema OKTUĞ for her valuable contributions, guidance and patience throughout this thesis work. I would also like to thank my colleagues Tolga ÇÖPLÜ and Mutlu ÖNDER for their help. Finally, I am grateful to my wife and my family who have always been there to support and encourage me.

May 2004 Burak USLU

CONTENTS

ABBREVIATIONS

WDM : Wavelength Division Multiplexing
MSN : Manhattan Street Network
NP : Non-Deterministic Polynomial-Time
PMX : Partially Mapped Crossover
CX : Cyclic Crossover
SA : Simulated Annealing
TWEID : Traffic-Weighted Embedded Internodal Distance
OXC : Optical Cross Connect
MLL : Mean Lightpath Length
NPO : Node Placement Optimization
ONAP : Optimal Node Assignment Problem
QAP : Quadratic Assignment Problem
PI : Performance Improvement
PD : Performance Degradation
t1 : Uniform Random Traffic
t2 : Uniform Random Traffic with Amplitude=90
t3 : Non-Uniform Random Traffic with Skew Factor γ=100
t4 : Varying Range Traffic
R : Range
t4(20) : Varying Range Traffic with R=20
t4(50) : Varying Range Traffic with R=50
t4(90) : Varying Range Traffic with R=90

LIST OF TABLES

Page Number

LIST OF FIGURES

Page Number

VIRTUAL TOPOLOGY DESIGN FOR OPTICAL NETWORKS USING THE MANHATTAN STREET NETWORK AND THE SHUFFLENET

ABSTRACT

In this thesis, two multiprocessor interconnection architectures, the Manhattan Street Network (MSN) and the Shufflenet, are adopted to be deployed as regular virtual topologies in arbitrary physical networks. We employ node placement optimization for optimal deployment and we use the iterative and simulated annealing algorithms since node placement optimization is a combinatorial optimization problem. The performance of the heuristics and the regular virtual topologies is evaluated under different traffic patterns. In addition, the impact of traffic and network size on the performance of algorithms is assessed. We also propose a best-case analysis for 8-node MSN and Shufflenet to evaluate how near the results obtained by heuristics are to the global minimum for the given traffic pattern. The experimental results show that simulated annealing algorithm outperforms iterative algorithm for 24 and 64-node MSN and Shufflenet while iterative algorithm is better for 8-node MSN and Shufflenet. It is encouraging to see that the performance of the employed heuristics tremendously increase when the traffic load becomes non-uniform. It is worth to note that the Shufflenet is more advantageous for 64-node topologies. For the remaining network sizes, the Shufflenet seems to be slightly better considering the overall simulation results.

MANHATTAN STREET NETWORK VE SHUFFLENET KULLANARAK OPTİK AĞLAR İÇİN SANAL TOPOLOJİ TASARIMI

ÖZET

Bu tezde, çoklu işlemci ara bağlantı mimarilerinden olan Manhattan Street Network ve Shufflenet fiziksel ağlar üzerinde düzenli sanal topoloji olarak konumlandırılmak üzere seçilmişlerdir. Bu konumlandırmanın en uygun şekilde olarak yapılabilmesi için düğüm konumlandırma eniyilemesi metod olarak benimsenmiştir. Bu metodun bileşimsel bir eniyileme metodu olması nedeni ile tekrarlamalı ve benzetimli tavlama sezgisel metodları en uygun konumlandırmayı bulmak üzere kullanılmıştır. Bu çalışmada, uyguladığımız sezgisel yöntemlerin başarımları kadar kullandığımız düzenli sanal topolojilerin de başarımları değişik trafik yükleri altında değerlendirilmiştir. Buna ek olarak, trafiğin ve ağ boyutunun ağ başarımı üzerine etkisi de incelenmiştir. Ayrıca , sezgisel yöntemlerinin en iyi konumlandırmaya ne kadar yakın olduklarını görmek için 8 düğümlü MSN ve Shufflenet için en iyi durum analizleri de sunulmuştur. Yaptığımız deneysel sonuçlardan, taklit tavlama yönteminin 24 ve 64 düğümlü MSN ve Shufflenet için, tekrarlamalı yöntemin de 8 düğümlü MSN ve Shufflenet için daha uygun olduğunu çıkarıyoruz. Trafik yükünün düzensizleşmesi ile beraber başarımın büyük oranda arttığını görüyoruz. Shufflenet, 64 düğümlü topolojiler için daha iyi avantajlı görünüyor. Diğer ağ boyutları için benzetim sonuçları genel olarak değerlendirildiğinde, Shufflenet çok az da olsa Manhattan Street Network'ten daha iyi sonuçlar veriyor.

1 INTRODUCTION

In recent years, there is an obvious demand for more bandwidth due to multimedia applications over the network such as real-time voice/video, distributed databases and distributed computing. Since fiber optic media provides tremendous bandwidth, it seems inevitable to benefit from optical networks as the transmission media. However, it is clear that the electronic processing units are not capable of utilizing a fiber line in terms of bandwidth. Although it is not possible for the electronic devices to reach the speed of a fiber optic transmission medium, it is possible to divide a fiber into wavelengths resulting in each wavelength to have a smaller bandwidth than that of the fiber itself. This emerging technology which divides a fiber into wavelengths is known to be Wavelength Division Multiplexing(WDM). Thanks to this technology, the bottlenecks caused by the electronic processing units are by-passed.

For WDM optical networks, wavelength assignment is as crucial as routing since have to choose a wavelength on the fiber as the transmission channel. Nonetheless, the construction of lightpaths is another crucial point for WDM optical networks. Taking into account that a lightpath can consist of one or more wavelength channels, it is a problem to decide among which node pairs the lightpaths should be established. The lightpaths can be established arbitrarily or the lightpaths can be established optimally with respect to an objective function. This problem is called Virtual Topology Design Problem which we are going to focus on in this work.

As mentioned in the previous paragraph, Virtual Topology Design Problem is how to establish lightpaths between nodes optimally and this problem has three input parameters: the traffic matrix, the physical topology, and the virtual topology [1]. We are already familiar with the physical topology and its corresponding traffic matrix. Here, we come across with the term “virtual topology”. Usually, virtual topologies are famous regular architectures with simple and distributed routing schemes and they are capable of avoiding or at least minimizing packet contention which enforces buffering [1]. The reason for avoiding packet contention is: 1) it is not practical to

implement buffers in the optical domain, and 2) optic-to-electronic and electronic-to-optic transformation is not tolerable since electronic devices are much slower than optical devices [2]. The regular topologies, which are appropriate to form regular virtual topologies, include the Shufflenet [3], [4], the de Bruijn Graph [4], [5], the Hypercube [4], [6], [7], the Manhattan Street Network (MSN) [8], [9] and the Kautz Graph [10].

There are three methods in solving virtual topology design problem. Each of these algorithms use different combination of 3 input parameters. These methods are the direct method, the dilation minimization method, and the node placement optimization method. The direct method takes physical topology, regular virtual topology and the corresponding traffic matrix as inputs whilst the dilation minimization method discards the traffic matrix and the node placement optimization method discards the physical topology [1]. Regular virtual topology as an input is essential for any of the methods.This is because all three methods refer to the regular virtual topology while establishing lightpaths between node pairs of the given topology.

Although we present the solution methods and the input parameters required by the methods, it is still not possible to reach a solution easily since virtual topology design problem is known to be *NP*-hard [4], [11], [12]. That's why we refer to heuristics to obtain near-optimal solutions. Some of the heuristics that can be applied are hill climbing [1], random search [1], simulated annealing [19], various implementations of genetic algorithms such as partially mapped crossover (PMX) and cycle crossover (CX) [1] and iterative algorithm [13], [14].

In this work, we are going to compare the performance of two regular virtual topologies-the MSN and the Shufflenet- and two heuristic methods- the iterative algorithm and the simulated annealing algorithm- under different traffic patterns. The dimensions of the MSN are 2x4, 4x6, and 8x8 while the dimensions of the Shufflenet are (2,2), (2,3) and (2,4).The solution method chosen is node placement optimization method. As the network size grows, we observe that the simulated annealing algorithm achieves better results than the iterative algorithm. Moreover, it is worth to note that the results obtained under non-uniform random traffic patterns are encouraging when compared to that of obtained under uniform random traffic. It is also obvious that (2,4) Shufflenet outperforms 8x8 MSN. Since the topologies, 2x4

MSN and (2,2) Shufflenet, are not so large in size, we also propose a best case analysis in order to assess the performance of the heuristics on the two regular virtual topologies. This thesis is organized as follows:

Section 2 introduces the design approaches. Mathematical formulations, objective functions and input parameters for direct, dilation minimization and node placement optimization methods are given in detail. The physical equivalent of each objective function, advantages and disadvantages of each approach are also provided. Moreover, the regular virtual topologies that are of interest in this work, the Manhattan Street Network and the Shufflenet, are introduced. The two topologies are illustrated and detailed information about their structures and routing schemes are given.

Section 3 briefly gives information about the heuristics in literature. Detailed information on iterative algorithm and simulated annealing is given as well as the mathematical basis of the algorithms. In addition, time complexity of iterative algorithm is taken into consideration.

In Section 4, our work is introduced. Traffic models, simulation principles, and best case analysis are given in detail. Moreover, simulation results are given and the performance of the regular virtual topologies and heuristics applied are assessed based on these results.

In Section 5, conclusion and future work are given.

2 VIRTUAL TOPOLOGY DESIGN APPROACHES, METHODS AND REGULAR VIRTUAL TOPOLOGIES

2.1 Virtual Topology Design Approaches and Methods

As we have mentioned in Section 1, virtual topology design problem simply is how to establish lightpaths optimally between nodes in an arbitrary physical topology. However, in literature virtual topology design can be approached in two ways. First, the connectivity of the virtual topology is already defined and the problem is to embed this virtual topology in a physical topology with respect to an defined cost. Second, the connectivity of the virtual topology is not critical since virtual topology is the output of the design process [1]. In this work, we deal with the first approach and try to embed a regular virtual topology in an arbitrary physical topology optimally.

Methods-direct, dilation minimization and node placement optimization-given in Section 1 approach the virtual topology design problem as an embedding problem, in other words node matching problem. In this section, we go into the details of these three methods. First, it is beneficial to illustrate the mathematical formulation network model [1].

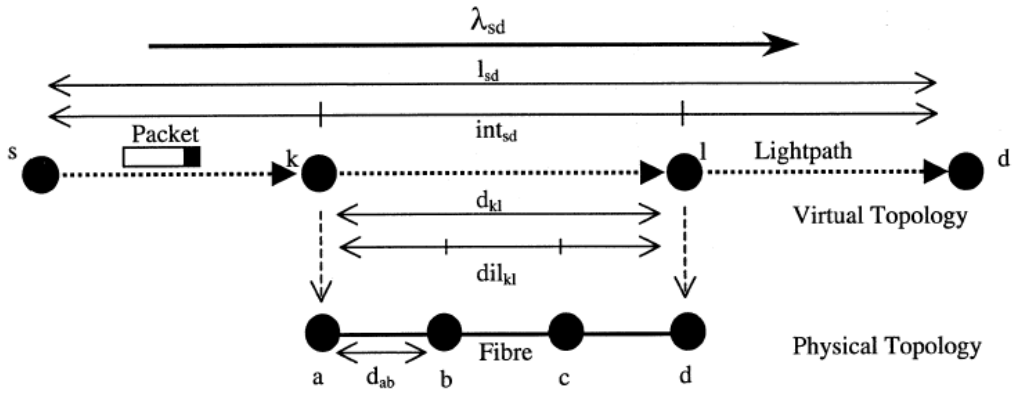

Figure 2.1 Mathematical Network Model [1]

As it is clear from Figure 2.1, virtual topology seems to be the upper layer and the physical topology is the underlying layer which forms the lightpaths between virtual

nodes. We can also say that a lightpath may traverse one or more fibers and a virtual node pair may be connected to each other through one or more lightpaths.In addition, the traffic load from one virtual node to another depends on the matching of physical and virtual nodes.

2.1.1 Direct Method

This method takes the traffic matrix, the physical topology and the regular virtual topology as inputs and tries to minimize the mean traffic-weighted embedded internodal distance (TWEID) which is the cost defined for this method. This cost is the indicative of both number of optical cross-connects (OXCs) and delay in physical domain [1]. The mathematical formulation of TWEID [1] is;

$$TWEID = \frac{1}{N(N-1)} \sum_{s,d} \left\{ \lambda_{sd} \sum_{k,l} \left[v_{kl}^{sd} \sum_{a,b} \left(u_{ab}^{kl} d_{ab} \right) \right] \right\} \qquad (2.1)$$

where

$\lambda_{sd} =$ Traffic between physical nodes onto which virtual nodes s and d are mapped

$$v_{kl}^{sd} = \begin{cases} 1 & \textit{if a packet from } s \textit{ to } d \textit{ traverses the lightpath from } k \textit{ to } l \\ 0 & \textit{otherwise} \end{cases}$$

$$u_{ab}^{kl} = \begin{cases} 1 & \textit{if the lightpath from } k \textit{ to } l \textit{ is routed over the fiber between node } a \textit{ and } b \\ 0 & \textit{otherwise} \end{cases}$$

$d_{ab} =$ Physical distance between node a and b

$N =$ Number of nodes

As shown in Figure 2.1, we can derive that a lightpath can be routed over one or more fibers and two nodes may be connected to each other via one or more lightpaths. From Equation 1, it is obvious that we focus on the implementation of a lightpath in the physical topology since we refer to the physical topology to obtain the physical distance of the lightpath by checking out the fibers the lightpath is routed over and physical distance of the traversed fibers.

TWEID is exactly what we have to minimize to reach an optimal solution in the physical world because it checks whether the construction of a lightpath is possible in the physical topology considering the physical constraints.

2.1.2 Dilation Minimization

This method takes virtual topology and physical topology as inputs but discards the traffic matrix. The reason for this is the neighbour nodes in the virtual topology should not fall apart when they are embedded in the physical topology regardless of the traffic load among them. Simulation results in [1] encourage this approach since the results obtained by dilation minimization method is rather close to the results obtained by direct method. Moreover, it eliminates the necessity of traffic matrix as well as the assumptions, predictions and calculations on the traffic matrix since it is traffic independent. This method tries to minimize mean lightpath length (MLL). In other words, it tries to minimize the dilation of virtual nodes after the embedding process. The mathematical formulation of MLL [1] is;

$$MLL = \frac{\sum_{k,l}\sum_{a,b}\left(u_{ab}^{kl} d_{ab}\right)}{\sum_{k}\Delta_k} \qquad (2.2)$$

where

$\Delta_k =$ Nodal degree of node k

$$u_{ab}^{kl} = \begin{cases} 1 & \textit{if the lightpath from } k \textit{ to } l \textit{ is routed over the fiber between node } a \textit{ and } b \\ 0 & \textit{otherwise} \end{cases}$$

$d_{ab} =$ Physical distance between node a and b

From Equation 1 and 2, we can see that this method is a decomposition of direct method which omits the traffic matrix hence the traffic loads between node pairs. Minimization of MLL results in the minimization of dilation of virtual nodes.

2.1.3 Node Placement Optimization

This method takes the traffic matrix and the virtual topology as inputs and tries to minimize the weighted mean hop distance ($\bar{h}$). The mathematical formulation of ($\bar{h}$) [15] is;

$$\bar{h} = \frac{1}{\Lambda}\sum_{s}\sum_{d}\lambda_{sd}\,\mathrm{int}_{sd} \qquad (2.3)$$

where

$$\Lambda = \sum_{i,j} \lambda_{ij}$$

$$\text{int}_{sd} = \sum_{k,l} v_{kl}^{sd}$$

$$v_{kl}^{sd} = \begin{cases} 1 & \textit{if a packet from } s \textit{ to } d \textit{ traverses the lightpath from } k \textit{ to } l \\ 0 & \textit{otherwise} \end{cases}$$

$\lambda_{sd} =$ Traffic between physical nodes onto which virtual nodes s and d are mapped

As it is obvious from Equation 3, node placement optimization (NPO) doesn't care about the implementation of a lightpath in the physical topology and it is a decomposition of direct method since it omits the embedding process. Therefore, the term "hop" refers to virtual link or lightpath. All virtual links are the same in length.

As a result, $\bar{h}$ is indicative of mean delay [1].

2.2 Regular Virtual Topologies

As mentioned in Section 1, regular virtual topologies are famous for their extremely simple routing schemes and their capabilities to avoid buffering in the optical domain. Therefore, regular virtual topologies are essential concepts in virtual topology designs. Although there is a number of well-known regular virtual topologies, we focus on two famous regular virtual topologies for this work, Manhattan Street Network and Shufflenet. In the following subsections, we detailly give information about the structures and routing schemes for both topologies.

2.2.1 Manhattan Street Network

Manhattan Street Network is one of the most attracting regular virtual topologies due to its novel routing scheme called "Clockwork Routing" . Due to this novel routing scheme, necessity for optical domain buffering and resequencing at destination nodes is eliminated [1].

In Clockwork Routing, all the nodes in an *n*x*n* MSN are synchronized by an global clock and the timeslots are organized in a modulo-*n* sequence of frames. Each of the nodes own a simple 2x2 cross-bar switch. Every node is in the cross state for the first *n*-1 timeslots in the timeframe it belongs to and in the bar state for the last timeslot in the timeframe it belongs to. Figure 3.1 illustrates the timeframe and timeslot

structure of a node which is a part of *2x2* MSN. By Inserting a packet into the correct timeslot on a particular output link, the packet is automatically routed to the destination. This results in no extra processing or buffering at the intermediate nodes which only determine whether the packet has reached its destination or not [1], [18]. This "for me or not for me" evaluation can be implemented in the optical domain [2], [16].

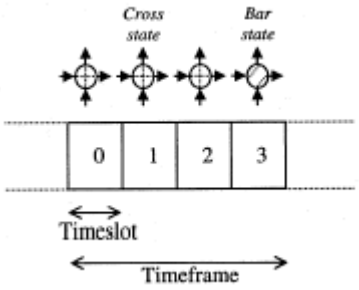

Figure 2.1 Structure of a Timeframe and Timeslot [1]

The dimension of a Manhattan Street Network is defined by two variables just like the dimensions of a matrix is defined. An mxn MSN consists of m rows each of which contains n nodes. Figure 3.2 shows what 2x4 MSN looks like.

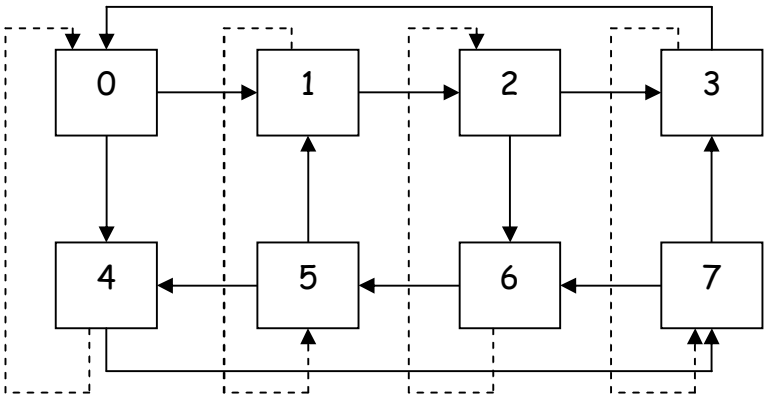

Figure 2.2 2 X 4 MSN

The inspiration of this regular virtual topology is the streets of Manhattan city itself. That's why this regular topology is called Manhattan Street Network.

2.2.2 Shufflenet

Shufflenet is another widely deployed regular virtual topology hence it has been frequently used for solving virtual topology design problem. Moreover, there is a number of heuristics developed especially for Shufflenet. Like the MSN, it has a very simple routing scheme.

The dimension of a Shufflenet is defined by parameters, Δ and k. (Δ,k) Shufflenet has k columns each of which contains Δ^k nodes . The nodes are labeled as follows;

($c, a_0a_1…..a_{k-1}$)

- c indicates the column number and c ε {0,1,….,k-1}

a_i ε {0,1,…., Δ-1} and i ε {0,1,…., k-1}

There is an edge from node i to node j in the following column if node j's string can be obtained from node i's string by one shift. ($c, a_0a_1…..a_{k-1}$) is connected to ((c+1) mod k, $a_0a_1…..a_{k-1}$*) where * ε {0,1,…., Δ-1} [17]. Figure 3.3 illustrates a (2,2) Shufflenet [17].

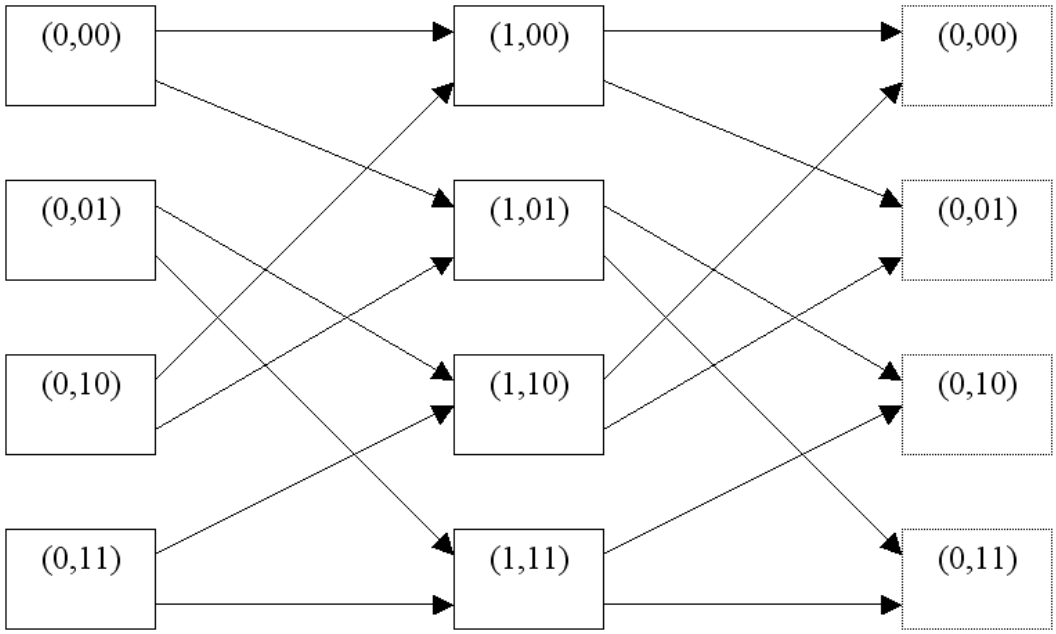

Figure 2.3 (2,2) Shufflenet

Routing in the Shufflenet is rather different than it is in MSN. Here, no clock synchronization is needed between the nodes. Nodes only decide which output should be chosen by checking the corresponding bit of the string with respect to the nodes' column number.

3 HEURISTICS

As stated in the first section, virtual topology design problem is known to be *NP*-hard. In addition, as stated in [19], virtual topology design problem which is given as Optimal Node Assignment Problem (ONAP) in the paper is a Quadratic Assignment Problem (QAP) and it is also *NP*-hard [20]. That's why, we are not capable of reaching a solution which minimizes the objective function although we have all the necessary input parameters and the objective function. However, this incapability is not valid for topologies with small dimensions. For example, for an 8-node topology, there are 8! possible matchings between physical and virtual nodes and it is possible to calculate the cost of each matching with respect to objective function thus we can obtain the matching which minimizes our defined cost. Unfortunately, this method doesn't work for larger topologies since the number of matchings tremendously increases due to factorial operator and the problem space is not tractable.

At this point, we refer to the heuristics to find near-optimal solutions. The results obtained by heuristics are (near)-optimal because these algorithms do not search the whole problem space due to their structures. Therefore, the result obtained by a heuristic algorithm is a local minimum which can be a global minimum as well. We can divide heuristic algorithms into two groups: 1)flow-based heuristics and 2)delay-based heuristics. Flow-based heuristics, greedy algorithms(Sorted First-Fit, First-Fit Supernodes, First-Fit on Binary Tree, Divide and Minimize Link Flow) and iterative algorithm [14], try to minimize the maximum flow on any link [21], [22]. On the other hand, delay-based heuristics, Load over Distance, Divide and Minimize Delay, Iterative algorithm, try to minimize the network-wide mean packet delay [14]. We see that the iterative algorithm can be interpreted as either flow- based or delay-based with minor changes in the algorithm. There are also some heuristics proposed in literature which are developed specially for a single regular virtual topology. In [13], global and local algorithms are proposed and these algorithms are designed depending on the structure of the Shufflenet. In addition, in [15], gradient algorithm which is a form of greedy algorithm, is proposed for Shufflenets. Furthermore, as

mentioned in Section 1, random search, hill climbing, simulated annealing and some implementations of genetic algorithms such as PMX and CX are some other heuristics employed in literature [1].

Hill climbing simply starts with a random initial embedding, in other words matching, and tries to improve the defined cost by moving to the neighborhood. The move is accepted provided that the defined cost is improved [1].

In random search, the cost function values of numerous randomly generated embeddings are calculated and the matching which improves the cost most is chosen [1].

Greedy algorithm which is designed for Shufflenet has a greedy approach to maximize the one-hop traffic in a Shufflenet. The elements of the traffic matrix are sorted in an ascending order. Then, the element on the top of the list is removed from the list and it is assigned to one of the links of the Shufflenet. This means the two nodes generating the relevant traffic load are placed as neighbours in the Shufflenet. This operation is repeated until all the nodes are placed in the Shufflenet [13].

Gradient algorithm which is a form of greedy algorithm has a rather radical approach to the problem. It operates on the gradient matrix which is formed by subtracting the transpose of the traffic matrix from the traffic matrix itself. In a (p,k) Shufflenet, each node can reach back to itself after traversing through the network which is called loop. This loop may be k hops long which is represented by k-loop or 2k hops long which represented by 2k-loop. Given two nodes, node i and node j, if node j is in the k-loop of node i and node i can reach node j in x hops, node j can reach node i in (k-x) hops. This is the same for 2k-loop nodes. Thus, the gradient matrix attracts attention because it focuses on traffic load between two nodes in both direction. The gradient matrix is sorted in an descending order and the node placement operations are accomplished starting from nodes with the highest gradient value. If both nodes are not placed and they can be placed in 1-hop distance. Otherwise, the algorithm decides on which loop to choose, k-loop or 2k-loop in order to place the unplaced nodes [15].

Local algorithm utilizes the structure of the Shufflenet in order to place the nodes optimally. In each of the k stages of a (p,k) Shufflenet, there are p^{k-1} disjoint sets

each of which has p nodes. Each node of the set is connected to the same nodes at the next stage. Taking the traffic matrix into account, two ordered sets, set A from the first stage and set B from the second stage, are formed such that traffic flow from set A to set B is extremely high compared to the traffic flow from set B to set A. Then, all the p-node sets in the first stage and the second stage are formed as described above. As soon as the first two stages are completed. The remaining stages are handled two by two and all the node placements are accomplished as in the first two stages [13].

In global algorithm, in order to assign a node to a location in Shufflenet, the traffic from this node to all previously placed nodes and to this node from all previously placed nodes is taken into consideration by using a penalty function. The location which minimizes the penalty function is chosen for the node to be placed. Then, penalty values of all unplaced nodes for all unassigned locations are renewed taking the newly placed node into account. The algorithm runs until all the nodes are placed [13].

As far as we concentrate on node placement optimization in our work and our objective is to minimize the weighted mean hop distance ($\bar{h}$) which corresponds to delay in physical domain, we refer to the delay-based implementation of iterative algorithm and simulated annealing in our work.

3.1 Iterative Algorithm

This algorithm is based on an iterative approach in order to obtain a Hamiltonion chain which minimizes a defined cost [14]. In our work, the cost is defined as the weighted mean hop distance, in other words delay. This algorithm starts with an initial point in the problem space which consists of N! points accomodating the values of our cost function. At each iteration, the algorithm finds a new point in the space. If this new point has a smaller cost function value, we move to this point and keep on repeating this operation until no further minimization is possible. Figure 3.1 illustrates the iterative algorithm [13].

let $G = \{0, 1, \cdots, N-1\}$ be the set of nodes.
NoOfIterations = $\log_2 N$;
{Could be performed for different number of iterations as well,
but this value appeared to perform quite well}
for $q := 1$ **to** NoOfIterations **do**
begin
  pick a random permutation of $[0, \cdots, N-1]$
    {or the output of an earlier algorithm}
    as the initial node sequence;
  let $\alpha = [\alpha_0, \cdots, \alpha_{N-1}]$ denote this initial node sequence,
    where α_i is the node placed at $\left([\, i \mod p^k \,], \left\lfloor i/p^k \right\rfloor \right)$;
  for $g := 0$ **to** N-2 **do**
  begin
    for $u := g+1$ **to** $N-1$ **do**
    begin
      let $\alpha' = [\alpha'_0, \cdots, \alpha'_{N-1}]$
        where $\alpha'_i = \alpha_i$ for $0 \le i \le (g-1)$ and $u+1 \le i \le N-1$,
              $\alpha'_g = \alpha_u$, and
              $\alpha'_i = \alpha_{i-1}$ for $(g+1) \le i \le u$.
      let $\overline{H}(\alpha)$ = average weighted hop-distance in the ShuffleNet represented by α.
      if $(\overline{H}(\alpha') < \overline{H}(\alpha))$ **then**
        $\alpha \leftarrow \alpha'$
    end;
  end;
end;

Figure 3.1 Iterative Algorithm [13]

The iterative algorithm given in Figure 3.1 is an example given for Shufflenet but there is no restriction on the topology thus this algorithm can also be applied to MSN. What is important here is the if statement. Since we intend to minimize the mean delay, the deciding factor, here, is the weighted mean hop distance. The probability of finding a global minimum, q_{global}, by applying iterative algorithm can be given as follows [13]:

$$q_{global}=1-(1-q)^w \qquad (3.1)$$

where

w : The number of independent initial points

q : The probability of finding a global optimum which can be formulated as

$$q = \sum_{y_i \in Y_O} q_i \qquad (3.2)$$

where

Y_O : The set of optimal solutions

q_i : The probability of finding local optimum y_i which can be formulated as

$$q_i=|X_i|/|X| \quad (3.3)$$

where

X_i : The set of initial points which produce local optimum y_i

X :The set of all permutations

Time complexity of the iterative algorithm for an N-node topology is $O(N^4)$ since the time complexity of calculating the weighted mean hop distance is $O(N^2)$ and this is repeated N^2 times. Thus, the time complexity of the iterative algorithm is $O(N^4)$ [13], [14]. This high complexity is the main drawback of this algorithm although the results obtained are very competitive.

3.2 Simulated Annealing

Simulated annealing approach can be implemented in a delay-based approach like the iterative algorithm.Thus, we can adopt simulated annealing in order to minimize our defined objective function or at least obtain a near-optimal solution. The main problem of many heuristics is to get caught in a local optimal point. However, simulated annealing approach overcomes this problem by allowing the algorithm, with a probability p, to jump to the points with worser objective function values [19].

Simulated Annealing method has a number of input parameters as well as the traffic matrix and the hop distance matrix which are mandatory for node placement optimization. These parameters are the initial temperature, cooling rate and stopping conditions [19]. The adoption of these parameters differ in literature. In our work, we refer to [23] to determine these parameters after presenting the algorithm. The simulated annealing algorithm we apply in our work is illustrated in Figure 3.2 [19].

1. Generate a random node assignment.
2. Initialize the temperature T.
3. While ($no\_of\_attempts < MaxAttempt$) do
   While ($no\_of\_moves \leq MaxMove$) do
      Randomly select two neighboring nodes i and j
      Compute Δ_{ij} if $n(i)$ and $n(j)$ were swapped.
      If $\Delta_{ij} > 0$, swap their locations; $no\_of\_attempts = 0$;
        increment no_of_moves
      If $\Delta_{ij} \leq 0$, swap their locations and increment no_of_moves
        with probability $e^{-\Delta_{ij}/T}$; increment $no\_of\_attempts$
   $T = T \times \alpha$

Figure 3.2 Simulated Annealing Algorithm [19]

Δ_{ij} is the reduction in the cost function if neighbor nodes i and j are swapped. If $\Delta_{ij}>0$, this means there is an improvement on the cost function thus swapping of neighbor nodes is performed. Otherwise, swapping is performed with probability $e^{\Delta_{ij}/T}$. In addition, α, the cooling rate, is set to 0.95.

Referring to [23], we set *MaxAttempt* to 10xN and *MaxMove* to N. *T* is computed as follows,

$$T = -\overline{\Delta_+} / \ln \chi \qquad (3.4)$$

where

$\overline{\Delta_+}$: The average change of Δ_{ij} for $\Delta_{ij}>0$

χ : The desired probability that a swapping will be accepted for initial solution, 0.6

It is stated in [24] that simulated annealing algorithm is sensitive to the annealing schedule and the parameters given above. Time complexity of simulated annealing algorithm is an open issue except for maximum matching problem [25].

4 OUR WORK

In this section, we go into the details of our work. We introduce the traffic patterns that we use to run our simulations. Following the introduction of traffic patterns, our simulation principles are given. Then, we provide simulation results together with their assessments. 8-node traffic matrices are presented in Appendix A.

4.1 Traffic Patterns

There is a great number of traffic models proposed in literature. However, in our simulations, we use four types of traffic some of which are ideal to represent the real world and some are there to assess the performance of the heuristics we apply in our simulations.

4.1.1 Uniform Random Traffic

This type of traffic, which will be denoted by "t1" in the proceeding subsections, does not actually represent the real world. This is because traffic load between all the nodes, except for the nodes themselves, is uniformly distributed random number between 0.0-1.0. This model was proposed in [15] in order to assess the impact of non-uniform traffic on the network performance.

4.1.2 Uniform Random Traffic with a Higher Amplitude

In our simulations, we use this traffic model, denoted by "t2" in the proceeding subsections, in order to assess the impact of amplitude on uniform random traffic. This model was proposed in [1] and it is similar to t1 in terms of uniform randomness except we set the amplitude to 90.0 not 1.0 for this traffic model.

4.1.3 Non-Uniform Random Traffic

For this type of traffic, some nodes are adopted as servers each of which is serving a number of clients. The traffic load between servers and their clients is a uniformly distributed random number between 0-γ where γ is the skew factor while the traffic load between non-server nodes is a uniformly distributed random number between 0-1 [15]. In our work, γ is set to 100 and this traffic model is denoted by "t3"

4.1.4 Varying Range Traffic

This traffic model was previously proposed in [1] and it accomodates traffic patterns with a varying range. The mean traffic load is kept constant while the range around the mean changes. In our work, for this traffic model, we set the mean to 100 and the range is chosen from a set S={20, 50, 90}. Therefore, the traffic pattern with a range 20 oscillates between (100-20) and (100+20). We denote this traffic model "t4(x)" where x is a member of S.

4.2 Simulation Principles

We run the simulations 10 times each starting with a different initial assignment for the given regular virtual topology and the traffic model. Initial assignments are read from a read-only file. Therefore, initial assignments for both regular virtual topologies and heuristics are the same. As a result, we obtain the best result among 10 trials and we also possess the average result of the heuristic we apply for the given traffic model and the regular virtual topology.

As we are going to go into details in Section 4.3.1, we evaluate the performance of our heuristics with respect to random assignment and come up with a term "PI" which stands for "Performance Improvement".The mathematical formulation of PI is given in Equation 4.1. This approach was adopted in [19]. Therefore, we randomly make 10 assignments for each topology under the given traffic model and the arithmetic mean of 10 random assignments are taken into consideration in order to calculate PIs for the given traffic matrix and the regular virtual topology.

$$PI = \frac{E(\bar{h}_{ra}) - \bar{h}_{oa}}{E(\bar{h}_{ra})} x100 \qquad (4.1)$$

where

$E(\bar{h}_{ra})$: Average of 10 random assignments

$\bar{h}_{oa}$: Best result obtained by the heuristic

PIs are also calculated for the average result obtained by optimal assignment. In this case, $\bar{h}_{oa}$ is replaced by $E(\bar{h}_{oa})$.

In Section 4.3.2, we evaluate the performance of the regular virtual topologies we use. The average and best results obtained by both heuristics for both regular virtual topologies are given.

In Section 4.3.3, we assess the performance of our heuristics for the given traffic model and 8-node regular virtual topology with respect to the best result which can be obtained for the given traffic model and regular virtual topology. For 8-node MSN and Shufflenet, there are 8! possible node assignments. We simply calculate the cost for each node assignment thus we reach the global minimum of the problem space. However, this is not possible for larger topologies since it is not feasible to calculate cost for all node assignments for the given problem space. Here, we define "PD" which stands for "Performance Degradation". The mathematical formulation of PD is given below in Equation 4.2.

$$PD = \frac{\bar{h}_{oa} - \bar{h}_{best}}{\bar{h}_{best}} x100 \qquad (4.2)$$

where

$\bar{h}_{oa}$: Best result obtained by the heuristic

$\bar{h}_{best}$: Global minimum of the problem space

In Section 4.3.4, we assess the impact of non-uniform random traffic on the performance of the iterative algorithm and the simulated annealing algorithm for both topologies by comparing the results obtained under uniform and non-uniform random traffic.

In Section 4.3.5, the impact of change of range is evaluated for varying range traffic. We assess the performance of both heuristics we for the two regular virtual topologies under varying range traffic with R values 20, 50 and 90.

Finally in Section 4.3.6, we calculate the average PIs for 10 uniform random traffic patterns as well as 10 non-uniform random traffic patterns. We compare these PIs with that of obtained for single uniform and non-uniform random traffic patterns. By doing this, we evaluate the impact of number of traffic matrices on the performance.

4.3 Experimental Results

In this subsection, we present the experimental results obtained throughout this work and make assessments on these results.

4.3.1 PI Results of The Heuristics

PI results obtained under traffic pattern t1

Considering the best results achieved, Figure 4.1 and 4.2 indicates that the iterative algorithm seems to give best PI for 8-node while the simulated annealing algorithm outperforms the iterative algorithm in terms of PI for larger topologies.

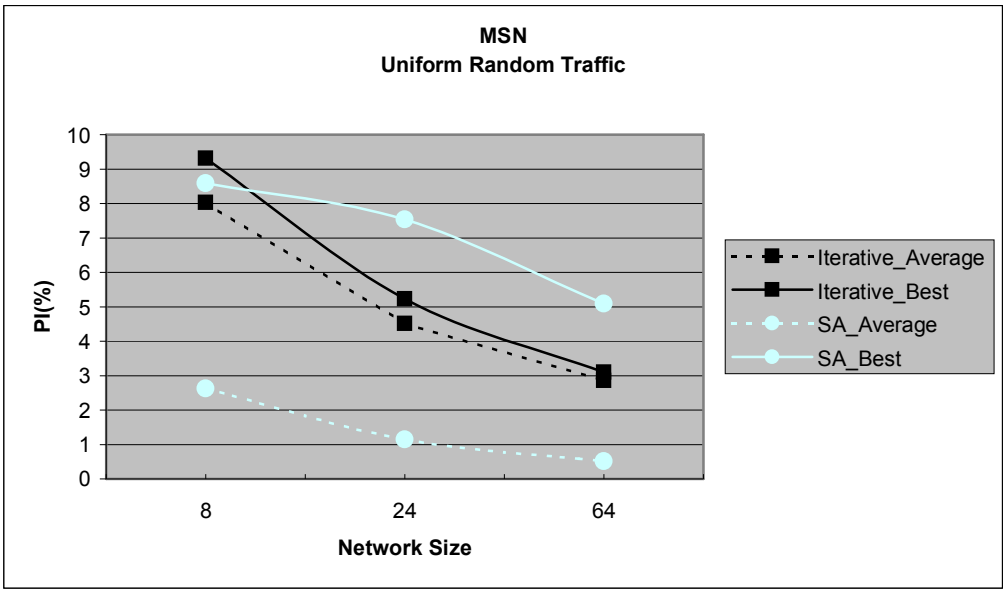

Figure 4.1 PI results obtained for MSN under "t1"

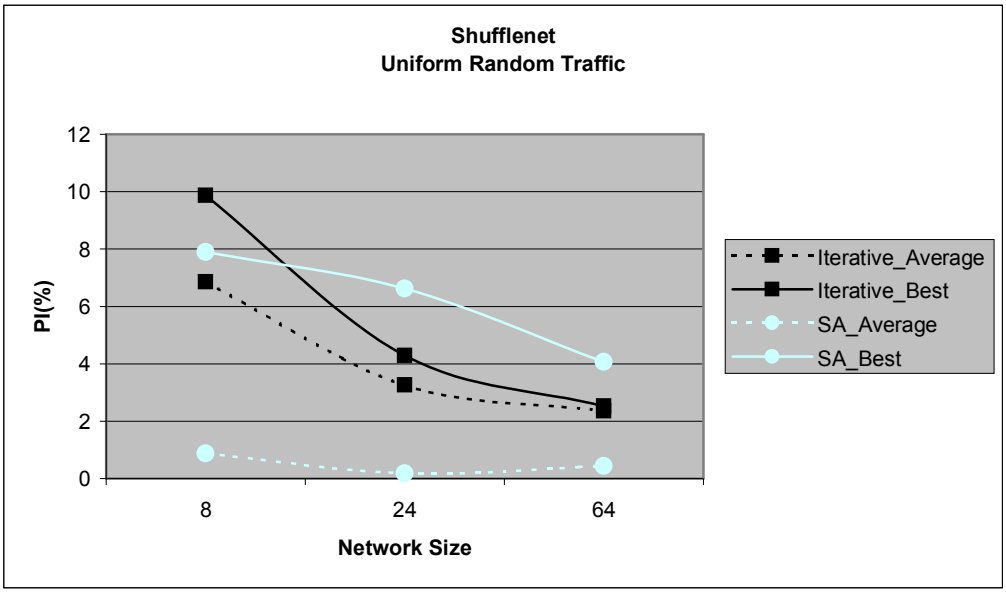

Figure 4.2 PI results obtained for Shufflenet under "t1"

PI results obtained under traffic pattern t2

Figure 4.3 and Figure 4.4 are similar to Figure 4.1 and Figure 4.2 respectively. However, there is a slight improvement in terms of PI. This implies the network performance is rather immune to the change in amplitude for uniform random traffic. The iterative algorithm outperforms the simulated annealing algorithm for 8-node while the simulated annealing algorithm outpaces the iterative algorithm for larger topologies considering the best results achieved.

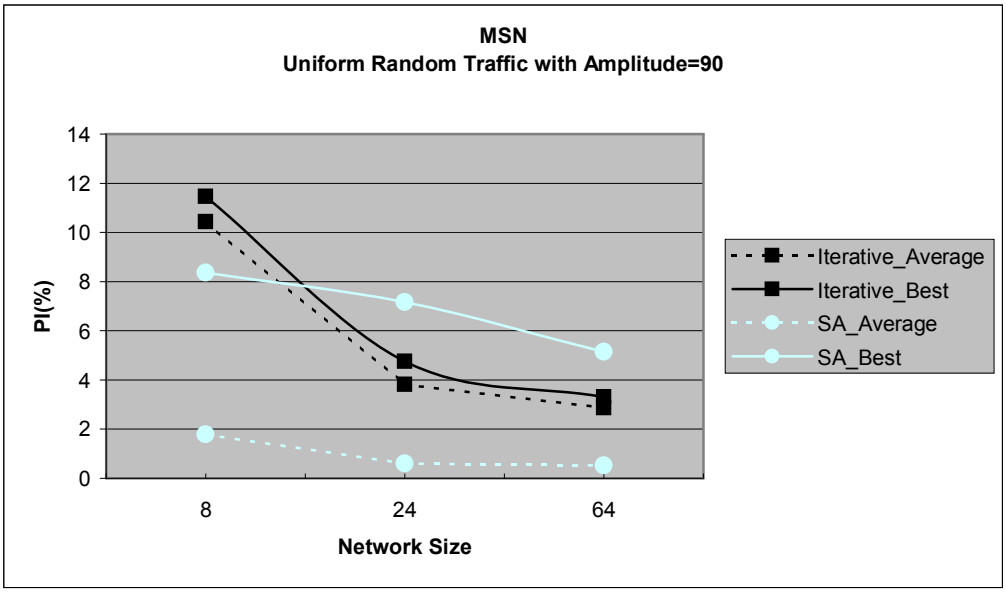

Figure 4.3 PI results obtained for MSN under traffic "t2"

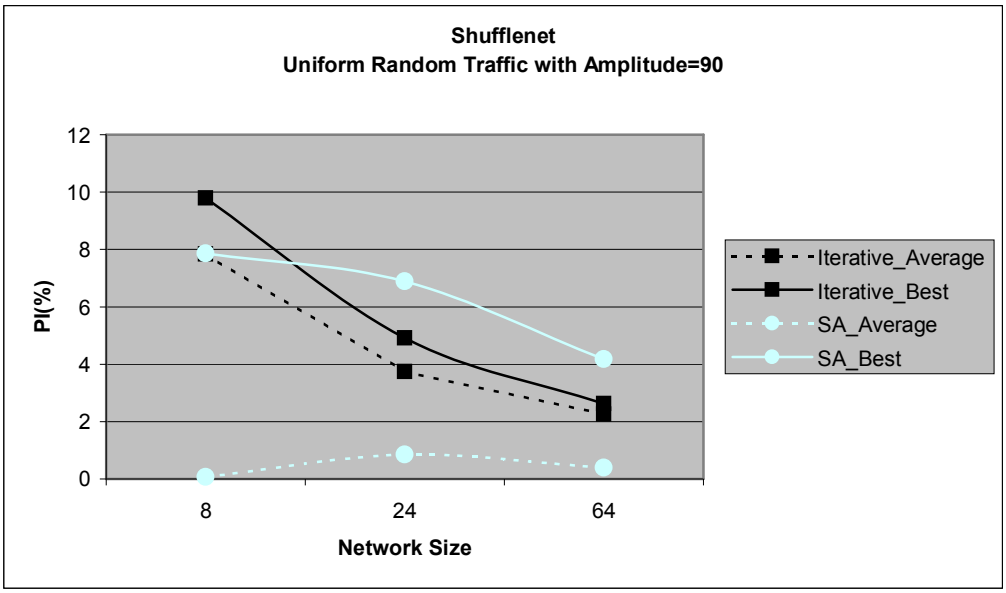

Figure 4.4 PI results obtained for Shufflenet under traffic "t2"

PI results obtained under traffic pattern t3

Considering the best results achieved, Figure 4.5 and Figure 4.6 imply that the iterative algorithm and simulated annealing algorithm obtain similar results for 8-node topology. On the other hand, SA outperforms iterative algorithm in terms of PI for larger topologies. We should also note that performance of simulated annealing on Shufflenet is better for 24-node topologies than that of 8-node Shufflenet. This may be because of the performance of random assignment for 24-node Shufflenet.

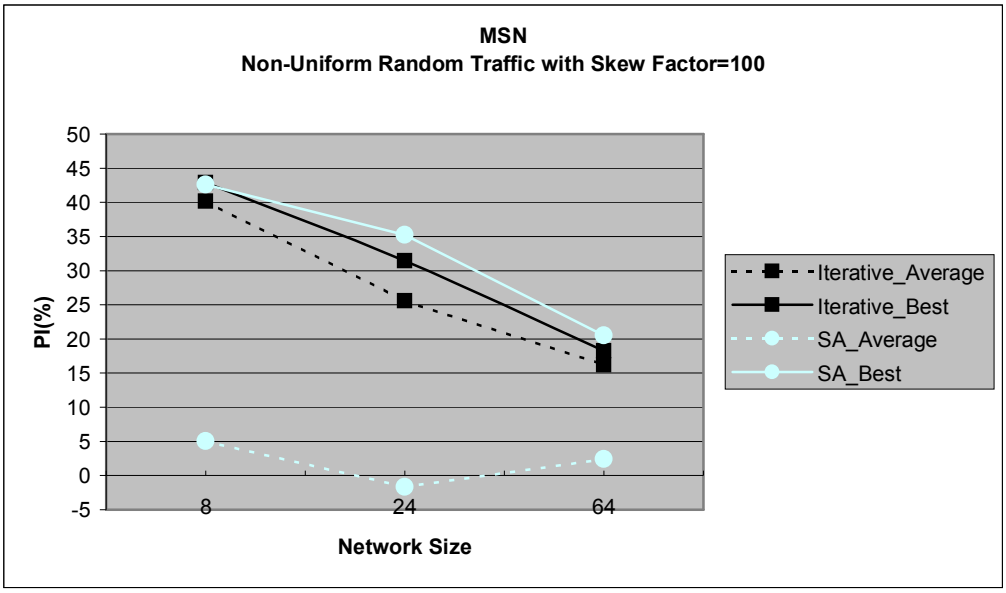

Figure 4.5 PI results obtained for MSN under "t3"

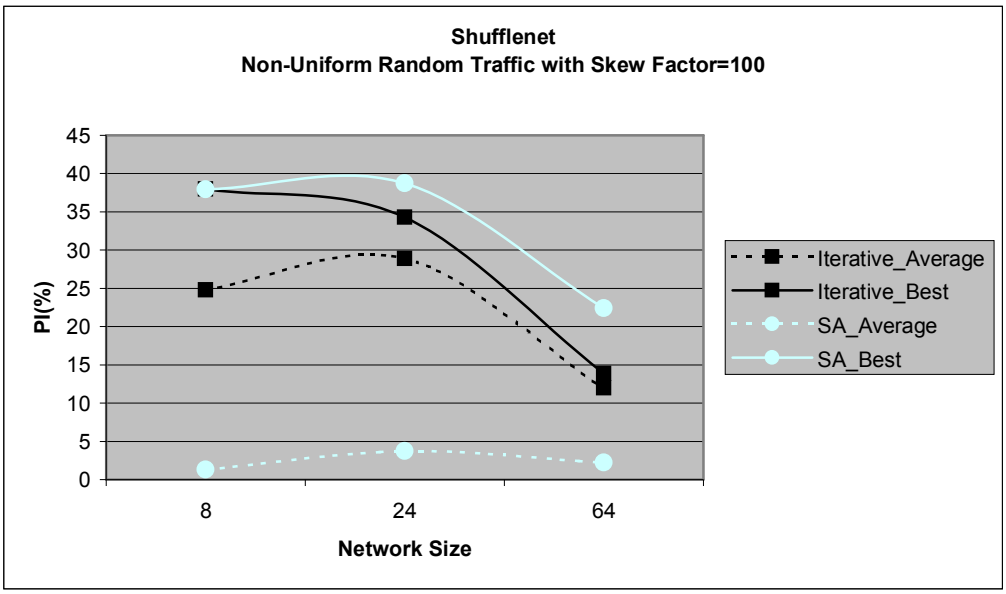

Figure 4.6 PI results obtained for Shufflenet under "t3"

PI results obtained under traffic pattern t4(20)

Taking the best results achieved into consideration, we can see from Figure 4.7 and Figure 4.8 that iterative algorithm outperforms the simulated annealing algorithm for 8-node topologies while the simulated annealing algorithm is better than iterative algorithm for larger topologies.

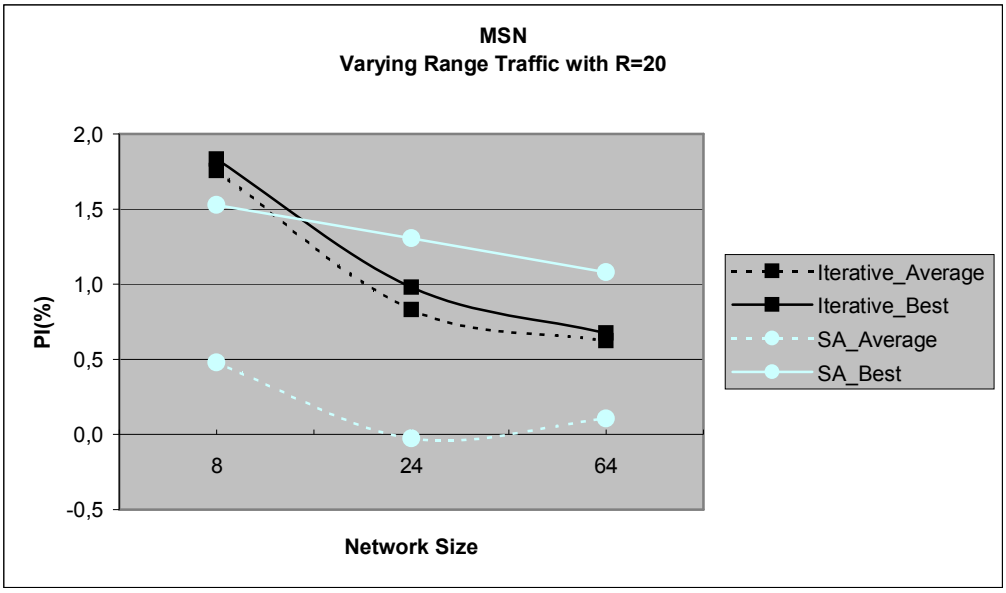

Figure 4.7 PI results obtained for MSN under "t4(20)"

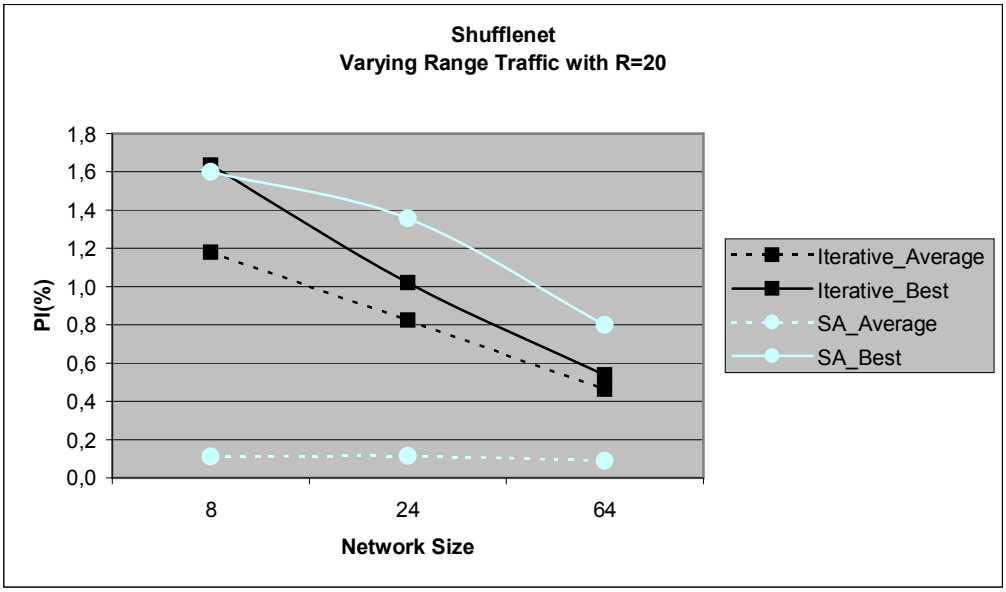

Figure 4.8 PI results obtained for Shufflenet under "t4(20)"

PI results obtained for t4(50)

Considering the best results achieved, Figure 4.9 and Figure 4.10 imply that the iterative algorithm outperforms the simulated annealing algorithm for 8-node topologies while the simulated annealing algorithm has an advantage over iterative algorithm for larger topologies.

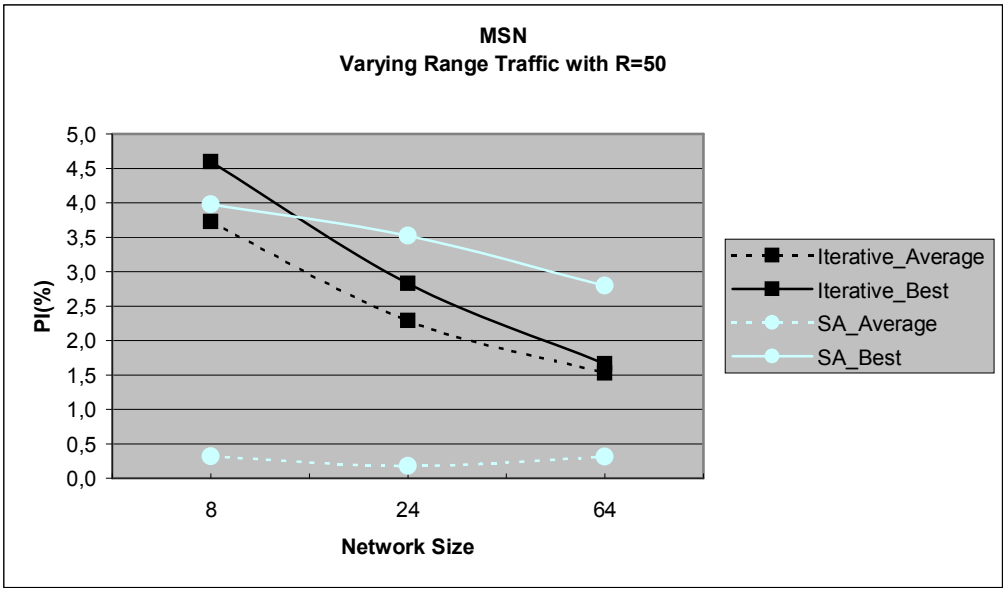

Figure 4.9 PI results obtained for MSN under "t4(50)"

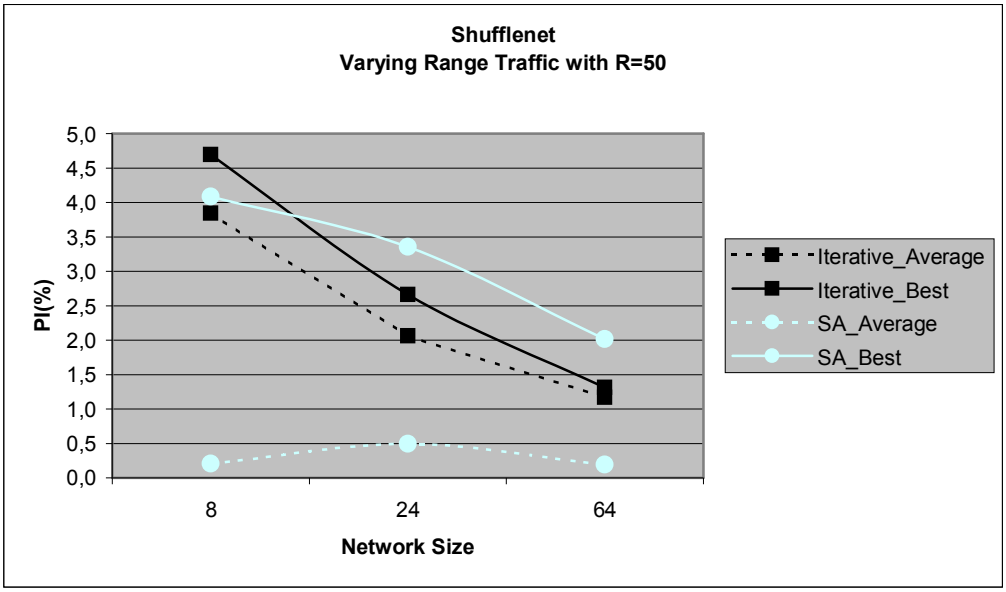

Figure 4.10 PI results obtained for Shufflenet under "t4(50)"

PI results obtained for t4(90)

Taking the best result achieved into consideration, we can see from Figure 4.11 and Figure 4.12 that the iterative and the simulated annealing algorithms achieve closer results for 8-node topologies. However, the simulated annealing algorithm outperforms the iterative algorithm for larger topologies.

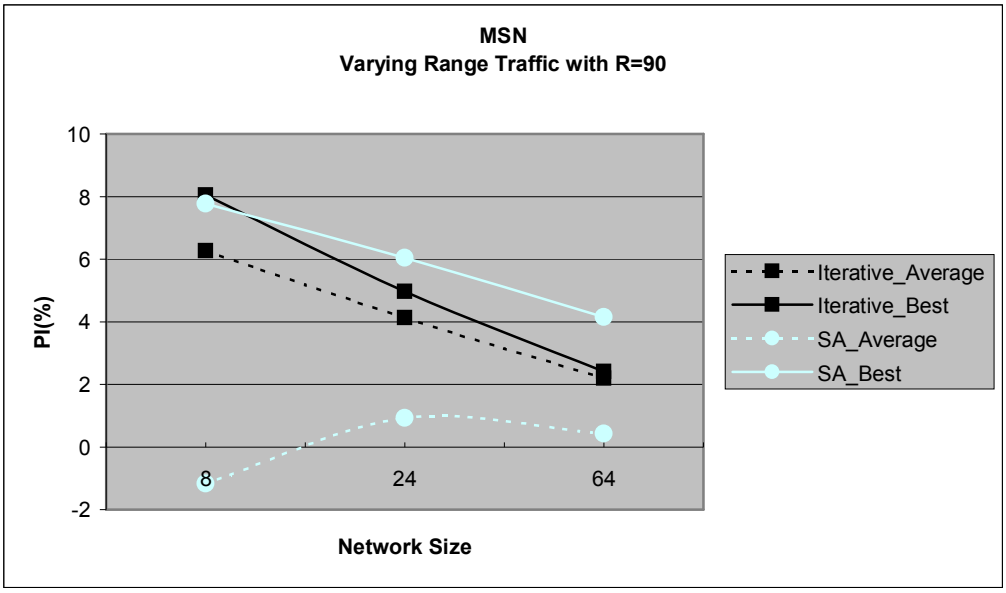

Figure 4.11 PI results obtained for MSN under "t4(90)"

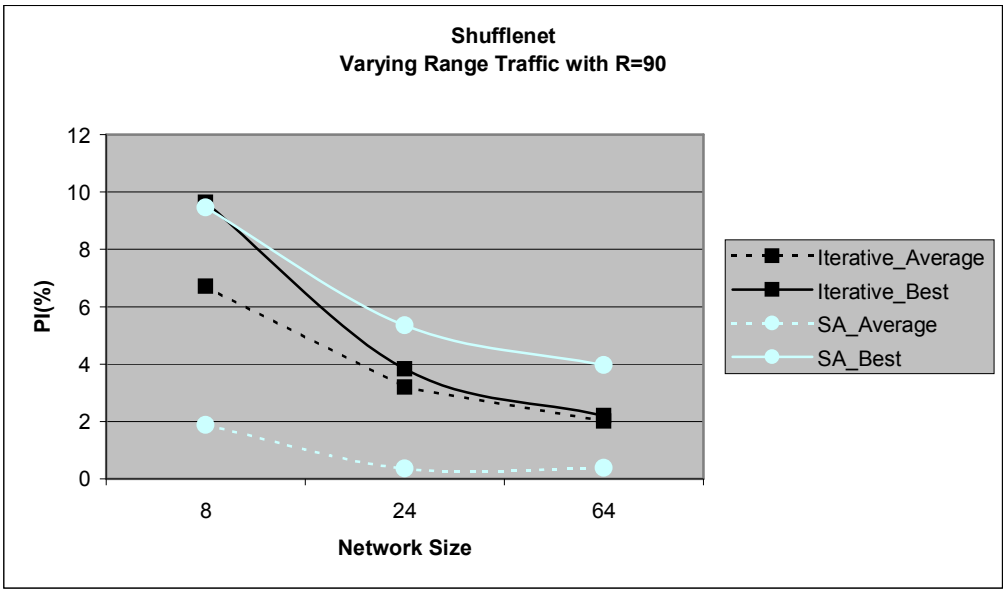

Figure 4.12 PI results obtained for Shufflenet under "t4(90)"

4.3.2 Performance Evaluation of the Manhattan Street Network and the Shufflenet

In this subsection, we analyze the performance of the Manhattan Street Network and the Shufflenet. The results are not normalized since the random assignment values are different for both of the regular virtual topologies.

Performance evaluation of the regular virtual topologies under traffic "t1"

From Table 4.1, we can say the Shufflenet is better than the Manhattan Street Network for 24 and 64-node topologies regardless of the two heuristics we apply. Moreover, we can reach a better result by applying the iterative algorithm on the Shufflenet for 8-node topologies

Table 4.1 Performance evaluation of the regular virtual topologies under "t1"

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.8628	1.8368	3.1737	3.1737	4,8860	4,8738
	SA	1.9721	1.8514	3.2855	3.2855	5,0038	4,7736
Shufflenet	Iterative	1.8765	1.8156	3.1408	3.1408	4,5299	4,5219
	SA	1.9969	1.8555	3.2410	3.2410	4,6189	4,4508

Performance evaluation of the regular virtual topologies under traffic "t2"

Table 4.2 is similar to Table 4.1 with an exception. The MSN is better for 8-node by applying iterative algorithm

Table 4.2 Performance evaluation of the regular virtual topologies under "t2"

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.8225	1.8016	3.1958	3.1646	4,8733	4,8511
	SA	1.9771	1.8445	3.3029	3.0848	4,9911	4,7595
Shufflenet	Iterative	1.8583	1.8188	3.1484	3.1099	4,5320	4,5151
	SA	1.9727	1.8189	3.2428	3.0453	4,6184	4,4429

Performance evaluation of the regular virtual topologies under traffic "t3"

It is obvious from Table 4.3 that under non-uniform random traffic, the MSN is better than the Shufflenet for 8 and 24-topologies. However, the Shufflenet is better for 64-node topologies.

Table 4.3 Performance evaluation of the regular virtual topologies under "t3"

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.3032	1.2440	2.2445	2.0684	3,8371	3,7423
	SA	2.0674	1.2496	3.0667	1.9532	4,4670	3,6391
Shufflenet	Iterative	1.5152	1.2497	2.2846	2.1112	3,9159	3,8328
	SA	1.9878	1.2505	3.0932	1.9690	4,3496	3,4536

Performance evaluation of the regular virtual topologies under traffic "t4(20)"

From Table 4.4, we can say that applying the iterative algorithm on the MSN is advantageous for 8-node topologies under traffic t4(20). However, the Shufflenet takes over the advantage for larger topologies regardless of the heuristic applied.

Table 4.4 Performance evaluation of the regular virtual topologies under "t4(20)"

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.9695	1.9680	3.2887	3.2837	4,9886	4,9861
	SA	1.9951	1.9741	3.3171	3.2729	5,0146	4,9657
Shufflenet	Iterative	1.9779	1.9688	3.2409	3.2345	4,6147	4,6112
	SA	1.9993	1.9695	3.2641	3.2235	4,632	4,5991

Performance evaluation of the regular virtual topologies under traffic "t4(50)"

It can be seen from Table 4.5 that the Shufflenet outpaces the MSN for all network sizes that are of interest regardless of the heuristic applied.

Table 4.5 Performance evaluation of the regular virtual topologies under “t4(50)”

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.9359	1.9184	3.2394	3.2214	4,9468	4,9402
	SA	2.0044	1.9308	3.3093	3.1985	5,0079	4,8831
Shufflenet	Iterative	1.9340	1.9169	3.2051	3.1856	4,5806	4,5739
	SA	2.0072	1.9292	3.2564	3.1628	4,6258	4,5412

Performance evaluation of the regular virtual topologies under traffic “t4(90)”

From Table 4.6, we can see that the Shufflenet is better for 24 and 64-node topologies while applying iterative algorithm on the MSN is advantageous for 8-node topologies.

Table 4.6 Performance evaluation of the regular virtual topologies under “t4(90)”

		8 Node		24 Node		64 Node	
		Average	Best	Average	Best	Average	Best
MSN	Iterative	1.8714	1.8359	3.1868	3.1589	4,8881	4,8772
	SA	2.0200	1.8414	3.2933	3.1234	4,9762	4,7905
Shufflenet	Iterative	1.8972	1.8378	3.1614	3.1408	4,5412	4,5327
	SA	1.9955	1.8414	3.2541	3.0913	4,6168	4,4507

4.3.3 PD Results

In this subsection, we evaluate the performance degradation of the heuristics- both the best and average results achieved- with respect to the global minimum for the given traffic pattern and the regular virtual topology. We should note that the global minimum values obtained for both topologies under each traffic pattern are the same. Figure 4.13 to Figure 4.18 summarize the performance degradation of the heuristics for both topologies under the traffic patterns that we use for our simulations.

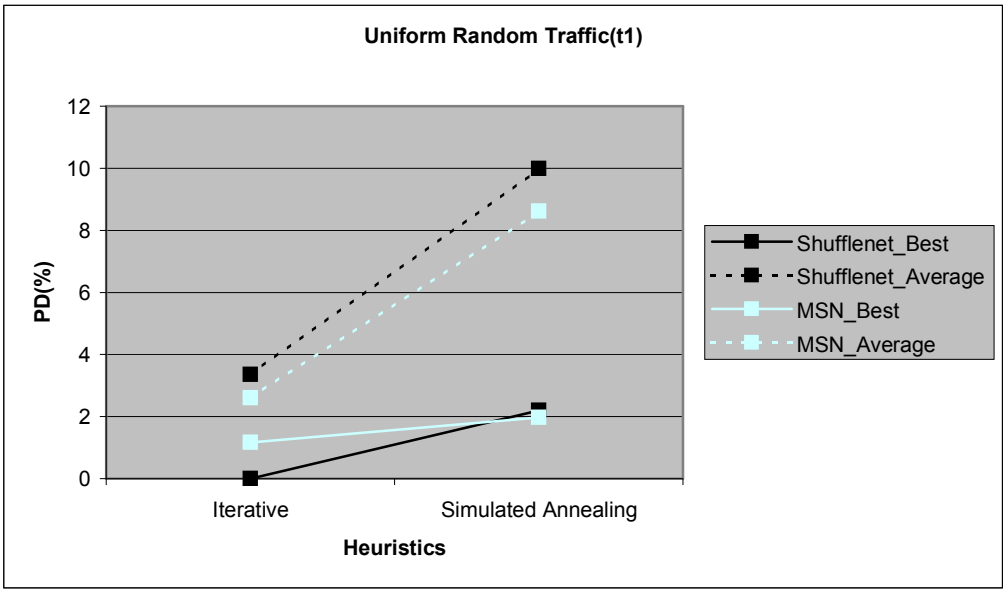

Figure 4.13 PD results obtained under "t1"

From Figure 4.13, we see that by applying the iterative algorithm on the Shufflenet we reach the global minimum of the space. Moreover, the gap between the average and the best results obtained by the simulated annealing algorithm is significant

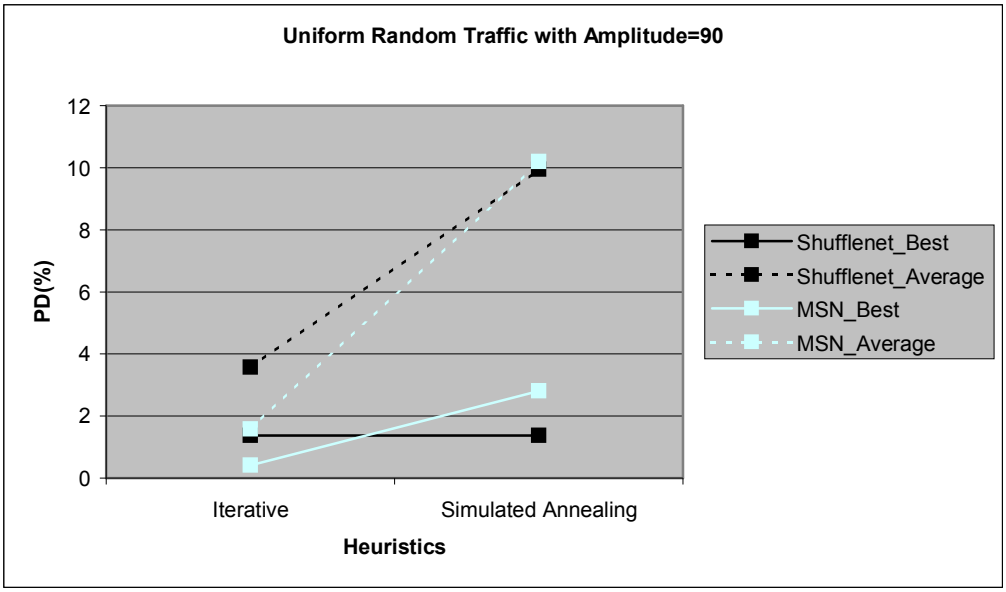

Figure 4.14 PD results obtained under "t2"

From Figure 4.14, we observe that both heuristics obtain similar results for the Shufflenet while the iterative algorithm is more convenient for the MSN. Again, the gap between the average and the best results obtained by the simulated annealing algorithm is significant.

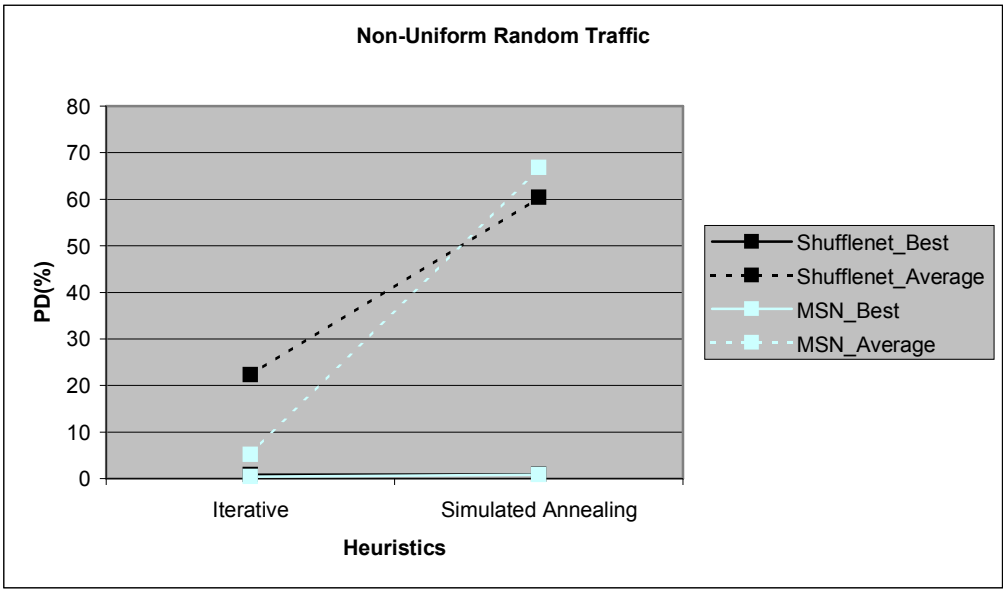

Figure 4.15 PD results obtained under "t3"

We can see from Figure 4.15 that both of the heuristics on both of the regular virtual topologies reach the global minimum.

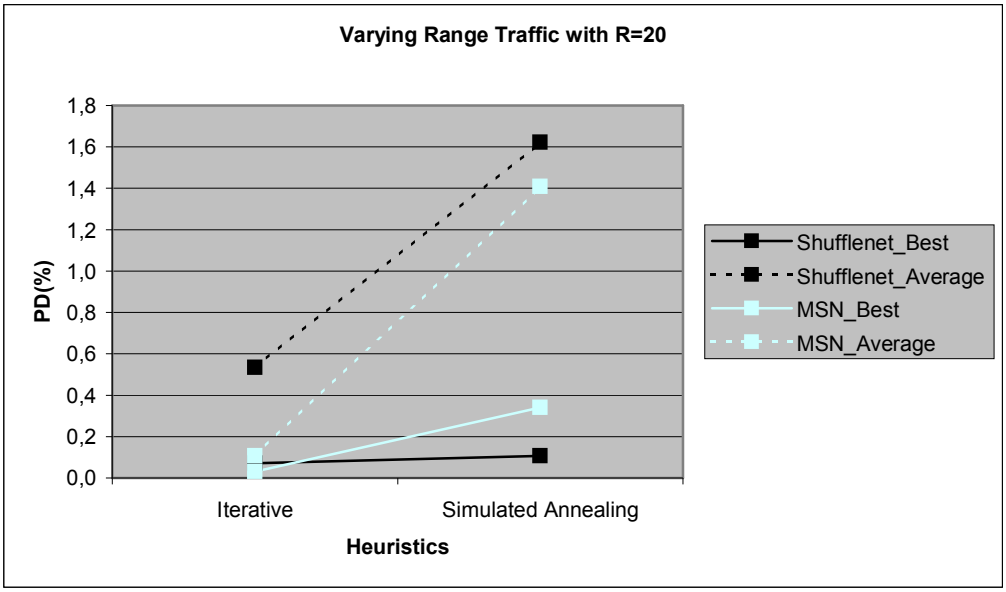

Figure 4.16 PD results obtained under "t4(20)"

Figure 4.16 shows that the performance of both the heuristics and the regular virtual topologies are almost perfect since they are quite close to the global minimum of the problem space.

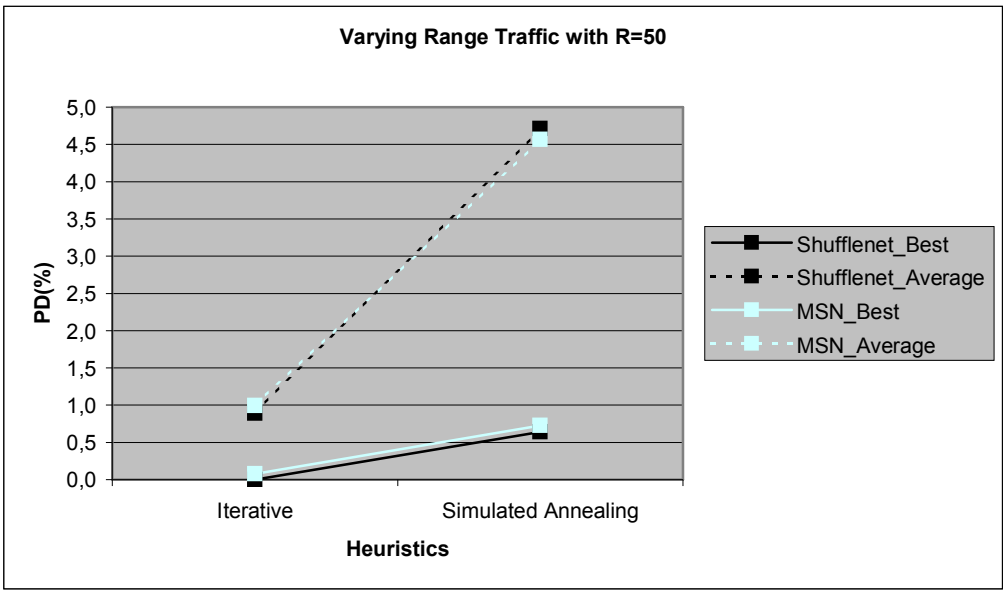

Figure 4.17 PD results obtained under "t4(50)"

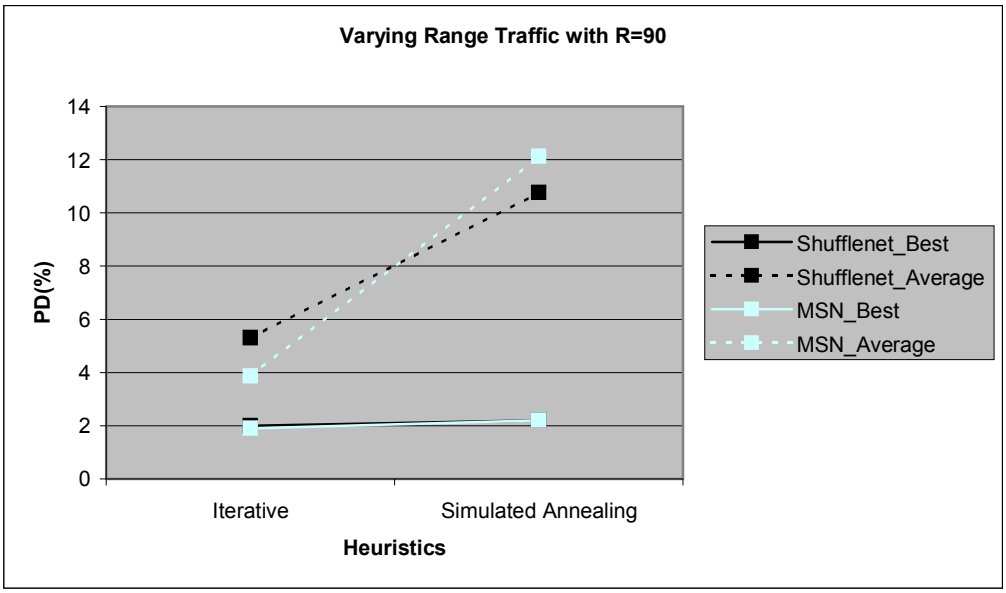

Figure 4.18 PD results obtained under "t4(90)"

Considering the best results achieved by the heuristics, we can see from Figure 4.17 and Figure 4.18 that both of the regular virtual topologies perform similarly when any of the heuristics are applied.

4.3.4 Impact of Non-Uniform Traffic on the Performance of the Heuristics

In this subsection, we evaluate the impact of non-uniform random traffic on the performance of the heuristics. The PI results given in Figure 4.19 and Figure 4.20 are the best results achieved by the heuristics. It is obvious that for both of the regular virtual topologies, PI values obtained by any of the heuristics for non-uniform

random traffic are are tremendously higher than that of obtained for uniform random traffic.

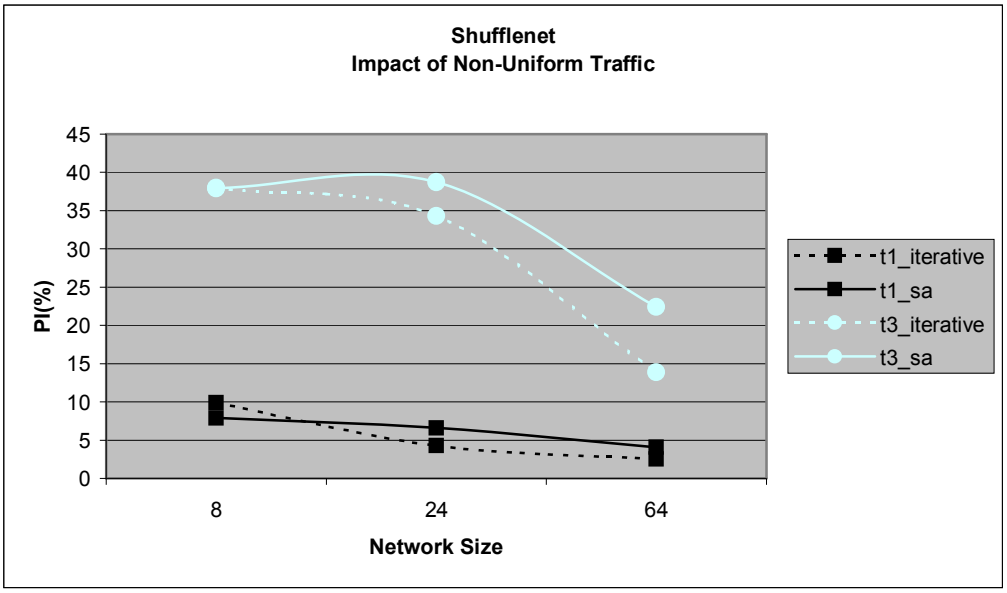

Figure 4.19 Impact of non-uniform traffic on the Shufflenet

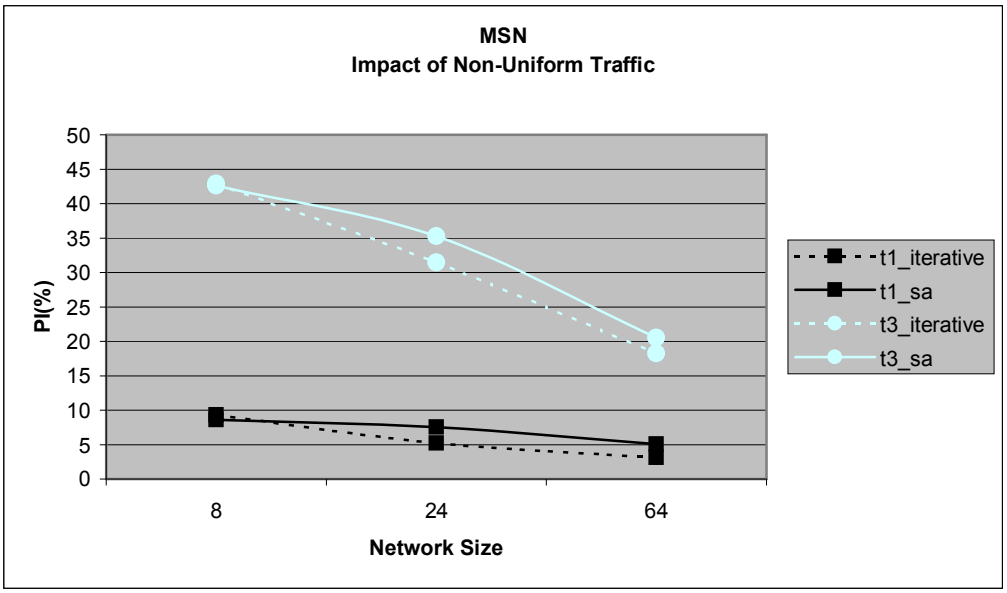

Figure 4.20 Impact of non-uniform traffic on the MSN

4.3.5 Impact of Change of Range on the Performance of the Heuristics

In order to assess the impact of change of range on the network performance, we use "t4" traffic pattern and we expect the network performance to improve as the range increases since the traffic load becomes more non-uniform. Therefore, we compare the best results achieved by the heuristics for both topologies under t4(20), t4(50) and t4(90). Figure 4.21 and Figure 4.22 state that we obtain higher PI values as range increases.

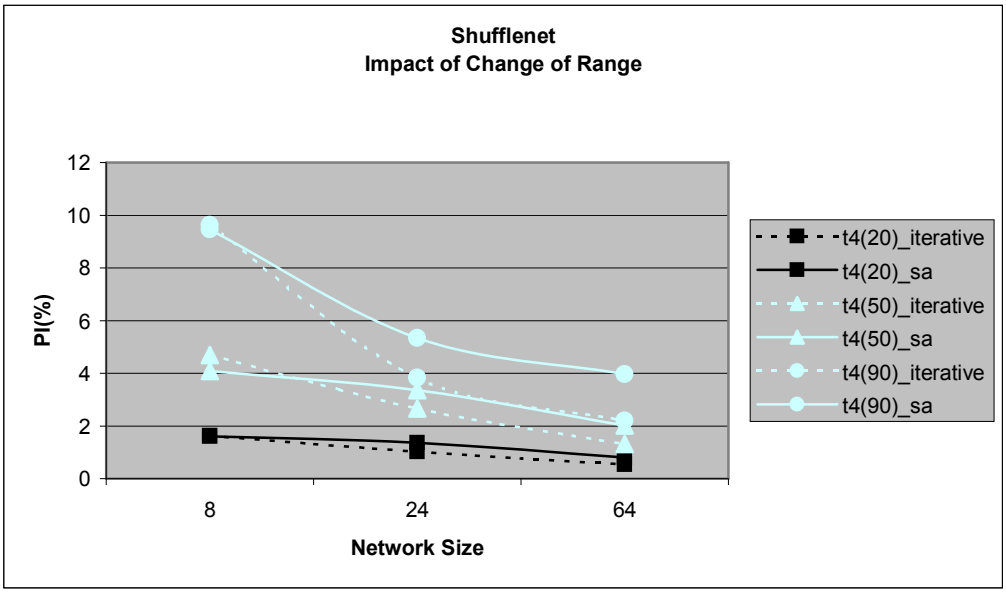

Figure 4.21 Impact of change of range on the Shufflenet

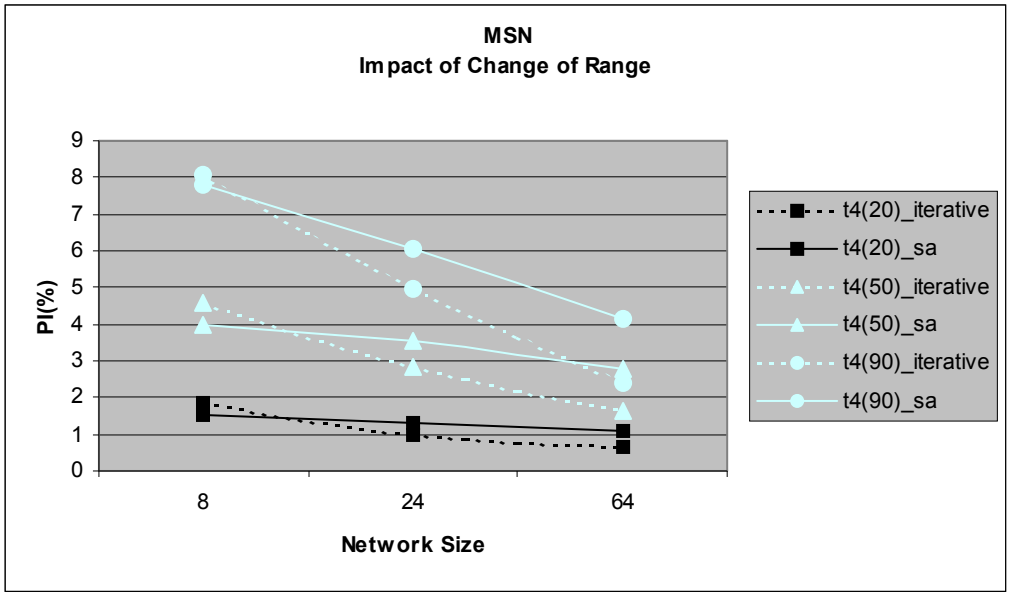

Figure 4.22 Impact of change of range on the MSN

4.3.6 Impact of the Number of Traffic Matrices

In this subsection, we examine the impact of the number of traffic matrices we use to obtain simulation results. Throughout this work, the simulation results are obtained by using single traffic matrix for each traffic model. Here, we obtain the average of the best results obtained under 10 traffic matrices for uniform and non-uniform random traffic models and we compare these results with that of obtained under single traffic matrix. The "_av" extension from Figure 4.23 to Figure 4.26 represent the averages explained above.

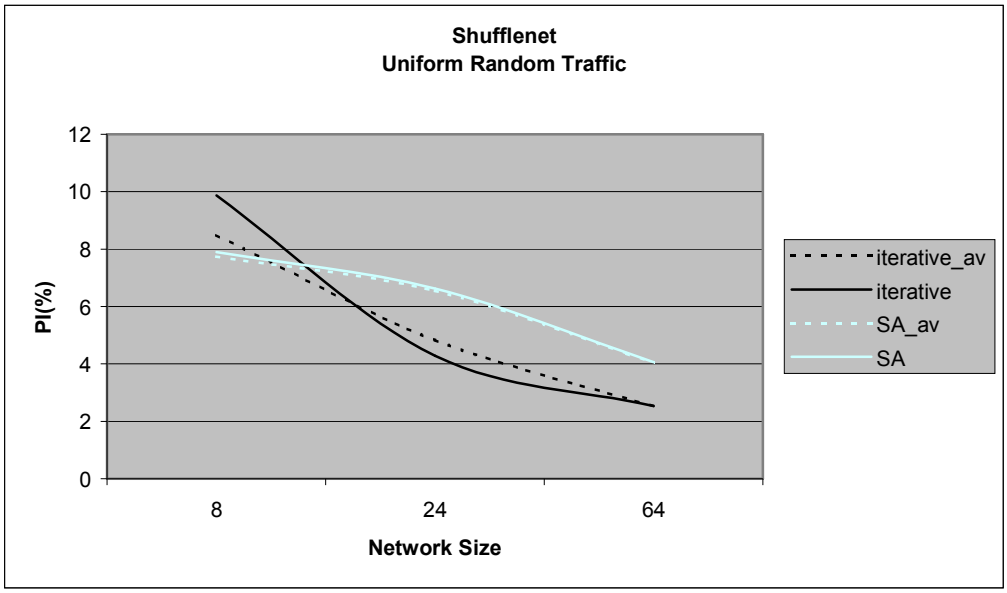

Figure 4.23 Impact of the number of traffic matrices on the Shufflenet under "t1"

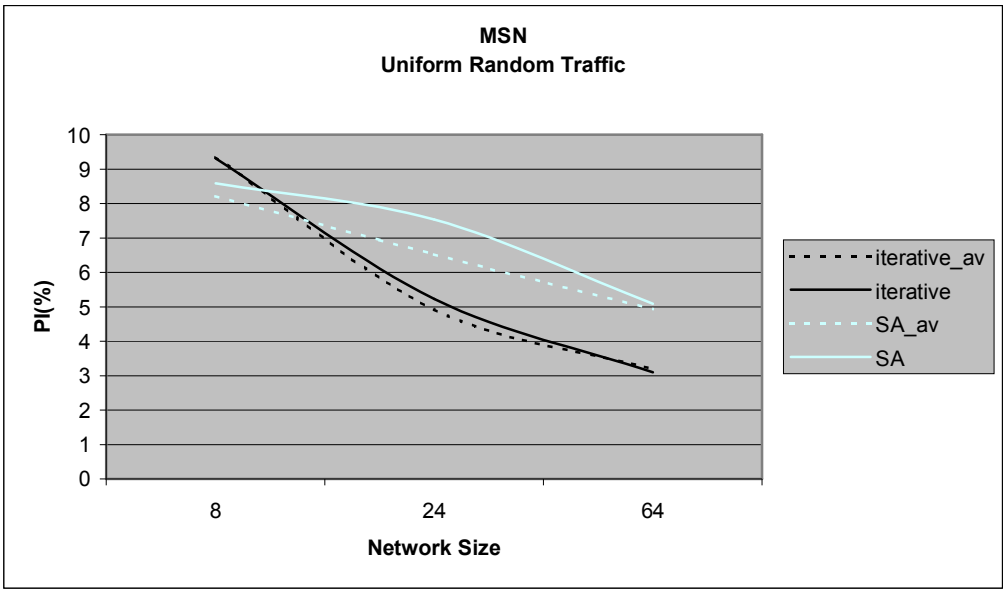

Figure 4.24 Impact of the number of traffic matrices on the MSN under "t1"

We see from Figure 4.23 and Figure 4.24 that the number of traffic matrices used to obtain simulation results does not have an significan effect on the PI results obtained by the heuristics we apply under uniform random traffic. In most the cases above, the results obtained from 10 traffic matrices tend to be slightly worse than that of obtained from single traffic matrix.

We can reach the same idea when we take a look at Figure 4.25 below. However, in Figure 4.26, it is obvious that obtaining the average of best results on the MSN by using the simulated annealing algorithm degrades the performance.

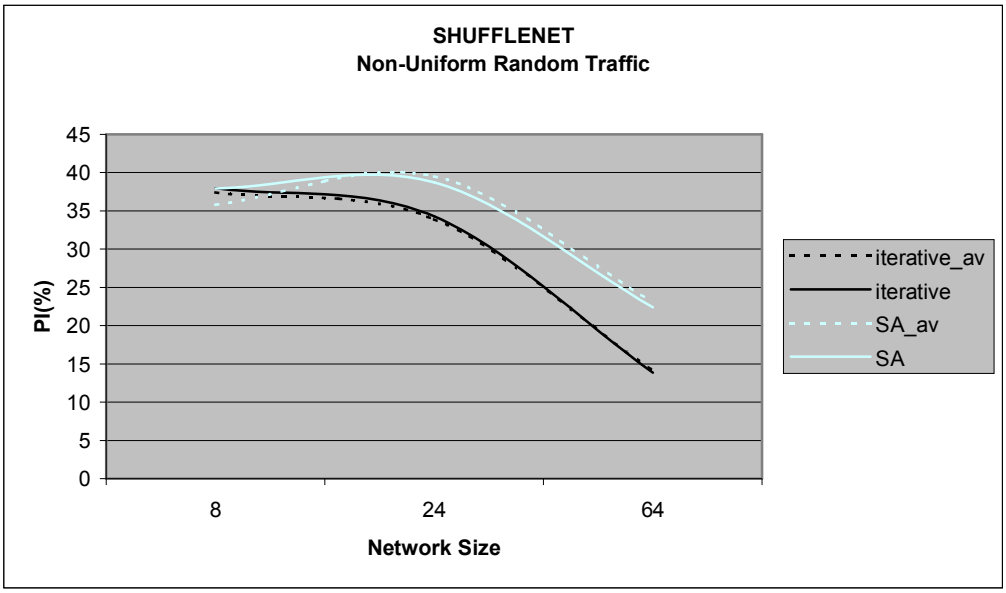

Figure 4.25 Impact of the number of traffic matrices on the Shufflenet under "t3"

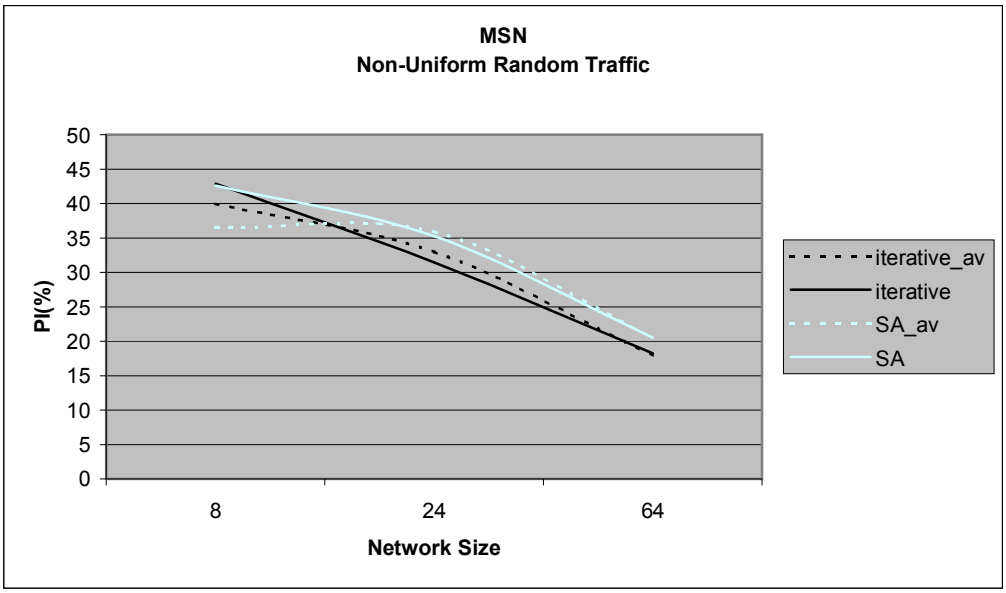

Figure 4.26 Impact of the number of traffic matrices on the MSN under "t3"

5 CONCLUSION

Various multiprocessor interconnection network architectures, including the MSN and the Shufflenet, have been adopted as optical WDM network backbones in literature. These architectures are deployed as regular virtual topologies in arbitrary physical networks.

The deployment of regular virtual topologies in arbitrary physical networks optimally is known as virtual topology design problem. The inputs to the virtual topology design problem are the traffic matrix, the physical topology and the regular virtual topology. This problem can be taken up directly or divided into subproblems, node placement optimization and dilation minimization. The former takes the regular virtual topology and the traffic matrix as inputs while the latter takes the regular virtual topology and physical topology as inputs. Unfortunately, virtual topology design problem is known to be *NP*-Hard and we should refer to heuristics to obtain (near)-optimal solutions.

In this work, we adopt node placement optimization as our solution method, iterative algorithm and simulated annealing as heuristics and the MSN and the Shufflenet as regular virtual topologies. We evaluate the performances of the MSN and the Shufflenet as well as the performances of the iterative algorithm and the simulated annealing under different traffic patterns considering the network size. For 8-node MSN and Shufflenet, we propose a best-case analysis, global minimum for the given topology and the traffic pattern, to assess the performance degradation of our heuristics. Moreover, we assess the impact of traffic on the network performance.

Our simulation results show that amplitude doesn't have a significant impact on the network performance for uniform random traffic patterns. However, as the traffic load between nodes becomes non-uniform, the performance of our heuristics increase tremendously when compared to uniform random traffic. This result is encouraging because it makes sense to model physical world by using non-uniform random traffic. For varying range traffic, the performance of the network tends to increase as the range increases. This is because the overall traffic load of the network becomes

more non-uniform. However, the change in network performance is not as tremendous as it is for uniform and non-uniform traffic.

When we evaluate the performance of our heuristics considering the network size, they degrade as the network size grows.What is significant here is that simulated annealing outperforms iterative algorithm for 24 and 64-node MSN and Shufflenet while iterative algorithm obtains better results for 8-node MSN and Shufflenet. Iterative algorithm tends to get stuck in a local optimum as the problem space grows.

We should also note that for both regular virtual topologies and all traffic patterns the gap between the average and the best result obtained by simulated annealing is huge while this gap is reasonable for iterative algorithm. This leads to the idea that the simulated annealing algorithm is too sensitive to the initial assignment.

When we evaluate the performance degradation of the heuristics for 8-node MSN and Shufflenet, we see that they obtain very close results to the global minimum for the given regular virtual topology and the traffic pattern. It is possible to state that iterative algorithm is slightly better. In addition, we should mention that the global minimum values obtainable for 8-node MSN and Shufflenet for the same traffic pattern are the same.

When we compare the performance of the regular virtual topologies, it is worth to note that the Shufflenet obtains better results for 64-node topologies and outperforms the MSN under all traffic models. For the remaining network sizes, the Shufflenet seems to be slightly better considering the overall results.

As a result, we can say that it is beneficial to use the iterative algorithm for 8-node topologies and the simulated annealing algorithms for 24 and 64-node topologies and non-uniform random traffic has a positive effect on the network performance.

For future work, it would be interesting to adapt a mechanism to iterative algorithm to overcome local optimality and test the performance of iterative algorithm on larger topologies. In addition, it is worth to assess the effect of number of initial assignments on the performance of heuristics since the initial assignments are random for the heuristics we use.

REFERENCES

[1] **O. Komolafe, D. Harle and D. Cotter,**"Impact of Non-Uniform Traffic on the Design of Multi-Hop Regular Virtual Topologies for Optical Packet Switching Over Arbitrary Physical Topologies," *IEEE Journal of Lightwave Technology*, vol.20,no.8,pp.1248-1263,August 2002.

[2] **D. Cotter et al.,** "Nonlinear optics for high-speed digital information processing," Science, vol. 286, pp. 1523–1528, Nov. 1999.

[3] **M. G. Hluchyj and M. J. Karol,** "Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks," *J. LightwaveTechnol.*, vol. 9, pp. 1386–1397, Oct. 1991.

[4] **B. Mukherjee,** "WDM-based lightwave networks—Part II: Multihop systems," *IEEE Network*, pp. 20–31, July 1992.

[5] **K. N. Sivarajan and R. Ramaswami,** "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Trans. Networking*, vol. 2, pp. 70–79, Feb.1994.

[6] **B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukherjee,** "Some principles for designing a wide-area optical network," in *Proc. IEEE Infocom*, 1994, pp. 110–119.

[7] **I. Chlamtac, A. Ganz, and G. Karmi,** "Lightnets: Topologies for highspeed optical networks," *J. Lightwave Technol.*, vol. 11, pp. 951–961,May/June 1993.

[8] **N. F. Maxemchuk,** "The Manhattan Street Network," in Proc. IEEE Globecom, 1985.

[9] **N. F. Maxemchuk,** "Routing in the Manhattan Street Network," *IEEE Trans.Commun.*, vol. Com-35, pp. 503–512, May 1987.

[10] **C. P. Ravikumar, T. Rai, and V. Verma,** "Kautz graphs as attractive logical topologies in multihop lightwave networks," *Comput. Commun* . , vol. 20, pp . 1259 – 1269 , 1997.

[11] **A. Rosenberg,** "Data encodings and their costs," Acta Informatica, vol. 9, pp. 273–292, 1978.

[12] **R. Duttra and G. N. Rouskas,** "A Survey Of Virtual Topology Design Algorithms for Wavelength Routed Optical Networks," Dept. of Computer Science, N.C. State University, TR-99-06, 1999.

[13] **S. Banarjee and B. Mukherjee,** "Algorithms for Optimized Node Arrangements in Shufflenet based Multihop Lightwave Networks," *Proc. IEEE INFOCOM'93*, pp. 557-564,1993.

[14] **S. Banarjee, B. Mukherjee and D. Sarkar,** "Heuristic Algorithms for constructing optimized structures of linear multihop lightwave networks," *IEEE Trans. Commun.*, vol. 42, pp. 1811–1826, Feb. /Mar. /Apr. 1994.

[15] **K. Yeung and Y. Tum,** "Node Placement Optimization in Shufflenets," *IEEE/ACM Transactions on Networking*, vol 8, pp. 319-324, June 1998

[16] **D. Cotter, J. K. Lucek, and D. D. Mercenac,** "Ultra-high-bit-rate networking: From the transcontinental backbone to the desktop," IEEE Commun. Mag., pp. 90–95, Apr. 1997.

[17] **R.Ramaswami,K.N.Sivarajan, "***Optical Networks: A Practical Perspective" 646-647*, 2nd edition. Academic Press,San Diego,CA,2002.

[18] **F. Chevalier, D. Cotter, and D. Harle,** "A new packet routing strategy for ultra-fast photonic networks," in Proc. IEEE Globecom, 1998.

[19] **Fai Siu, Rocky K.C. Chang,** "Effectiveness of optimal node assignments in wavelength division multiplexing networks with fixed regular virtual topologies," Computer Networks, pp.61-74, 2002

[20] **S. Sahni, T. Gonzalez,** "P-complete approximation problems," J.ACM 23, pp. 555-565,1976

[21] **T. D. Todd, Z. Khurshid, A. M. Bignell and S. Sivakumaran,** "Photonic Multihop Bus Networks," Proc., IEEE INFOCOM '91, Bal Harbor, FL, pp. 981-990, April 1991.

[22] **J.-F. P. Labourdette and A. S. Acampora,** "Logically rearrangeable multihop lightwave networks," IEEE Trans. Commun., Vol. 39, No. 8, pp. 1223-1230, Aug. 1991.

[23] **K. Nurmela,** "Constructing combinatorial design by local search," Research Report No.27 in Series A, Department of Computer Science, Helsinki University of Technology, Finland 1993

[24] **M.R. Wilhelm, T.L. Ward,** "Solving quadratic assignment problems by simulated annealing," IEE Trans. 19, pp.107-119, 1987

[25] **G.H. Sasaki, B. Hajek,** "The time complexity of maximum matching by simulated annealing," J.ACM 35, pp.387-403,1988

APPENDIX A

Uniform Random Traffic "t1"

.0	.8	.9	.1	.4	.8	.3	.4
.2	.0	.9	.2	.9	.0	.2	.9
.6	.6	.0	.2	.5	.7	.2	.9
.5	.8	.9	.0	.4	.4	.7	.6
.9	.9	.1	.3	.0	.8	.3	.5
.8	.7	.4	.2	.5	.0	.5	.9
.5	.2	.8	.0	.2	.7	.0	.8
.0	.4	.0	.7	.7	.4	.7	.0

Uniform Random Traffic with Amplitude=90 "t2"

0	5.2	51.8	87.6	29.9	22.9	84.6	77.6
42.7	0	25.9	83.1	26.3	53.3	39.1	2.8
1.6	65.1	0	86.3	36.	47	84.2	32.9
73.8	3	8.3	0	3.1	11.6	44.7	8.4
5.2	62.6	26.3	85.5	0	.3	87.4	30.1
8.1	19.9	74	28.6	19.8	0	86.2	.7
62.7	56.3	89.5	23.9	50.9	32.5	0	73.4
13.2	46.2	6.1	5.5	78.3	89.5	58.3	.0

Non-Uniform Random Traffic with Skew Factor=100 "t3"

0	.8	.9	.1	.4	.8	.3	.4
.2	0	91.7	20.3	.9	0	.2	86
.6	.6	0	.2	.5	.7	.2	.9
.5	.8	.9	0	.4	.4	.7	.6
.9	.9	.1	.3	0	.8	.3	.5
76.2	.7	.3	.2	52.5	0	54.2	.9
.5	.2	.8	0	.2	.7	0	.8
0	.4	0	.8	.7	.4	.7	0

Varying Range Traffic with R=20 “t4(20)”

0	110.1	105.7	100.3	90.5	114.4	117	104.4
98.6	0	107.9	86.7	81.9	111.4	84.8	114.1
106.3	89.2	0	85.5	109	109.2	119	89.6
87.8	81.2	114.4	0	83.9	91.1	88.8	81.3
103.5	93.2	106.1	94.8	0	98.2	100.3	85.7
87.6	108	97.8	85.8	103.3	0	111.9	95.4
104.9	106.2	91.1	115.7	107.9	92.7	0	91
88.3	91.8	109.3	119.7	89.6	106.1	119.5	0

Varying Range Traffic with R=50 “t4(50)”

0	79.2	113.9	90.3	109.9	145	114.6	99.7
145.4	0	127.8	54.5	66.3	65.4	104.9	71.3
99.6	66.4	0	146.1	53.6	78	119.9	62.8
129.4	146.9	147.6	0	117.2	135.1	116.7	59.7
72.9	82.8	88.4	61.6	0	130.5	147.2	68.1
67.3	71.9	76.1	147.4	130.8	0	141.9	87.4
115.7	94.1	126.3	140.7	100.2	142.8	0	107.2
111.6	108.5	60.4	146.2	50.9	105.6	102.1	0

Varying Range Traffic with R=90 “t4(90)”

0	96.1	133.1	24.1	181	51.7	106.9	163.6
123.4	0	70.4	20.7	84.6	94	35.1	115.2
146.3	37.6	0	37	96.1	55.6	31.2	26.8
42.7	184.2	126.8	0	56.9	137.6	159.2	142
136.1	143	116	187.7	0	73.7	172.9	165
56.2	88.4	37.8	136.2	41.5	0	66.3	189.5
167.5	56.5	76.2	111.8	185.7	167.8	0	67.4
111.4	14.9	58.8	34.8	141.2	91.3	66.6	0

BIOGRAPHY

Burak Uslu was born in Istanbul in 1978. He graduated from Kabataş High School in 1997 and attended Faculty of Electrical and Electronic Engineering, Istanbul Technical University. He graduated from Department of Control and Computer Engineering in 2001 and he has been working as a design engineer in Kontel Electronics Inc. since then. He is married for a year.