Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 15, No. 1, 2023 Article ID IJIM-1512, 11 pages
DOI: http://dx.doi.org/10.30495/ijim.2022.60117.1512
Research Article

GSOCPP Optimization for Predicting the Proper Number of Controllers in SDN

A. Amin [*], M. Jahanshahi [†‡], M. R. Meybodi [§]

Received Date: 2022-06-15 Revised Date: 2022-11-15 Accepted Date: 2022-12-02

Abstract

In Software Defined Network (SDN), the controller layer that is separated from the data layer is responsible for all system functionalities. However, centralized solutions suffer from single-point-of-failure and bottleneck problems. Several controllers are used to increase availability and performance in large networks to solve the aforementioned problems. One of the main concerns is finding the optimal number of controllers and their locations, which is known as an NP-hard problem. To do this, in this paper, in addition to presenting an efficient algorithm based on Garter snake algorithm (GSO), a new statistical analysis for determining the number of controllers is figured out

Keywords : Software Defined Network; heuristic algorithm; GSO; Multiple Linear Regression; Controller Placement Problem.

1 Introduction

In a software defined network (SDN) architecture, the control layer is separated from the data layer for better network management and planning considerations [1]. Consequently, policies are implemented very quickly and effectively despite their inherent centralized nature. However, having a controller in the network will cause two problems: single-point-of-failure and bottleneck. Thus, several controllers are inevitably used in large networks. By having many controllers, these issues can be tackled. However, multiple controllers raises new considerations such as the optimal number of controllers and their placements [2] [4]. The switch-controller propagation delay is just taken into account in some studies in order to determine the optimal number and location of the controllers. Furthermore, controllers are presumed to be without capacity and limitations. However, since delays between two controllers and their capacities affect the performance of a network, ignoring them may not lead to realistic results. Thus, this research concentrates mainly on the switch-controller propagation delays, while inter-controller delays are regarded as the primary limitations for controllers. Since this issue

[*]Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.
[†]Corresponding author. mjahanshahi@iauctb.ac.ir, Tel:+98(21)4600046.
[‡]Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.
[§]Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran.

has been elaborated as NP-hard in 2012[3], meta-heuristic algorithms are one of the most widely used and suitable solutions to cope with this issue. For example, in [5], GSOCPP was used to solve the CPP problem. Simulation results show that GSOCPP outperforms Firefly, K-Means++, and PSO. The rest of the paper is organized as follows: the related works are described in Section 2. Sections 3 and 4 discuss the GSO and GSOCPP algorithms, respectively. In section 5, Multiple Linear Regression is described. Section 6 is dedicated to emphasize the proposed algorithm. The experimental scenarios are presented in section 7, and finally concluding remarks are given in Section 8.

2 Related works

In some studies, inter-controller delays have not been considered, while this kind of delay definitely affects the network performance. Therefore, researchers have taken into account this point recently [6]. On the other hand, although most of the conducted researches indicates that controller capacity is another significant factor in speeding up the packet delivery, in some researches the capacity is considered unlimited leading to inaccurate or not applicable in all situations. In addition, network size is a vital factor in the problem. Due to the fact that meta-heuristic algorithms are appropriate solutions for large-scale networks in NP-Hard problems, we apply GSOCPP in the research. In [7] to [17] the controllers have unlimited capacity and inter-controller delays have not been considered. All of these researches focus on WAN networks. Clearly, among these studies, [7] to [10], [14], and [15] have used some meta-heuristic algorithms to overcome the CPP, while spectral clustering [17], K-means clustering [16], K-media clustering [12], and Genetic algorithm [13] have been used to determine the controller's location. Additionally, [18] to [20] have ignored the inter-controller delays and the capacity of the controllers in presence of small networks, among which [19] have used BLP for CPP and [18] and [20] used clustering based idea. [21] to [27] considered controllers with unlimited capacity in WAN networks. However, they considered the inter-controller delays. To overcome the CPP, meta-heuristic algorithms have been used in [22] [23] [26] and spectral clustering [24] [25], K-means clustering and Density clustering [27]. Likewise, [28] to [38] considered both the capacity of the controllers and the inter-controller latencies. Of course, [32] and [34] do not focus on WANS, so [34] was used to program Model Linear, and [32] miscellaneous was used to find the controller. Since [36] [37] [33] [31] [29] [35] [38] [28] researches have focused on large networks and all of them used meta-heuristic algorithms, and in [30] K-means clustering was used to overcome the large network. In [39] to [46] the controllers have also been considered to capacity; However, inter-controller delays have been ignored. Of course, the goal is not WANS in [45] and [46]. In [45] clustering-based switch migration algorithm is proposed, and in [46] a mathematical model is used by CPLEX for solving CPP. This research is categorized as follows:

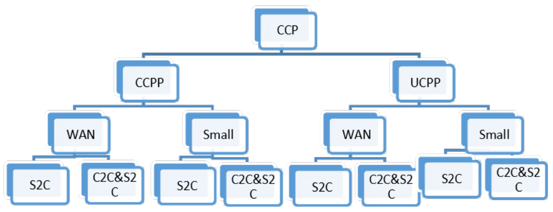


Fig. 1. classification of the related works in the scope of CPP

3 GSO algorithm

The Garter snake algorithm, as the name implies, is inspired by the behavior of a snake called Garter. In this meta-heuristic algorithm, three search agents (male, female, and she-male) are considered and each behaves differently. This algorithm formulation is as follows:

$$N_{male} = N_{pop} \times (0.95 - rand(0,1)) \quad (3.1)$$

$$N_{female} = N_{male} - N_{pop} \quad (3.2)$$

The population of Nmale is calculated by (3.1), which is 95% of the total population or Npop. The she-male element has been regarded as a random number in the male population. There are three subgroups of snakes, M, F, and SM, that are calculated in (3.2). The total snake population, which has N components, includes the following:

$$\mathcal{F} \;=\; \{f_1, f_2, ..., f_N\} \qquad (3.3)$$

$$\mathcal{M} \;=\; \{m_1, m_2, ..., m_N\} \qquad (3.4)$$

$$S\mathcal{M} \;=\; \{sm_1, sm_2, ..., sm_N\} \qquad (3.5)$$

Which in that:

$$\begin{aligned} &S = \mathcal{F} \cup \mathcal{M} \cup S\mathcal{M} \\ &such\ that\ \ S = \ \{s_1, s_2, ..., s_N\} \end{aligned} \qquad (3.6)$$

In this algorithm O(x) is objective function, so that consider the following unconstrained:

$$min\ O(x), \qquad x_1 \leq x \leq x_u \qquad (3.7)$$

Weight and appearance affect the efficiency of snakes communication in their biological representation. For every individual, the quality of the solution (w_i), is determined as follows:

$$w_i = \frac{worst_s \ - \ o(s) \ + t_s}{worst_s \ - \ best_s \ + t_s} \qquad (3.8)$$

The function O(x) actually consists of original objective function minus a penalty function P(x).

$$O(x) \;=\; [\mathcal{H}(x)] \;+\; P(x) \qquad (3.9)$$

If the function $O(_{si})$ is the fitness value obtained from the evaluation of the snake position si according to the objective function O(x). The best value ($best_s$) and the worst value ($worst_s$) are defined as following:

$$\begin{aligned} &best_s \ = min((Sk)) \\ &k \in \{1, 2, ..., N\} \end{aligned} \qquad (3.10)$$

$$\begin{aligned} &worst_s \ = max((Sk)) \\ &k \in \{1, 2, ..., N\} \end{aligned} \qquad (3.11)$$

Also, t_s=1/T_i indicates that T_i is the snake's temperature, i. In the snake population, male snakes try to mate with female ones. They try to choose female snakes who are the heaviest in weight. Garter snakes also like heat, so they stick together to generate heat.

$$\Gamma(t) \;=\; rand(0.1) w_i exp(-t_i \lambda^2 \ _{ij}) \qquad (3.12)$$

Eq. 3.12 demonstrates the coefficient of the she-male element in this algorithm. Where rand (0.1) is a random number, because neither the number of she-male nor happening again in next generation is known. Also $\lambda\ _{i,j}$ is defined as the Euclidean difference between a snake (i) and the she-male (j) such that, , $\lambda\ _{i,j} = \|si - sj\|$. Now let discuss four possible situations.

1. Individual i is a female snake, hence w_i ,t_i and λ are small. In this condition Γ_j(t) has no movement, because this individual snake is in the optimal position.

$$\Gamma(t) = rand(0, 1) w_{min.}\ t_{min} exp(-\lambda^2_{i,}) \qquad (3.13)$$

2. Individual i is male and t_i , λ_i and w are big. The calculation of Γj(t) is as follows:

$$\Gamma(t) \;=\; rand(0.1) w_{max.} t_{max} exp(-\lambda^2 \ _{i,}) \qquad (3.14)$$

3. Individual i is male and w_i is small and t_i and λ are big. In this condition Γ(t) is obtained by the formula:

$$\Gamma(t) \;=\; rand(0.1) w_{min.} t_{max} exp(-\lambda^2 \ _{i,}) \qquad (3.15)$$

4. Individual i is male and w_i is big, however t_i and λ are small. In this case Γ(t) is given by the formula:

$$\Gamma(t) \;=\; rand(0.1) w_{max.} t_{min} exp(-\lambda^2 \ _{i,j}) \qquad (3.16)$$

The first step of this algorithm implementation is completely random, and it will be as follows:

$$\begin{aligned} &f^0_{i,j} \;=\; x^l_j + (0, 1)\ .\ (x^u_j \ - \ x^l_j) \\ &i = 1, 2, ..., N_f;\ j = 1, 2, ..., n \end{aligned} \qquad (3.17)$$

$m^0_{k,j} \ = x^l_j + rand(0, 1)\ .\ (x^u_j \ - \ x^l_j)$
k=1,2,...,N_m; j=1,2,...,n

sh$^0_{z,j}$ $= x^l_j + (0,1)\,.\,(x^u_j \;-\; x^l_j)$
z=1,2,...,N$_s h$; j=1,2,...,n

Initial number of the population of female, male, and she-male are obtained respectively.
In the next step, α produce a random number between 0 and 1, and the algorithm will be executed as follows [4]:

$$f_i^{k+1} \;=\; f_i^k \;+\; \alpha\Gamma_i(s - f_i^k)+(0,1)$$

$$m_i^{k+1} = m_i^k \;+\; \alpha\Gamma_i\frac{m_i^k+w_i}{w_i}+\text{rand}(0,1)$$

$$sh_i^{k+1} = sh_i^k + rand(0,1) \tag{3.18}$$

4 GSOCPP algorithm

In [5], the proposed algorithm for CPP is modelled based on GSO algorithm. In this research switchcontroller propagation delays and inter-controller delays are considered to achieve the sub-optimal number of controllers as well as their locations based on the haversine distance [51]. The fitness value is the summed up of the delays for each controller is calculated as following:

$$w_i = \frac{worst_s - \pi_{s_i}^{total-delay} + norm_{s_i}}{worst_s \;-\; best_s \;+ norm_{s_i}} \tag{4.19}$$

In this research, assume T$_i$ is the temperature of snake i and t$_{s_i}$= 1/T$_i$, The variable Ti, also the temperature of controller i, is calculated as the flow-setup time using Eq. 4.20 in [51]. This value depends on the number of switches, which is managed by the controller ($|c_s^i|$), the processing rate value of the controller (μ_c), and the total rate of switches request rates (λ):

$$T_i = \frac{|c_s^i|+1}{2(\mu_c - \lambda|c_s^i|)}, \qquad \mu_c \;>\; \lambda C_s^{\,i} \tag{4.20}$$

According to Eq. 4.20, the controller temperature is related to the number of the switches and the controllers processing rates. Furthermore,$norm$, the normalized version of the inverse temperature for snake i, is the numerical value that represents the weight of controllers as shown in (4.21). The calculation of $norm_{s_i}$ is as follows:

$$norm_{s_i} = \frac{t_{s_i} - min(t_{s_i})}{max(t_{s_i}) - min(t_{s_i})} \tag{4.21}$$

Male snakes attempt to find the best partner which is usually the heaviest weight female snake in the mating operator. These properties can be summarized by 22:

$$\Gamma_i(t) = w_i t_i exp(-\lambda_{ij}^2) \tag{4.22}$$

Where λ_{ij} is the Euclidean distance between the temperature of the individual snake i and she-male snake j; that is to calculate, $\lambda_{i,j} = \|S_{snakei} - S_{snakej}\|$.
In this paper when a controller is qualified, it means that the controller is in an appropriate position, and it doesnt need to move to the next step:

1. If the snake (controller) i is a female and its t_i, w_i and λ are small, then this individual has already the best position in the network.
2. If the snake (controller) i is a male and its t_i, w_i and λ are large, then this individual also has the best position in the network.
3. If the snake (controller) i is a male and its t_i and λ are large and w_i is small, it has a low propagation delay with few switches included in the subset of its cluster.
4. If the snake (controller) i is a male, t_i and λ are small and w_i is large, then this individual cannot participate in the mating ball. Subsequently, these controllers are removed from the set of Garter snakes (available controllers), and are replaced by a new male snake (controller).

In the next step, depending on the gender of the snake, the controllers (snakes) are used to move the controller to an adjacent position according as follows:

$$f_i^{k+1} = \begin{cases} f_i^k \\ rand(S-F), otherwise \end{cases} \tag{4.23}$$

$$m_i^{k+1} = \begin{cases} m_j^k,\ rand(v)|v\ is\ neighbor\ of\ m_i^k \\ rand(v)|v\ \notin SM\ and\ v \notin S \end{cases}$$
$$sh_i^{k+1} = rand(v|v \notin SM)$$

If a female snake (controller) has properties according to 1, it means its position is a suitable location. Otherwise, other female snakes randomly select the rest of network nodes. If a male snake (controller) has properties according to one of the conditions 2 or 3, also remain in their previous location. The male snake (controller) that is according to condition 4 is removed and then a node of the other nodes is randomly selected. The rest of n-set controllers that are not contained in these sections just move into one of neighboring nodes in the next step.

5 Multiple Linear Regression

Multiple Linear Regression (MLR) is a sample of supervised machine learning techniques that can predict dependent variable using independent variables. In reality, MLR computed linear relationship between dependent and independent variables. In [47] to [49] MLR is considered as one the estimation techniques.
MLR is defined by the following formula:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon \quad (5.24)$$

$X1, X2, , X_n$ are regressions. β_0 is an intercept parameter and β_1, β_2, ...,β_n are regression coefficients, and ε is the error component. Regression may be used to determine the effect of an independent parameter on the dependent parameter, or to predict the effect of an independent parameter on the dependent parameter. [47]
The main assumptions of MLR algorithms are as follows:

1. Independent coefficients have no relationships with each other.

2. The result variable must have a linear relationship with any predictor variables.

3. Observations should be randomly selected, however they have to selected independently.

4. The residual distribution should be normal.[50]

6 MGSOCPP: Proposed algorithm

Since CPP is more investigated in large scale SDNs, meta-heuristic algorithms are one of the suitable solutions to solve CPP as an NP-hard problem. GSO was used for solving CPP in [5] that outperforms firefly and PSO algorithms. Hence, in the research, initially this algorithm is improved and then, a statistical analysis called MGSOCPP (Modify-GSOCPP) is recommended for specifying the number of controllers using MLR. In the following section the objective function is presented and then in section 7 the MGSOCPP algorithm is proposed.

Objective function
In the research, the objective function and problem constraints are defined as follows:

1. To consider the capacity of the controller

2. To consider the number of controller ports

$$\mu_c \geq \sum_{v \in C_s} \lambda_{C_s} \quad (6.25)$$

$$\{c^i \neq c^j and\ c_s^i \cap c_s^j = \varnothing \mid c^i, c^j \in C\} \quad (6.26)$$

$$c_{port}^j \geq \sum_{v \in C_s} |C_s^j| \quad (6.27)$$

First this location is due to the delay between the controller to the switch, π^{aveC2S} and the controller to the controller π^{aveC2C} , then we consider the delay factor to be the sum of these two delays ($\pi_{delay}^{total} = \pi^{aveC2S} + \pi^{aveC2C}$).The average are calculated as follows:

$$\pi^{aveC2S} = \frac{a}{n} \sum_{c \in V} \min_{c \in C} d(v, c) \quad (6.28)$$

$$\pi^{avgC2C} = \frac{b}{\binom{n_c}{2}} \sum_{c_i, c_j \in S'} sp_c \quad (6.29)$$

7 MGSOCPP

Subsequently to what was discussed about GSO algorithm in section 3, Garter Snakes are categorized into three types of Male, Female and She-Male randomly. In addition, the same as those controllers are chosen out of three types randomly. Since controllers are only different in terms of capacity and number of ports, gender distinction is not a logical approach; Despite the fact that female and male snakes compare their situation in relation to She-male gender, considering a controller randomly as she-male will not present a practical algorithm. Hence, MGSOCPP algorithm expects the following changes:

1. Female Garter snakes do not exist and Male and She-Male snakes are only available.
2. She-male snake is not chosen randomly; in change, The Garter snake with the highest temperature is considered as a She-male.

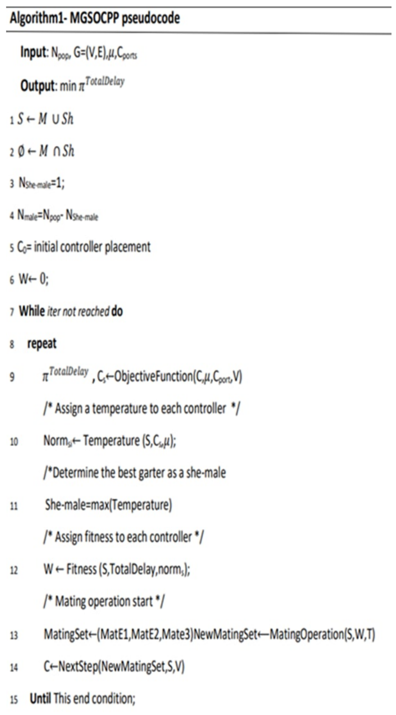
In order to have a match toward MGSOCPP, some changes are needed in terms of mating in the main research; to assert the qualified controller as having the proper position. Therefore, there is no need to transform the controller in later steps:

1. If t , w and λ are maximum in i-operator, then the operator is in a proper position.
2. If w is minimum and t and λ are maximum in i-operator, then the operator is in a proper position. Because the controller with less weight(less processing rate) was able to handle more switches.
3. If t and λ are minimum and w is maximum in i-operator, then the operator is not in a proper position. Because the controller with more weight (more processing rate) was able to handle less switches. Therefore, the expected controller (Snake) could not participate in mating and will be removed and substituted in the following steps.

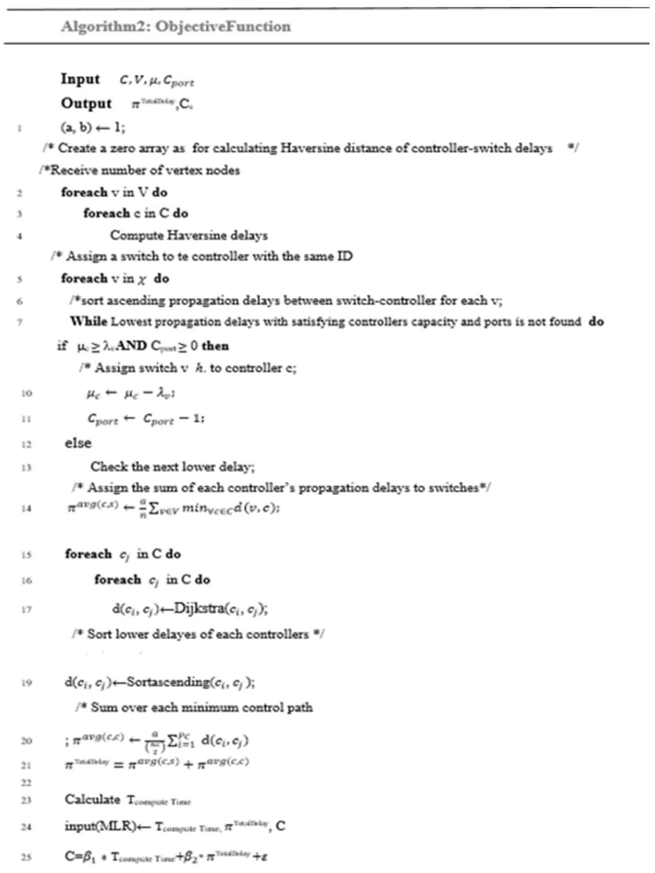
Next, based on the value achieved in the process of mating, controllers are expected to have variable movements in the network similar to Eq. 7.30:

$$snake_i^{k+1} = \begin{cases} snake_i^k \\ rand(v)|v \text{ is neighbor of } snake_i^k \text{ and } v \notin \mathbb{S} \\ rand(v)|v \notin \mathbb{S}, \end{cases} \tag{7.30}$$

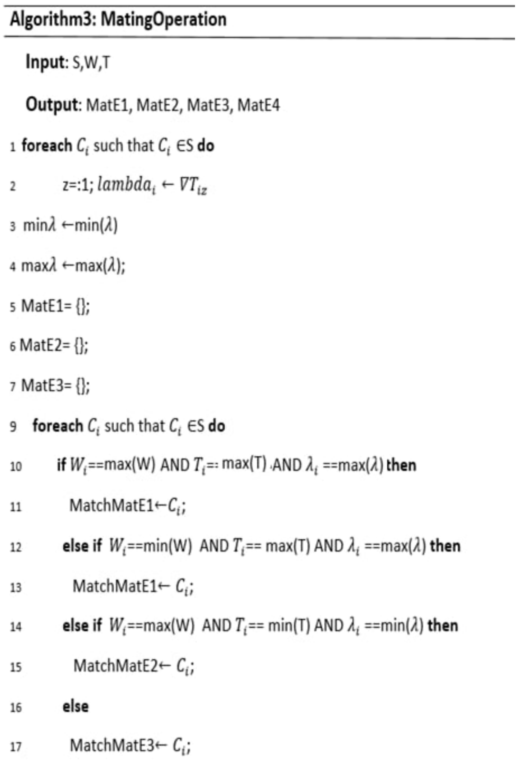
The controllers which are in first and second situation, remain unchanged. Controllers that are in the third situation are removed and are randomly placed in a node throughout the network. The remaining controllers from set N are not included in these categories and they will be only moved toward one of its neighbors. Therefore, they have the possibility to achieve better results.
Finally, it is noted, the controller with maximum temperature will be considered as She-male snake in the proposed algorithm.
In the proposed algorithm, Temperature, Computation and Fitness Assignment are defined according to [5] and the other modules are demonstrated with the Algorithm 1 to 4.


Algorithm1- MGSOCPP pseudocode

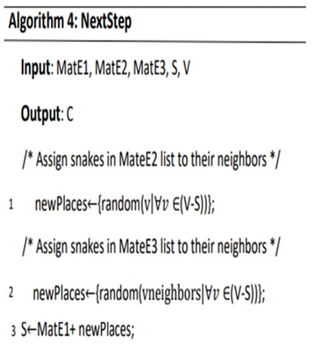
```
  Input: $N_{pop}$, $G=(V,E)$,$\mu$,$C_{ports}$
  Output: min $\pi^{TotalDelay}$
1 $S \leftarrow M \cup Sh$
2 $\emptyset \leftarrow M \cap Sh$
3 $N_{She\text{-}male}=1$;
4 $N_{male}=N_{pop}-N_{She\text{-}male}$
5 $C_0$= initial controller placement
6 W← 0;
7 While iter not reached do
8   repeat
9       $\pi^{TotalDelay}$, $C_s$←ObjectiveFunction($C$,$\mu$,$C_{port}$,V)
        /* Assign a temperature to each controller */
10      $Norm_{si}$← Temperature (S,$C_s$,$\mu$);
        /*Determine the best garter as a she-male
11      She-male=max(Temperature)
        /* Assign fitness to each controller */
12      W ← Fitness (S,TotalDelay,$norm_s$);
        /* Mating operation start */
13      MatingSet←(MatE1,MatE2,Mate3)NewMatingSet←MatingOperation(S,W,T)
14      C←NextStep(NewMatingSet,S,V)
15  Until This end condition;
```



```
Algorithm2: ObjectiveFunction

Input   $C, V, \mu, C_{port}$
Output  $\pi^{TotalDelay}$, C.
(a, b) ← 1;
/* Create a zero array as  for calculating Haversine distance of controller-switch delays   */
/*Receive number of vertex nodes
foreach v in V do
    foreach c in C do
        Compute Haversine delays
/* Assign a switch to te controller with the same ID
foreach v in $\chi$ do
    /*sort ascending propagation delays between switch-controller for each v;
    While Lowest propagation delays with satisfying controllers capacity and ports is not found do
    if $\mu_c \geq \lambda_v$ AND $C_{port} \geq 0$ then
        /* Assign switch v  $h$. to controller c;
        $\mu_c \leftarrow \mu_c - \lambda_v$;
        $C_{port} \leftarrow C_{port} - 1$;
    else
        Check the next lower delay;
    /* Assign the sum of each controller's propagation delays to switches*/
    $\pi^{avg(c,s)} \leftarrow \frac{a}{n} \sum_{v \in V} min_{\forall c \in C} d(v, c)$;

    foreach $c_j$ in C do
        foreach $c_j$ in C do
            d($c_i$, $c_j$)←Dijkstra($c_i$, $c_j$);
    /* Sort lower delayes of each controllers */

    d($c_i$, $c_j$)←Sortascending($c_i$, $c_j$ );
    /* Sum over each minimum control path
    ; $\pi^{avg(c,c)} \leftarrow \frac{a}{\binom{n c}{2}} \sum_{i=1}^{pc}$ d($c_i$, $c_j$)
    $\pi^{TotalDelay} = \pi^{avg(c,s)} + \pi^{avg(c,c)}$

    Calculate $T_{compute\ Time}$
    input(MLR)← $T_{compute\ Time}$, $\pi^{TotalDelay}$, C
    C=$\beta_1$ * $T_{compute\ Time}$+$\beta_2$* $\pi^{TotalDelay}$ +$\varepsilon$
```



```
Algorithm3: MatingOperation

Input: S,W,T
Output: MatE1, MatE2, MatE3, MatE4
foreach $C_i$ such that $C_i$ ∈S do
    z=:1; $lambda_i \leftarrow \nabla T_{iz}$
min$\lambda$ ←min($\lambda$)
max$\lambda$ ←max($\lambda$);
MatE1= {};
MatE2= {};
MatE3= {};
foreach $C_i$ such that $C_i$ ∈S do
    if $W_i$==max(W) AND $T_i$== max(T) AND $\lambda_i$ ==max($\lambda$) then
        MatchMatE1←$C_i$;
    else if $W_i$==min(W) AND $T_i$== max(T) AND $\lambda_i$ ==max($\lambda$) then
        MatchMatE1← $C_i$;
    else if $W_i$==max(W) AND $T_i$== min(T) AND $\lambda_i$ ==min($\lambda$) then
        MatchMatE2← $C_i$;
    else
        MatchMatE3← $C_i$;
```



```
Algorithm 4: NextStep

Input: MatE1, MatE2, MatE3, S, V
Output: C
/* Assign snakes in MateE2 list to their neighbors */
newPlaces←{random(v|∀v ∈(V-S))};
/* Assign snakes in MateE3 list to their neighbors */
newPlaces←{random(vneighbors|∀v ∈(V-S))};
S←MatE1+ newPlaces;
```

Finally, delays of switchcontroller propagation and inter-controller are introduced as explanatory variables, and number of controllers is regarded as a dependent variable in SPSS.

Table 1: network specifications

Aarnet Topology		
Number of controllers	Process rate	Number of ports
3	733	8
4	575	6
5	480	5
6	416	5
7	371	4
8	337.5	4
9	311	4
10	290	3
DFN Topology		
Number of controllers	Process rate	Number of ports
3	1800	18
4	1375	13
5	1120	11
6	950	9
7	829	8
8	738	7
9	670	6
10	610	5
Colt Topology		
Number of controllers	Process rate	Number of ports
3	5200	52
4	3925	39
5	3160	31
6	2650	26
7	2285	22
8	2012	20
9	1800	18
10	1630	16
Cogent Topology		
Number of controllers	Process rate	Number of ports
10	1960	20
15	1340	14
20	1030	11

8 Performance Evaluation

This section describes the conditions of the conducted experiment. The hardware and software that were used for the algorithms are as listed:
[Intel Core i7 CPU - 32GB RAM Ubuntu 19.04-64bit OS]
During the execution of the proposed algorithm, variables in Eqs. 6.27 and 6.28 are the same as each other and both values are considered as 1. MGSOCPP was ran for 100 times using the parallel processor.
Datasets of Internet Topology Zoo (ITZ) were used for these evaluations; Aarnet Network Topology in a small scale, DFN in a medium scale, Colt in a large scale, and Cognet in a very large scale are used throughout the experiments. These networks specifications are given in Table 1. Also, their features are illustrated in Table 2.

Table 2: features of topologies

Network name	Geographical area	Network location	Nodes number	Links number
Aarnet	state	Australia	19	24
DFN	state	Germany	63	89
Colt	continental	Western Europe	153	191
Cogent	intercontinental	Europethe USA	197	245

In addition, Python programming language was used for simulations. Then, the results are exported to SPSS software to present a statistical analysis using Multiple Linear Regression. In the right side of the chart is GSO and the left side is dedicated to MGSO. From the charts, since in the proposed idea, female-type snakes are ignored and She-Male is specified using aforementioned criterion, computation time of MGSOCPP will go up with the increase of number of controllers.

The number of controllers in this research is set to 3 to 10 in Aarnet topology with the same processing rate. In GSOCPP, as it is shown in the Fig. 2, when the number of controllers changes from 9 to 10, a unexpectedly increase in computation time is observed. This behavior may be occurred at any time during simulations. In contrast, using MGSOCPP leads to smooth behavior of computation time.

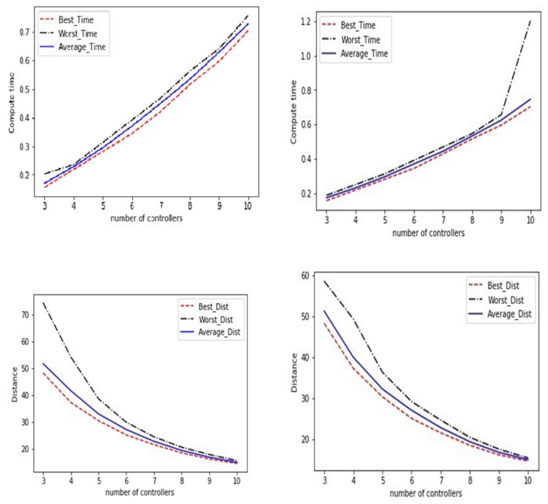


Fig. 2- computation time of GSOCPP versus MGSOCPP on Aarnet topology

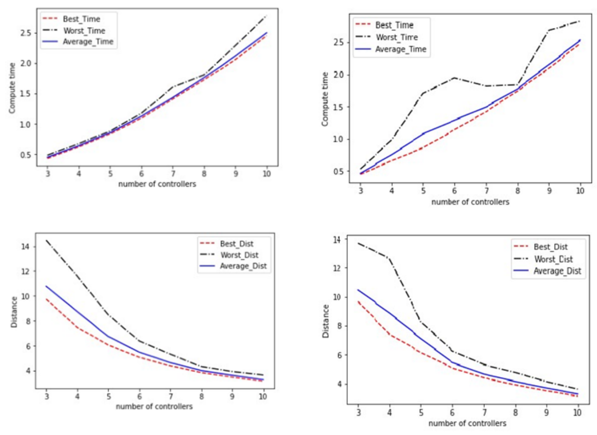


Fig.3- Computation time of GSOCPP versus MGSOCPP on DFN topology

Similarly, in DFN topology, the number of controllers is set to 3 to 10. This topology is in Germany, and the number of nodes is 63. In contrary to MGSOCPP, in GSOCPP implementation, when the number of controllers are varies from 3 to 5 and 8 to 10, computation time is not reasonable.

In Cogent topology, the number of controllers is set to 10, 15, and 20. Cogent is an intercontinental topology with 197 nodes. Since the ratio of the number of nodes to the number of controllers is large, the achieved results of GSOCPP is the same as MGSOCPP.

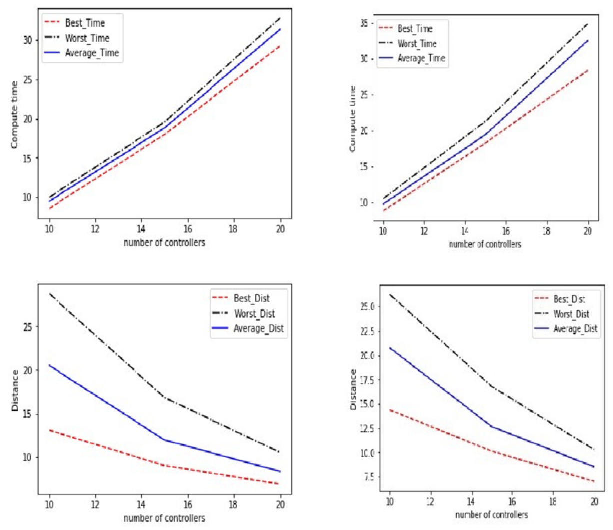


Fig.4- computation time of GSOCPP versus MGSOCPP on Cogent topology

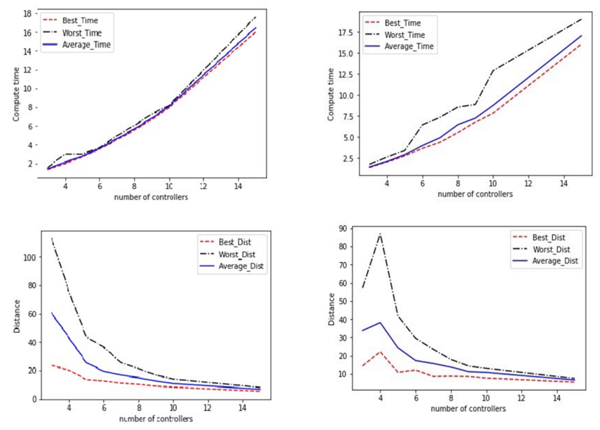
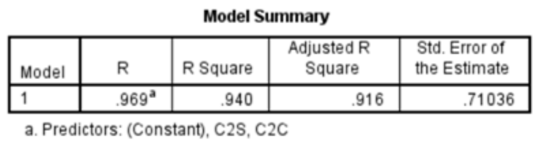


Fig.5- computation time of GSOCPP versus MGSOCPP on Colt topology

The number of controllers is set to 3 to 10, and 15 in Colt topology. As it is shown in the Fig. 5, when the number of controllers has changed from 6 to 10, an irrational trend is shown. In MGSOCPP, ignoring the female gender and choosing the best snake as the she-male snake leads to predictable computation time as it is expected.

Finally, in the research, MGSOCPP is used for determining the number of required controllers in CPP. Here, the switchcontroller propagation delay and inter-controller delay are given as input parameters, and the number of controllers as an output variable in SPSS.

Delays of switchcontroller propagation and inter-controller are considered as C2C and C2S. As it is illustrated in Fig. 6, Correlation and Determination coefficients are R=0.969 and RSquare=0.940 respectively. Also, Adjusted R Square is equal to 0.916 and because of that, regression model is preferable. Owing it to the fact that the closer the values are to 1, the more connection there is between dependent and independent values.



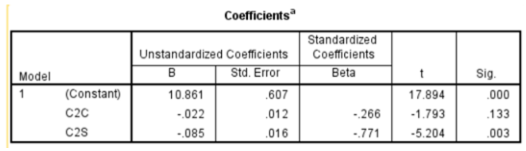
Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.969[a]	.940	.916	.71036

a. Predictors: (Constant), C2S, C2C

Fig.6- Regression

Now according to coefficients table, we can find a linear relation between the number of controllers C2C and C2S.



Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	10.861	.607		17.894	.000
	C2C	-.022	.012	-.266	-1.793	.133
	C2S	-.085	.016	-.771	-5.204	.003

Fig.7- Coeffician

According to these coefficients, the following regression model can be displayed.

$$\begin{aligned} &Controller\ number = \\ &10.861 - 0.022 * C2C - 0.085 * C2S \end{aligned} \tag{8.31}$$

In this research, we consider C2C and C2S as effective operators for finding the number of controllers.

9 Conclusion and Future Works

In large scale Software Defined Networks, several controllers have been deployed improving efficiency and better managing the network traffic. The sub-optimal number of controllers as well as their suitable positions are considered as an NP hard problem, which is one of outstanding challenges in SDN architecture. Heuristic algorithms have registered by many researchers as a de facto solution in the mentioned area. In current scientific studies, compared to firefly and GSO, the GSOCPP algorithm has proved superior performance. Therefore, in this fact-finding we first improved the GSO algorithm through ignoring gender factor. The results on the four topologies Aarnet, DFN, Cogent and Colt show that the proposed MGSOCPP algorithm outperforms GSO in terms of computation time. In the end, a statistical analysis is performed with the purpose of determining the sub-optimal required number of controllers. In that regard, we carried out this assessment by taking the advantage of Multi Linear Regression to comprehend the delays between switch-controller and inter-controller.

References

[1] M. F. Tuysuz, Z. K. Ankarali, D. Gpek, A survey on energy efficiency in software defined networks, *Computer Networks* 113 (2017) 188-204.

[2] O. Salman, I. H. Elhajj, A. Kayssi, A. Chehab, A comparative study, 18th Mediterranean Electro technical Conference (MELECON), *IEEE* 11 (2016) 1-6.

[3] B. Heller, R. Sherwood, N. McKeown, The controller placement problem, *ACM SIGCOMM Computer Communication Review* 42 (2012) 473-478

[4] M. Naghdiani, M. Jahanshahi, A New Solution for Solving Unconstrained Optimization Tasks Using Garter Snake's Behavior, *2017 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE* 3 (2017) 328-333

[5] S. Torkamani-Azar, M. Jahanshahi, A new GSO based method for SDN controller placement, *Computer Communications* 163 (2020) 91-108.

[6] A. K. Singh, S. Srivastava, A survey and classification of controller placement problem in SDN, *International Journal of Network Management* 28 (2018) 24-29

[7] M. Priyadarsini, J. C. Mukherjee, P. Bera, S. Kumar, A. H. M. Jakaria, M. A. Rahman, An adaptive load balancing scheme for software-defined network controllers, Computer Networks 164 (2019) 106-118.

[8] B. Yong, W. Muqing, S. Jing, Z. Min, Optimization Strategy of SDN Control Deployment Based on Simulated Annealing-Genetic Hybrid Algorithm, *2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE* (2018) 2238-2242.

[9] K. S. Sahoo, S. Sahoo, A. Sarkar, B. Sahoo, R. Dash, On the placement of controllers for designing a wide area software defined networks, *TENCON 2017 IEEE Region 10 Conference. IEEE* (2017) 3123-3128.

[10] Y. Li, Sh. Guan, C. Zhang, W. Sun, Parameter Optimization Model of Heuristic Algorithms for Controller Placement Problem in Large-Scale SDN, *IEEE Access* 8 (2020) 151668-151680.

[11] Z. Fan, J. Yao, X. Yang, Z. Wang, X. Wa, Multi-controller placement strategy based on delay and reliability optimization in SDN, *2019 28th Wireless and Optical Communications Conference (WOCC). IEEE* (2019) 1-5.

[12] K. S. Sahoo, S. Sahoo, S. Mishra, S. Mohanty, B. Sahoo, Analyzing controller placement in software defined networks, *IJCA Proceedings on National Conference on Next Generation Computing and its Applications in Computer Science and Technology NGCAST* (2016) 16-12.

[13] J. M. Sanner, Y. Hadjadj-Aoul, M. Ouzzif, G. Rubino, An evolutionary controllers' placement algorithm for reliable SDN networks, *2017 13th international conference on network and service management (CNSM). IEEE* (2017) 1-6.

[14] B. Zhang, X. Wang, Min Huang, Multiobjective optimization controller placement problem in internet-oriented software defined network, *Computer Communications* 123 (2018) 24-35.

[15] H. Li, R. E. De Grande, A. Boukerche, An efficient CPP solution for resilienceoriented SDN controller deployment, *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE* (2017) 540-549.

[16] A. Alowa, T. Fevens, Combined degreebased with independent dominating set approach for controller placement problem in software defined networks, *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). IEEE* (2019) 269-276.

[17] J. Lu, Z. Zhen, T. Hu, Spectral clustering based approach for controller placement problem in software defined networking, *Journal of Physics: Conference Series. IOP Publishing* (2018) 42-53.

[18] D. Tuncer, M. Charalambides, S. Clayman, G. Pavlou, On the placement of management and control functionality in software defined networks, *2015 11th International Conference on Network and Service Management (CNSM). IEEE* (2015).

[19] Y. Hu, T. Luo, W. Wang, C. Deng, On the load balanced controller placement problem in Software defined networks, *2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE* (2016) 2430-2434.

[20] H. Aoki, N. Shinomiya, Controller placement problem to enhance performance in multi-domain networks, *In Proc. ICN* (2016) 120.

[21] G. Ishigaki, R. Gour, A. Yousefpour, N. Shinomiya, J. P. Jue, Cluster leader election problem for distributed controller placement in sdn, *GLOBECOM* (2017) 1-6.

[22] K. S. Sahoo, A. Sarkar, S. K. Mishra, B. Sahoo, D. Puthal, M. S. Obaidat, B. Sadoun, Metaheuristic solutions for solving controller place-

ment problem in SDN-based WAN architecture, *ICETE 2017-Proceedings of the 14th International Joint Conference on e-Business and Telecommunications* (2017).

[23] V. Ahmadi, A. Jalili, M. Khorramizadeh, Multi-Objective Controller Placement Problem: issues and solution by heuristics, *International Journal of Computer Science and Information Security* 14 (2016) 543-547.

[24] K. S. Sahoo, B. Sahoo, R. Dash, M. Tiwary, Solving multi-controller placement problem in software defined network, *2016 International Conference on Information Technology (ICIT) IEEE* (2016) 188-192.

[25] P. Xiao, W. Qu, H. Qi, Z. Li, Y. Xu, The SDN controller placement problem for WAN, *In 2014 IEEE/CIC International Conference on Communications in China (ICCC) IEEE* (2014) 220-224.

[26] H. K. Rath, V. Revoori, S. M. Nadaf, A. Simha, Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game, *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. IEEE* (2014) 1-6.

[27] H. Xiaolan, W. Muqing, X. Weiyao, A Controller Placement Algorithm Based on Density Clustering in SDN, *In 2018 IEEE/CIC International Conference on Communications in China (ICCC). IEEE* (2018) 184-189.

[28] B. P. R. Killi, S. V. Rao, Capacitated next controller placement in software defined networks, *IEEE Transactions on Network and Service Management* 14 (2017) 514-527.

[29] S. Lange, S. Gebert, T. Zinner, P. TranGia, D. Hock, M. Jarschel, M. Hoffmann, Heuristic approaches to the controller placement problem in large scale SDN networks, *IEEE Transactions on Network and Service Management* 12 (2015) 4-17.

[30] L. Zhu, R. Chai, Q. Chen, Control plane delay minimization based SDN controller placement scheme, *In 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP) IEEE* (2017) 1-6.

[31] V. Ahmadi, M. Khorramizadeh, An adaptive heuristic for multi-objective controller placement in software-defined networks, *Computers & Electrical Engineering* 66 (2018) 204-228.

[32] M. T. I. Ul Huque, G. Jourjon, V. Gramoli, Revisiting the controller placement problem, *2015 IEEE 40th Conference on Local Computer Networks (LCN). IEEE* (2015) 450-453.

[33] G. Wang, Y. Zhao, J. Huang, W. Wang, The controller placement problem for software-defined networks, *2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE* (2016) 2435-2439.

[34] A. Sallahi, M. St-Hilaire, Optimal model for the controller placement problem in software defined networks, *IEEE communications letters* 19 (2014) 30-33.

[35] B. Zhang, X. Wang, M. Huang, Multiobjective optimization controller placement problem in internet-oriented software defined network, *Computer Communications* 123 (2018) 24-35.

[36] F. Li, X. Xu, A discrete cuckoo search algorithm for the controller placement problem in Software Defined etworks, *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE* (2018) 292-296.

[37] C. Gao, H. Wang, F. Zhu, L. Zhai, S. Yi, Particle swarm optimization algorithm for controller placement problem in software defined network, *International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham* (2015) 44-54.

[38] Kh. Mostafa, V. Ahmadi, Capacity and load-aware software-defined network controller placement in heterogeneous environments, *Computer Communications* 129 (2018) 226-247.

[39] B. Zhang, X. Wang, L. Ma, M. Huang, Optimal controller placement problem in Internet-oriented software defined network, *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE* (2016) 481-488.

[40] A. Jalili, M. Keshtgari, R. Akbari, R. Javidan, Multi criteria analysis of controller placement problem in software defined networks, *Computer Communications* 133 (2019) 115-128.

[41] M. Ramasamy, S. Pawar, Pareto-optimal multi-controller placement in software defined network, *2018 3rd International Conference for Convergence in Technology (I2CT). IEEE*, (2018) 1-7.

[42] T. Hu, P. Yi, Z. Guo, J. Lan, Y. Hu, Dynamic slave controller assignment for enhancing control plane robustness in software-defined networks, *Future Generation Computer Systems* 95 (2019) 681-693.

[43] Y. Jimenez, C. Cervello-Pastor, A. J. Garca, On the controller placement for designing a distributed SDN control layer, *In 2014 IFIP Networking Conference. IEEE* (2014) 1-9.

[44] F. Bannour, S. Souihi, A. Mellouk, Calability and reliability aware SDN controller placement strategies, *13th International Conference on Network and Service Management (CNSM). IEEE* (2017) 1-4.

[45] L. Yao, P. Hong, W. Zhang, J. Li, D. Ni, Controller placement and flow based dynamic management problem towards SDN, *In 2015 IEEE International Conference on Communication Workshop. IEEE* (2015) 363-368.

[46] A. Sallahi, M. St-Hilaire, Expansion model for the controller placement problem in software defined networks, *IEEE Communications Letters* 21 (2016) 274-277.

[47] S. Rathaur, N. Kamath, U. Ghanekar, Software Defect Density Prediction based on Multiple Linear Regression, *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE* (2020) 434-439.

[48] Y. Singh, P. K. Bhatia, A. Kaur, O. Sangwan, Review of studies on effort estimation techniques of software development, *In Proc. Conference Mathematical Techniques: Emerging Paradigms for Electronics and IT Industries* (2008) 188-196.

[49] B. Boehm, C. Abts, S. Chulani, Software development cost estimation approachesA survey, *Annals of software engineering* 10 (2000) 177-205.

[50] M. A. Poole, P. N. O'farrell, The assumptions of the linear regression model, *Transactions of the Institute of British Geographers* (1971) 145-158.

[51] K. Sood, Y. Xiang, The controller placement problem or the controller selection problem?, *Communications and Information Networks* 2 (2017) 1-9.

Azam Amin received the B.Sc. degree from the Department of Computer Enginering, South Tehran Branch, Islamic Azad University, Tehran, Iran, in 2003, the M.Sc. degree from the Department of Computer Enginering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, in 2011, and now she is a Ph.D. candidate in Islamic Azad University (Central Tehran Branch), Tehran, Iran, all in computer engineering.

Mohsen Jahanshahi, senior member, IEEE, (SM'16) received the the Ph.D. degree from the Islamic Azad University (Tehran Science and Research Branch), Tehran, Iran, in 2011 in computer engineering. He joined the Department of Computer Engineering, Islamic Azad University (Central Tehran Branch) in 2005. He was promoted to Associate Professor in 2015. His research interests include performance evaluation, multistage interconnection networks, wireless networks, learning systems, mathematical optimization, and soft computing. Dr. Jahanshahi has been a member of Young Researchers and Elite club since 2012. Besides, he is currently dean of technical faculty.

Mohammad Reza Meybodi received the B.Sc. and M.Sc. degrees in economics from Shahid Beheshti University, Tehran, Iran, in 1973 and 1977, respectively, the M.Sc. and Ph.D. degrees in computer science from Oklahoma University, Norman, OK, USA, in 1980 and 1983, respectively. He was an Assistant Professor with Western Michigan University, Kalamazoo, MI, USA, from 1983 to 1985, and an Associate Professor with Ohio University, Athens, OH, USA, from 1985 to 1991. He is currently a Full Professor with the Computer Engineering Department, Amirkabir University of Technology, Tehran. His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing, and software development.