
Hybridisation of genetic algorithm with simulated annealing for vertical-handover in heterogeneous wireless networks

Shidrokh Goudarzi* and Wan Haslina Hassan

Communication System and Network (iKohza) Research Group,
Malaysia-Japan International Institute of Technology (MJIT),
Universiti Teknologi Malaysia,
Jalan Semarak, Kuala Lumpur 54100, Malaysia
Email: shidrokhgoudarzi@gmail.com
Email: wanhaslina.kl@utm.my
*Corresponding author

Seyed Ahmad Soleymani

Faculty of Computing,
Universiti Teknologi Malaysia,
UTM Johor Bahru, Johor 81310, Malaysia
Email: asseyed4@live.utm.my

Mohammad Hossein Anisi

Faculty of Computer Science & Information Technology,
Department of Computer System & Technology,
University of Malaya,
50603 Kuala Lumpur, Malaysia
Email: anisi@um.edu.my

Abstract: To provide the seamless mobility in heterogeneous wireless networks two significant methods, Simulated annealing (SA) and genetic algorithms (GAs) are hybrid. In this paradigm, vertical handovers (VHs) are necessary for seamless mobility. In this paper, the hybrid algorithm has the ability to find the optimal network to connect with a good quality of service (QoS) in accordance with the user's preferences. The intelligent algorithm was developed to provide solutions near to real time and to avoid slow and considerable computations according to the features of the mobile devices. Moreover, a cost function is used to sustain the chosen QoS during transition between networks, which is measured in terms of the bandwidth, BER, ABR, SNR and monetary cost. Simulation results presented that choosing the SA rules would minimise the cost function and the GA-SA algorithm could reduce the number of unnecessary handovers, and thereby avoid the 'Ping-Pong' effect.

Keywords: vertical handovers; simulated annealing; SA; genetic algorithms; GAs; heterogeneous wireless networks.

Reference to this paper should be made as follows: Goudarzi, S., Hassan, W.H., Soleymani, S.A. and Anisi, M.H. (2017) 'Hybridisation of genetic algorithm with simulated annealing for vertical-handover in heterogeneous wireless networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 24, Nos. 1/2, pp.4–21.

Biographical notes: Shidrokh Goudarzi is a PhD student in Electronic System Engineering at Communication Systems and Networks (CSN) Research Group, Department of Electronic Systems Engineering, Malaysia-Japan International Institute of Technology (MJIT), Universiti Teknologi Malaysia Kuala Lumpur (UTM). She is completed her Master's degree in Computer Science (Information Technology) at UTM. Her research and thesis include computational intelligence, mobile and computer networks, wireless communication. She is currently working on wireless networks, vertical handovers, and network selection strategies.

Wan Haslina Hassan is an Associate Professor of Computer Communications Computer Networks. She obtained her PhD from Universiti Teknologi Malaysia (UTM). Her research interests lie in the area of mobile computing, intelligent architectures and network security.

Sayed Ahmad Soleymani is a Research Assistant in Malaysia-Japan International Institute of Technology (MIIT). His current research interests are in wireless sensor network (WSN), wireless nano sensor network (WNSN), mobile ad hoc network (MANET), vehicular ad hoc network (VANET), mobile communication, visible light communication (VLC) and internet of nano things (IoNT).

Mohammad Hossein Anisi is a Senior Lecturer at the Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya. He obtained his PhD from Universiti Teknologi Malaysia (UTM) while being awarded as the best postgraduate student. He worked as Post-Doctoral Research Fellow at UTM and was a member of pervasive computing research group (PCRG), a research group under K-Economy Research Alliance in Malaysia. His research interests lie in the area of wireless sensor networks and their applications, mobile ad hoc networks and intelligent transportation systems. He has also collaborated actively with researchers in several other disciplines of computer science.

1 Introduction

In recent decades, we have witnessed the astonishing developments in wireless applications, devices and networks. In heterogeneous wireless networks (HWN), the vertical handover (VH) is an important factor in the provision of seamless mobility between varied network environments and an essential feature of all next-generation all-IP mobile network endeavours. Because no single access technology can offer ubiquitous coverage and continuously high quality of service (QoS), multimode mobile terminals will have to roam between the various accesses technologies to retain network connectivity and user satisfaction. These heterogeneous wireless access networks classically vary in terms of signal strength, coverage, data rate, latency and loss rate. VH occurs when a mobile terminal decides to switch between networks. The VH process comprises three main functions: system discovery, handover decision and handover execution. The VH decision is important for providing a low cost, highly available network environment that can achieve the desired QoS or quality of experience (QoE). During the handover decision stage, the mobile device decides to which network it should connect.

Many challenges are present in the VH decision phase during the handover process. One of the challenging problems during VH is the selection of an optimal network that maximises the end users' satisfaction. On the one hand, sometimes the terminal is moving rapidly along its route, so the algorithms that provide the VH decision phase must also be fast and able to give a solution near to real time in such dynamic scenarios (the mobility aspect is a key driver for future internet, within the field of mobility and ubiquitous access to networks). On the other hand, some decision algorithms handle many parameters that involve quite a lot of floating-point arithmetic calculations, and the computational effort increases with the required precision for the solutions, the number of QoS parameters or available networks discovered during the movement of the terminal. A high computational effort is in conflict with the low response time restriction, especially taking into account,

the low-performance processors embedded in many mobile devices.

Hence, there is a critical need to develop effective vertical handover decision algorithms (VHDA) that not only cultivate a hybrid VHD algorithm that utilises some form of intelligence for decision-making but also able to dynamically adapt to varying conditions in a timely manner given the rapid change in the wireless environment. Some VHDA including those involving computational intelligence techniques have been proposed, in the current research literature. According to Chandralekha and Behera (2010), the decision problem is formulated as a group of multi-objective optimisation problems and simulated using genetic algorithms (GAs). The simulation result showed that the number of handovers could be minimised if optimised network parameter values were taken into consideration.

Another category uses hybrid techniques where the algorithm was inspired by the Nash equilibrium (Zhang et al., 2013), known from the game theory. This study tried to seek a Pareto optimum solution under the Nash equilibrium of user utility and network provider utility, and used an elitist selection and individual migration of multi-objective GA to improve the decision efficiency. Jaraiz-Simon et al. (2014) proposed a new model to develop intelligent algorithms that avoided the slow and massive computations associated with direct search techniques, hence reducing the computation time. The authors developed an algorithm based on GAs in order to obtain a better performance than SEFISA and overcame the limitations exposed before, mainly the limitation of the number of generations, the appearance of compulsory stop criterion, the overflow of search space limits, stagnation of optimal solutions along generations and so on.

According to Lin et al. (1993), the hybridisation of GAs with existing algorithms showed better performance when compared to either the GAs or the existing algorithms used alone (Li and Wei, 2008). As reported by Giupponi et al. (2005) and Wilson et al. (2005), there were several existing multi-criteria-based algorithms with the aid of artificial intelligence tools such as fuzzy logic, neural networks and

GAs that suffered from scalability and modularity problems. They were unable to cope easily with the increased numbers of Radio Access Technologies (RATs) and criteria in the HWN. These algorithms took all the inputs from the different RATs at once to one fuzzy logic block, hence they suffered from scalability and complexity problems when more RATs or membership functions were added due to the exponential increase on the number of inference rules (Giupponi et al., 2005; Wilson et al., 2005).

In the context of future wireless networks, many analyses, studies and tutorials have been proposed in the literature (Ahmed et al., 2014; Movahedi et al., 2012; Paul et al., 2011; TalebiFard et al., 2010). These algorithms were classified into different groups based on the decision technique expended. Rakovic and Gavrilovska (2010) proposed a novel method for RAT selection, namely, the Hopfield neural network RAT selection mechanism (HRM), that utilised the Hopfield neural networks as a strong decision-making tool. A new approach using information about data rate, monetary cost and received signal strength as different parameters to make a handover decision has been reported by Çalhan and Çeken (2013a). The main weaknesses were exactly linked to the computation of the error function and Jacobian inversion for acquiring a matrix in which the dimensions were equal to the total of all the weights in the neural network. Hence, the requirement for memory was very high (Kordos and Duch, 2004; Lera and Pinzolas, 2002). Existing algorithms (Çalhan and Çeken, 2013a) considered service fee, received signal strength information (RSSI), user preference, etc. The proposed algorithm compared to the traditional RSSI-based algorithm, enhanced outcomes significantly for both user and network as a consequence of the proposed fuzzy-based handover systems (Çalhan and Çeken, 2013b). In terms of hybrid categories, Nan et al. (2011) proposed a PSO-FNN-based VHDA that could make a reasonable handover decision intelligently based on the study of network status. Liu and Jiang (2012) reported a novel VHDA built on fuzzy logic with the assistance of grey theory and dynamic weights adaptation. A neuro-fuzzy multi-parameter-based VHDA was proposed by Singhrova and Prakash (2012) where the results of performance evaluation, carried out by a handover quality indicator (used to quantify QoS) which is related upon the 'Ping-Pong' effect, ESA and throughput, proved that the proposed VHDA offered better QoS than existing VH methods. Pahlavan et al. (2000) was a good representation of the application of a fuzzy logic-based normalised quantitative decision algorithm and a differential prediction algorithm that had good accuracy.

Although the VH strategies mentioned above had their own advantages, they did not consider the complexity conditions of the network selection, and the allocation of lower computation cost function was also unreasonable. Clearly, the decision process needed to rely on a fast intelligent algorithm to perform an accurate decision, and to switch to the best candidate network within a very short period of time.

Hence, the objective of this study was to propose a novel, hybrid network selection optimisation algorithm that too advantage of two existing optimisation methods to meet the above-stated requirements. It was expected to provide more reliable results than a single run of the GA. The combination of SA rules with GA has three purposes:

- to serve as a validation algorithm for the outcome of the GA
- to detect calculation results that are unacceptable, i.e., higher than the global optimum
- to merge new generations of crossover and mutation steps.

In this algorithm, to create new populations, the rules of SA were used. The proposed method is a step towards future computer-based optimisation methods where huge uncertainties by the optimisation algorithm need to be avoided. To do this, a GA was combined with the simulated annealing (SA) optimisation. The proposed approach synthesises the capabilities of both search methods to avoid the drawbacks of GA. The combined algorithm differed from the already introduced evolutionary search methods in which it used a modified, SA optimisation method to validate the outcome of the GA. The proposed GA-SA-based VHDA could make reasonable handover decisions intelligently according to the study of network status. The cost function was also expected to be minimised by choosing the SA rules. Also, this algorithm is expected to reduce the number of unnecessary handovers, and hence, avoid the 'Ping-Pong' effect. The outline of this paper is as follows: first, the optimisation problem is described in Section 2. Then, the methodology of optimisation algorithms are explained in Section 3. The system model is described in Section 4. Section 5 provides the calculations and discussion of the results. Finally, Section 6 summarises the conclusions.

2 The optimisation problem

The efficient adjusting of the QoS weights to determine the best network among available ones is very significant for wireless networks. To find the best network, we need to know the merit of each available network. For this purpose, we need to design a function or metric to acquire the merit of this network. Firstly, a set of weights assigned to each one of the QoS parameters is used to measure the quality of the network, and they are based on the user's preferences and network characteristics. A general profile that includes any QoS parameter can have any weight assigned between 0 and 1. This measure is given by a function, which is called cost function. This function is evaluated in the VH decision-making stage. Thus, the optimisation problem consists of searching the best solution that, when applied to each network, returns the lowest cost to the network, which will then be chosen as the best solution for the VH decision stage. The GA component assigns a suitable weight

(w_1, w_2, \dots, w_i) for each initial decision according to the objective function that is specified by the operator according to the importance and sensitivities of the access network selection criteria to the different characteristics of a wireless heterogeneous environment.

2.1 Cost function

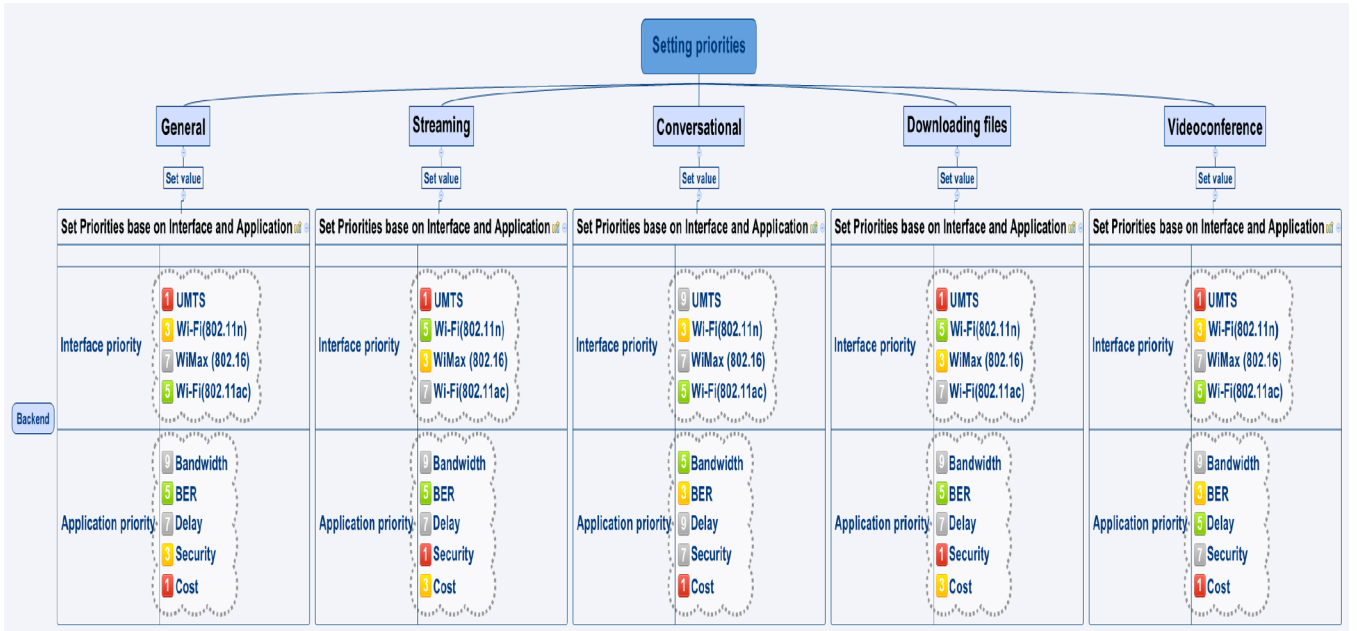
The VH cost function is a measurement of the cost utilised by handing off to a particular network. It is evaluated for each network that covers the service area of a user. The network choice that results in the lowest calculated value of the cost function is the network that would provide the most benefits to the user. As is known, different types of services require various combinations of reliability, latency and data rate. So, we consider service type as the main metric. Also, monetary cost is of major consideration to users, as different networks may employ different billing strategies that may affect the user's choice to handover. In terms of network conditions, network-related parameters such as available bandwidth and network latency may need to be considered for effective network usage. Use of network information in the choice to handover can also be useful for load balancing across different networks. Also, to guarantee the system performance, a variety of parameters can be employed in the handover decision to guarantee the system performance, for example, the bit error rate (BER). Mobile terminal conditions also include dynamic factors such as velocity, moving pattern, moving histories and location information.

For any of the service types, two sets of relative priorities are defined. These two sets are interface priorities

and application priorities where relative priorities among available interfaces in a device are interface priorities (Ahmed et al., 2006) and relative priorities among five types of services are application priorities. On the basis of the arrangement of the literal priority, scores between 1 and 9 are assigned automatically at the backend, where 9 denotes the most preferred and 1 denotes the least preferred priority. Setting priorities among available interfaces in the multimode terminal should be an important parameter from the user side because, for example, most users may like to give the cellular interface the highest priority especially for voice applications due to its almost ubiquitous coverage. QoS is another important parameter in order to ensure user satisfaction to the fullest extent. The user can establish preferences for the different QoS parameters depending on the service required. The experimentation is more realistic considering different sets for the user's preferences, where the QoS parameters that are more important have higher values of their weights. The whole process is illustrated in Figure 1.

At this stage, suitable limit values (upper and lower) for the five QoS parameters are mapped at the backend for each of the five service types. While fixing the limit values, it is important to note that high values are not always better for all the five QoS parameters. It is always preferable to have values as high as possible for bandwidth, but as low as possible for delay and BER. These values are based on the contexts such as QoS requirements of specific service types and interface capabilities.

Figure 1 Interface priorities and application priorities (see online version for colours)



Source: Singh and Mishra (2012), Sourangsu and Rahul (2013), Rakesh and Dalal (2010), Shaddad et al. (2014)

At the next stage, ranking of the available networks is performed based on the interface priorities scores and application priorities scores assigned at stage 1. Consider a set of candidate networks $S = \{s_1, \dots, s_N\}$ and a set of QoS factors $Q = \{q_1, \dots, q_M\}$, where N is the number of candidate networks and M is the number of QoS factors. Also, we consider that each QoS factor has its own weight and this weight shows the effect of the factor on network or user. So cost function for each network can be calculated using equation (1) where W_N is calculated by analytic hierarchy process (AHP) (Saaty, 1988). This process is chosen due to its ability to vary its weighting between each factor based on the network conditions and the user preferences.

$$C_N = W_{\text{Interface}} \times \sum_{j=1}^M q_j \times W_j \quad (1)$$

With the above definitions, the AHP method is described as follows.

In continue, the relative scores among the QoS scores set are calculated. Relative scores between any two particular scores are calculated using equation (2) where $R_{q_i q_j}$ is the relative score between parameters q_i and q_j , and S_{q_i} and S_{q_j} are their respective scores.

$$\begin{cases} R_{q_i q_j} = \left(1 - \frac{S_{q_i}}{S_{q_j}}\right) \times 10; & j > i \\ R_{q_j q_i} = \frac{1}{R_{q_i q_j}}; & j < i \\ R_{q_i q_i} = 1; & i = j \end{cases} \quad (2)$$

$X = \{X_{ij}\}$ is $M \times M$ matrix which X_{ij} represents the priority scores of each factor, is initialised as follows (3):

$$X = \begin{bmatrix} 1 & R_{q_1 q_2} & R_{q_1 q_3} & R_{q_1 q_4} & R_{q_1 q_5} \\ \frac{1}{R_{q_1 q_2}} & 1 & R_{q_2 q_3} & R_{q_2 q_4} & R_{q_2 q_5} \\ \frac{1}{R_{q_1 q_3}} & \frac{1}{R_{q_2 q_3}} & 1 & R_{q_3 q_4} & R_{q_3 q_5} \\ \frac{1}{R_{q_1 q_4}} & \frac{1}{R_{q_2 q_4}} & \frac{1}{R_{q_3 q_4}} & 1 & R_{q_4 q_5} \\ \frac{1}{R_{q_1 q_5}} & \frac{1}{R_{q_2 q_5}} & \frac{1}{R_{q_3 q_5}} & \frac{1}{R_{q_4 q_5}} & 1 \end{bmatrix} \quad (3)$$

Then when each element of the matrix X is divided with the sum of its column (4), the normalised relative weight is obtained.

$$X_{ij} = \frac{X_{ij}}{\sum_{i=1}^M X_{ij}} \quad (4)$$

The normalised matrix X is called w_{norm} which is shown in equation (5).

$$w_{\text{norm}} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \\ w_{51} & w_{52} & w_{53} & w_{54} & w_{55} \end{bmatrix} \quad (5)$$

Next, the average values of each row are calculated to give the priorities for each factor by equation (6) which is shown in equation (7).

$$\bar{w}_i = \frac{w_{i1} + w_{i2} + w_{i3} + w_{i4} + w_{i5}}{5} \quad (6)$$

The normalised vector equation (7) is called the priority vector. Since it is normalised, the sum of all the elements in priority vector is 1. The priority vector shows relative weights among them.

$$W_N = \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ \bar{w}_4 \\ \bar{w}_5 \end{bmatrix} \quad (7)$$

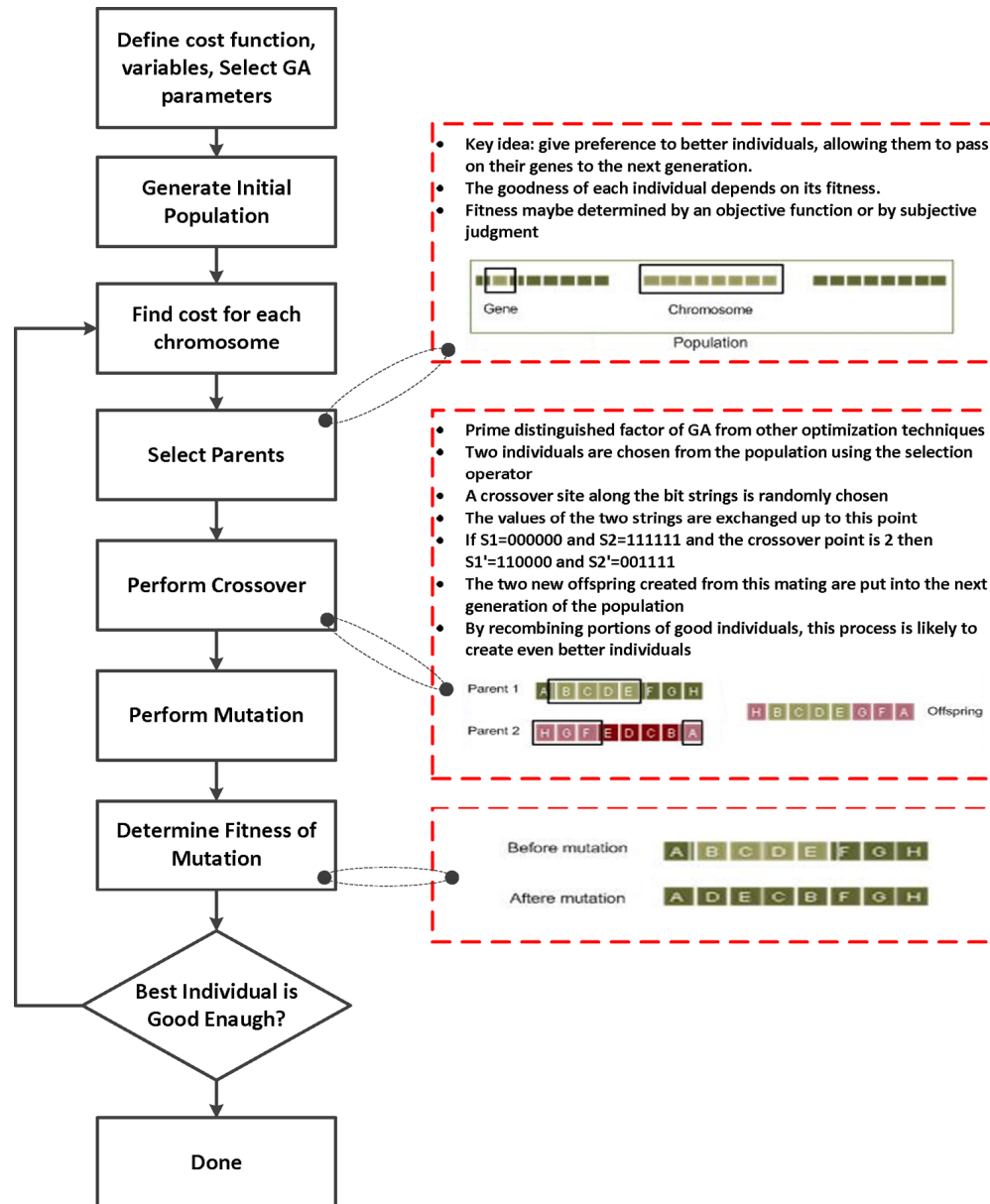
The set of QoS is shown by vector Q_N in equation (8):

$$Q_N = [B \ E \ A \ S \ C] \quad (8)$$

Which these QoS parameters are as follows: B = available bandwidth (Mbps), E = BER (dB), A = available ABR (bps), S = signal-to-noise ratio (SNR) (dB) and C = monetary cost (eur/MB). At the end, based on equation (1), the cost values of each user's requested services from the network can be computed.

3 Methods

In this paper, we consider a heterogeneous wireless environment with the co-existence of WLANs which are faster, high-bandwidth, lower-cost and short-distance access and Universal Mobile Telecommunications System (UMTS) with slower, higher-cost and long-range access networks that are always connected access networks. The algorithm can be programmed with small-sized codes, which is needed for embedded applications. It must be taken into consideration that, these days, there is a demand that mobile nodes provide the same performance at a lower price in which this can lead to reduced power consumption and in the end can improve the performance and mobility. We have provided a brief introduction for each algorithm and the operation is explained with a pseudo code (Appendix A) to facilitate the implementation and use of such algorithm by researchers and practitioners and figures as a visual aid. Finally, we indicate that our final scheme to solve the VH optimisation problem relies on the proposed algorithm, due to its better performance, which will be shown in the later sections.

Figure 2 Flow chart of the genetic algorithm (GA) (see online version for colours)

3.1 Genetic algorithm (GA)

GA search methods are rooted in the mechanisms of evolution and natural genetics. The interest in heuristic search algorithms with underpinnings in natural and physical processes began as early as the 1970s, when Holland (1992) first proposed GAs. In the field of artificial intelligence, GAs have emerged as powerful tools to solve the NP-hard problem. It is very simple to program GA. GAs are based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals using the following foundations:

- individuals in a population compete for resources and mates
- those most successful individuals in each 'competition' will produce more offspring than those individuals that perform poorly

- genes from 'good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent
- thus each successive generation will become more suited to their environment.

GAs are very popular but they suffer from three main problems (Li and Schonfeld, 2015). The first problem is mostly related to premature convergence in GA optimisation, whereby this problem is a result of high reliance on crossover. This can have an effect on the population by making the population more homogeneous and consequently, the search for the best solution is very slow in the mutation step. The second problem of GAs is related to the convergence of the optimal solution after finding a near-optimal solution. The third problem is the high level of memory usage by GAs. Since a GA must keep a large population of solutions in its memory, this results

in increased memory requirements when the problem dimensions increases. The flowchart of the GA is presented in Figure 2.

In this study, we show the hybridisation of GAs-SA to improve the performance of a GA for wireless network selection optimisation. The idea of this hybrid GSAVHO algorithm is to use the SA rules to improve the efficiency of the GAs mutation and crossover. Section 4 presents the GSAVHO hybrid algorithm with its detailed algorithmic steps.

3.2 Simulated annealing

Annealing is the cooling process all through which a low energy state is achieved in a solid. In the annealing method, the solid is first heated and melted at a high temperature, in which position all particles move at random. The solid is then slowly cooled until the particles place themselves in their lowest energy state, which is the ground condition. To avoid the resulting crystals from showing defects or even lacking all crystalline order, the cooling schedule should be sufficiently slow (Van Laarhoven and Aarts, 1987). In the SA technique, a new solution is created through a neighbourhood structure and a generation method. The neighbourhood structure states a set of solutions, S_s , which is ‘close’ to the present solution, s , in some sense. The generation method is a means of choosing a new solution from the neighbourhood S_s of the solutions. The operation of SA is shown in the following steps:

Step 1: Initialisation: Initialise the iteration count $k = 0$, and the temperature T_0 to be sufficiently high.

Step 2: Repeat for each temperature T^k : Execute Steps 2–4 until an equilibrium criterion is satisfied.

Step 3: Neighbourhood solution: Generate a trial solution x^{k+1} in the neighbourhood of the current solution x^k .

Step 4: Acceptance criterion. Let $\Delta = f(x^{k+1}) - f(x^k)$ and r is a random number uniformly distributed over $[0, 1]$. If $\Delta < 0$ (i.e., the solution is improved), the trial solution is accepted.

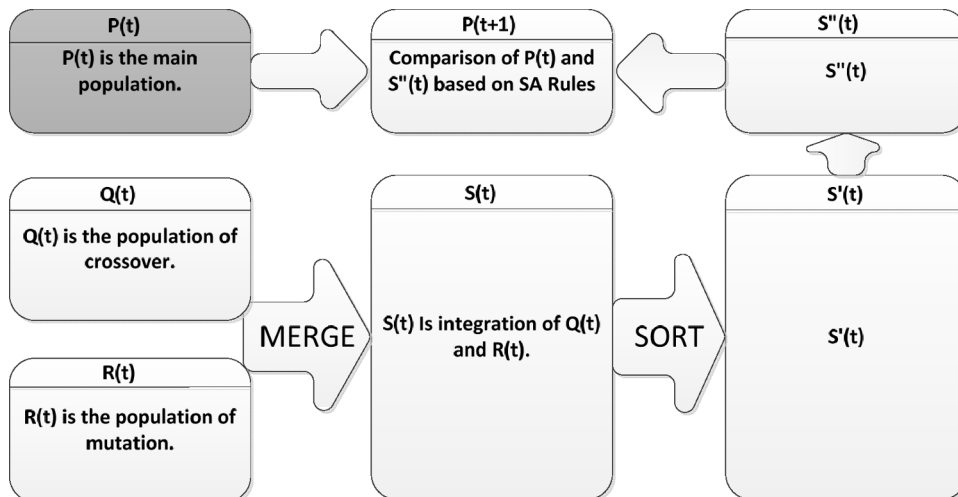
Otherwise, the trial solution is accepted with the probability $\exp(-\Delta/T^k) > r$.

3.3 Hybrid of Genetic and simulated annealing algorithms

In this work, we have developed an algorithm based on the hybridisation of GAs and SA algorithms in order to obtain better performance and overcome the limitations of GA which has been outlined in Section 3.2. The reasons behind the choice of the GA and SA hybrid are because they have proven to be efficient and strong in search processes, and have produced near optimal solutions, making them suitable for solving large combinatorial optimisation problems. Unlike the other algorithms, GAs are biologically inspired algorithms for conducting random search and optimisation guided by the principles of natural evolution and genetics. As mentioned before, GA has a strong global search ability in solving the aforementioned problems, but also has limitations such as a premature and slow convergence rate, local optimum and ignoring cooperation between populations as well as weak local search ability (Tsai et al., 2011). On the other hand, SA has strong local search ability and no premature problems. Therefore, the hybridisation of GA and SA can overcome the limits of each of the two methods, bringing into play their respective advantages and improve the solving efficiency.

To hybridise GA with SA, we have used SA rules to combine and compare the populations of GA. Figure 3 shows the workflow of the GA-SA algorithm. In Figure 3, $P(t)$ is the main population, $Q(t)$ is the population of crossover and $R(t)$ is the population of mutation. The integration of $Q(t)$ and $R(t)$ products $S(t)$. In the next step, $S(t)$ must be sorted and the sorted population is called $S'(t)$. In $S'(t)$, top population are better solutions. In $P(t)$, the number of elements are $\text{npop}()$, then we have selected the same elements $\text{npop}()$ of the top populations of $S'(t)$ and labelled them $S''(t)$. Then, the comparison of $P(t)$ and $S''(t)$ based on SA r rules is needed. At this stage, the population of a new generation has been created.

Figure 3 Workflow of (GSAVHO)



GAs essentially uses three genetic machinists: selection, crossover and mutation, which simulate the procedure of natural selection and provide them their powerful search ability. Subsequently, the main operators have been explained:

Selection: Selection is the process of choosing potentially better individuals to form a mating pool. There are many selection schemes such as Roulette Wheel selection, Boltzmann selection, Rank selection, Tournament selection and Elitist selection. Elitist selection is expanded in this paper and the latter is used to prevent the best solution disappearing in the next generation. The idea of Elitist selection is that some of the best performing individuals are kept and used to replace the same number of the worst individuals in the next generation. That is, the offspring will progress to be at least as good as their parents.

Crossover: The crossover operator simulates the development of gene recombination, which aims to pass the best genes on to the next generation. Usual crossover methods comprise one-point crossover and two-point crossover. According to Tsai et al. (2011), two-point crossover is greater to one-point crossover; so two-point crossover is used in this study. The idea of two-point crossover is that two crossover sites are chosen randomly and then the genes amongst the two sites in the two parent networks are switched.

Mutation: Mutation is an essential genetic operator that raises the variety of the population, even though the probability of mutation is minimal. The idea of the mutation operator is that with a certain probability and in a way that is specific to each individual's every gene, the gene is exchanged with a new gene not contained in the individual. Notice that the list of the network must be sorted and it should be shown that there are not any duplicate networks in the list after the operation.

Creation of new generation by SA rules: On the basis of the literature (Dahal and Chakpitak, 2007; Zameer et al., 2014), some strategies are proposed to merge the population in GAs. In one strategy called predefined shares, each set has the same portion and $n(\text{pop})$ is a combination of $P(t) + Q(t) + R(t)$ in which $P(t)$, $Q(t)$ and $R(t)$ are the main population, crossover population and mutation population, respectively. Also, in another strategy, called predefined probabilities; each set has the same probabilities for integration. In this approach, the total probability of three sets is equal to 1. In our strategy, the combination and comparison amongst the populations of parents and children are based on SA rules.

The whole population must create at a fixed temperature. With this purpose in mind, we defined that in the external iteration, the temperature should be decreased. In other words, all the processes involved in the creation of a new population including crossover and mutation were moved into the internal iteration, which was named

MaxSubIt () for showing the maximum number of sub-iterations. We found that the number of parameters increased because the parameters of GA and SA should be defined, for instance, the initial temperature (T_0) and reduction rate of temperature was shown with $\alpha = 0.99$, and were related to SA.

Pseudo-Code: Creation of new generation by SA rules

```

1: Initialise Temperature
2: GA Main Loop
3: For it = 1: MaxIt
4:   For subit = 1: MaxSubIt
5:     Perform Crossover
6:     Select First Parent ( $w_1$ ) according to  $C$  values
7:     Select Second Parent ( $w_2$ ) according to  $C$  values
8:     Perform Crossover
9:     Evaluate Offspring
10:    End (MaxSubIt)
11:    Perform Mutation
12:    For  $i = 1:nm$ 
13:      Select Parent ( $w_i$ ) according to  $C$  values
14:      Perform Mutation
15:      Evaluate Mutants
16:    End ( $nm$ )
17: Merge Offspring's Population
18: Sort NEWPOP
19: Compare using SA Rule

```

Also, the number of function evaluations is an effective parameter for the comparison between algorithms. The efficiency is determined by recording the number of function evaluations N_{fe} for each optimisation algorithm, where a low value of N_{fe} means a higher efficiency. This paper has used the number of function evaluations (N_{fe}) to measure the performance of algorithms and defines the optimal model to be the one that consumes the fewest (N_{fe}) to solve the problem. For GSAVHO and other algorithms, the numbers of function evaluations were calculated by equation (9) (Dede and Ayvaz, 2015):

$$N_{fe} = 2 \times G_n \times P_n + P_n \quad (9)$$

where G_n is number of generations where the best solution was obtained and P_n is the number of populations.

4 System model

As was discussed in Section 2.1, the cost function can measure the quality of each network to establish the connections. This metric is able to give the level of merit of each available network in a fixed time. This measurement is computed from a set of weights assigned to each QoS parameter by the AHP method (Ahmed et al., 2006), where this allocation can be based on the user's preferences and network conditions. This calculation is shown by the cost function (C), which is computed in the VH decision phase.

Algorithm 1 GSAVHO pseudo-code

Parameters Definition: Initial Temperature, Temp. Reduction Rate, NN, RSS, Set QoS parameters, NQ, nm= Number of Mutants, nPop=Population Size, BT = Bandwidth threshold, RT = RSS threshold, ...

```

1: Calculate the Sum of RSS if  $R_n \geq \text{SUM}/n$  continue; else no handover
2: Initialise Population
3: For each node  $n_i$  in the network:
4: Set strategy and percentage for mutation and crossover
5: Evaluate the cost function of each individual, where:
6: Calculate priority vector to show relative weights among them by AHP
7: If  $B_i > B_T$  and  $R_i > R_T$ 
8: Calculate  $D_{\text{Bandwidth}}$  and  $D_{\text{RSS}}$ :
9: If  $D_{\text{Bandwidth}} > 0$  and  $D_{\text{RSS}} > 0$ 
10: Import the parameters into the GA
11: Else
12: No handover happens
13: Sort Population
14: Initialize Temperature
15: // GA Main Loop
16: For it = 1: MaxIt
17: // Perform Crossover
18:   For subit=1: MaxSubIt
19:     Select First Parent ( $w_1$ ) according to  $C$  values
20:     Select Second Parent ( $w_2$ ) according to  $C$  values
21:     Perform Crossover
22:     Evaluate Offspring
23:   End (MaxSubIt)
24: // Perform Mutation
25:   For  $i = 1$ : nm
26:     Select Parent ( $w_i$ ) according to  $C$  values
27:     Perform Mutation
28:     Evaluate Mutants
29:   End (nm)
30: Merge Offspring's Population
31: Sort NEWPOP
32: Compare using SA Rule
33: For  $i = 1$ : nPop
34:   if new pop( $i$ ).Cost  $\leq$  pop( $i$ ).Cost
35:     pop ( $i$ ) = new pop( $i$ );
36:   else
37:     DELTA = (new pop( $i$ ).Cost - pop( $i$ ).Cost) / pop( $i$ ).Cost;
38:     P = exp(-DELTA/T);
39:     if rand  $\leq$  P
40:       pop ( $i$ ) = new pop( $i$ );
41:   End (if)
42: Update Best Solution Ever Found
43:   End (nPop)
44:   End (MaxIt)
45: Store Best Cost
46: Temperature. Reduction
47: Update Best Solution Ever Found

```

We have designed an algorithm based on the hybridisation of GAs and SA algorithms called genetic simulated annealing vertical handover (GSAVHO). In our algorithm, the individuals of a population are vectors of QoS weights, which are possible solutions of the optimisation problem. The population is initially generated with an even number of

individuals. Each individual is identified by pop_{ij} where i is the index matching with the number of individuals in the population and j is an index that corresponds with a QoS parameter. Also, each individual has a set of cost values, because there are different available networks. The initial population is obtained using random selection. This way,

GSAVHO can start from an initial population where the individuals represent all the areas of the space of solutions, allowing a high degree of diversity for the successive evolutions of the population. The GSAVHO operation is shown in Algorithm 1. Starting from a determined dataset, user's profile, number of networks and QoS parameters, precision level, population size, initial temperature ($T_0 = 10$), temperature, reduction rate ($\alpha = 0.99$), number of parents (offsprings) ($nc = 2 * \text{round}(pc * nPop/2)$), number of mutants ($nm = \text{round}(pm * nPop)$), number of iterations ($\text{Max It} = 1000$), number of sub-iterations of SA ($\text{MaxSubIt} = 10$) (line #1), in (line #2) to avoid 'Ping-Pong' effect, we need to know whether handover is necessary or not. For this purpose, we focus on the SUM of the received signal strength in n networks in a time period T_s based on equation (10).

$$\text{SUM} = \sum_{T=0}^{T=s} R_T \quad (10)$$

where s is the number of times that the received signal strength is higher than the trigger threshold of the handover R in a time period; $0 \leq T \leq s$;

$$\begin{cases} R_n < \text{SUM}/n, \text{ no handoff} \\ R_n \geq \text{SUM}/n, \text{ execute handover} \end{cases} \quad (11)$$

By using the above-handover triggering and after checking R_n for n th network and comparing with the average RSS of networks in the time period s , the 'Ping-Pong' effect can be prevented. At the end of this calculation, the amount of SUM must be reset. The population is initialised (line #3), to continue, crossover percentage (pc) and mutation percentage (pm) are fixed at this stage (line #4), after sorting the population, (line #13). We have defined the pre-decision method to filter the requested services of the users before sending into the GA unit. B_i and R_i are used for the pre-decision stage where they are the available bandwidth (B) and received signal strength (R) of the candidate networks. They are calculated by following equations (12) and (13).

$$D_{\text{Bandwidth}} = B_2 - B_1 \quad (12)$$

$$D_{\text{RSS}} = R_2 - R_1 \quad (13)$$

where $D_{\text{Bandwidth}}$ is the diversity of available bandwidth between networks and D_{RSS} is the diversity of received signal strength between the networks. On the basis of the following conditions, equation (14):

$$\begin{cases} \text{if } D_{\text{Bandwidth}} > 0 \text{ and} \\ D_{\text{RSS}} > 0 \text{ Import QoS parameters to decision unit} \\ \text{if } D_{\text{Bandwidth}} \leq 0 \text{ and } D_{\text{RSS}} \leq 0 \text{ No handoff} \end{cases} \quad (14)$$

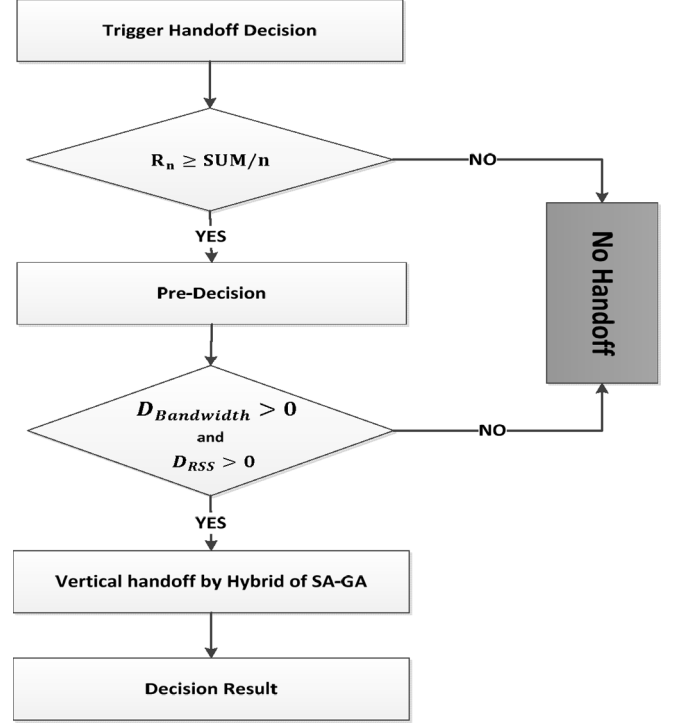
In continuation, (line #18 till end), we show the related stages of our optimisation algorithm that includes the crossover, mutation, SA rules, and so on.

In line 4, the related strategies for mutation and crossover are completely explained in Section 3.3. To better

understanding, we showed the main codes of our algorithm in Appendix A.

The following flowchart presents the main steps of our heuristic (Figure 4).

Figure 4 Flowchart of (GSAVHO)



5 Computational experiments

5.1 Evaluation of proposed VHO scheme

To demonstrate the performance of the VHO optimisation in HWN, we considered two main experiments: with GSAVHO optimisation and with GAVHO optimisation. This section focuses on the efficiency of GA-SA as tested against the GA algorithm with different QoS parameters. The networks characterised by the values of the following QoS parameters, are listed in Table 1. In this table, we introduce QoS parameters of proposed network selection algorithm. The heterogeneous environment may consist of a number of different wireless networks. For simplicity, without loss-of-generality, we consider a HWN consisting of UMTS, Wi-Fi (802.11n), WiMax (802.16) and Wi-Fi (802.11ac).

Table 1 QoS parameters

Net	Bandwidth (Mbps)	BER (dB)	Delay (ms)	Security level	Monetary cost (eur/MB)
UMTS	14.4	0.001	19	8	0.9
Wi-Fi (802.11n)	30	0.01	20	2	0.2
WiMax (802.16)	75	0.01	30	3	0.3
Wi-Fi (802.11ac)	72	0.01	15	3	0.2

Selection metrics of cellular networks in the heterogeneous environment are discussed as follows. Handover delay refers to the duration between the initiation and completion of the handover process. Handover delay is related to the complexity of the VHD process, and reduction of the handover delay is especially important for delay-sensitive voice or multimedia sessions. If D_N consider as the normalised handoff delay in handoff transition region then for the real time service the normalised handoff delay is as follows (Park et al. 2003):

$$D_N = N\Delta / T_t \quad (15)$$

where Δ is the handoff completion time and T_t is the region extending from the point at which the power falls below threshold for the first time to the point at which power falls below threshold permanently.

For delay-tolerant applications, approximate value of the residual bandwidth of the WLAN is evaluated by the following formula (16) (Yan et al., 2010):

$$\begin{aligned} &\text{Residual bandwidth} \\ &= \text{throughput} \times (1 - \alpha \times \text{channel_utilisation}) \\ &\quad \times 1 - \text{packet_loss_rate} \end{aligned} \quad (16)$$

where throughput is the throughput that can be shared among mobile terminals in the WLAN, channel_utilisation is the percentage of time the access point senses the medium is busy using the carrier sense mechanism, α is a factor that reflects IEEE 802.11 MAC overhead (it is set to 1.25 in Lee et al. (2005)), and packet_loss_rate is the portion of transmitted medium access control (MAC) protocol data units (MPDUs) that require retransmission, or are discarded as undeliverable. The values of channel_utilisation and packet_loss_rate are obtained from the information in the beacon frame carrying the QoS basic service set (QBSS) load sent by an access point, as defined in IEEE 802.11e (Mangold et al., 2003).

Another metric for network selection algorithm is BER. BER is the probability of BER of cellular network present in heterogeneous environment. Probability of BER is calculated by using equation (17), which depends upon signal-to-noise. Probability of BER defined in equation (16) is for non-coherent FSK in a white Gaussian noise (Reza, 2012).

$$\text{BER}_i(\gamma_i) = \frac{1}{2} e^{-\gamma_i/2} \quad (17)$$

where γ_i is the signal to noise ratio, $i = 1, 2, 3$ and 4 for UMTS, Wi-Fi (802.11n), WiMax (802.16) and Wi-Fi (802.11ac) cell environments, respectively.

On the basis of the obtained results, we can see that in GAVHO in first iterations of such as iteration between 2 and 15, network 3 is selected as best solution then network 1 is selected as best network. These values show a level of the network quality by means of cost function. Figures 5 and 6 show results based on different QoS parameters, number of population equal 8 and iteration 100.

Several experiments were performed to find the best wireless network optimised with the hybridisation of GA and SA. The main idea was to achieve a comparison of the results obtained with GAVHO optimisation and also hybridisation of GA-SA optimisation, for the best network selection during VH. As it is mentioned before, each QoS factor includes bandwidth, BER, delay, security level and monetary cost within own weight affect on network or user. The values of the QoS parameters characterise a given network and each parameter is combined with own weight, also each weight has obtained by AHP method. We selected the most main parameters for VH; we then defined and simulated a basic scenario and finally by analysing the selected parameters, measured and calculated RSS, ABR, SNR and throughput. We will explain the details of the scenario in the following.

Figure 5 Results of different parameters found by of GAVHO with iteration = 100 and network's number = 8 (see online version for colours)

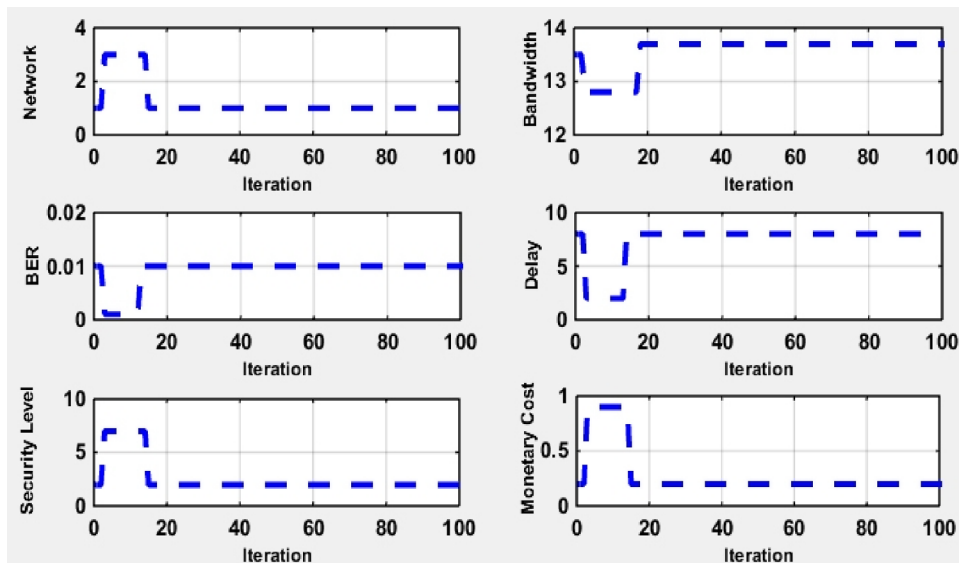


Figure 6 Results of different parameters found by of GSAVHO with iteration = 100 and network's number = 8 (see online version for colours)

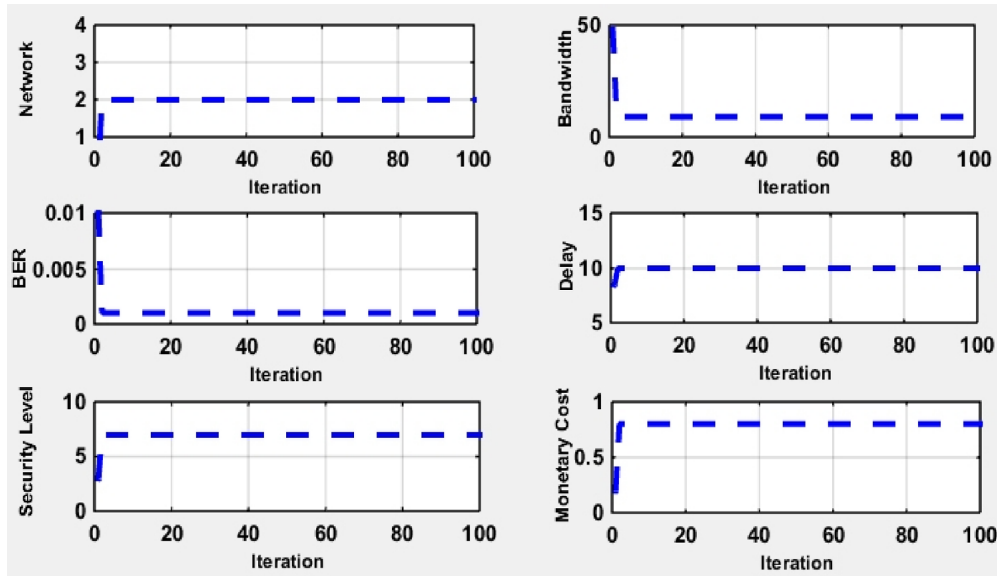
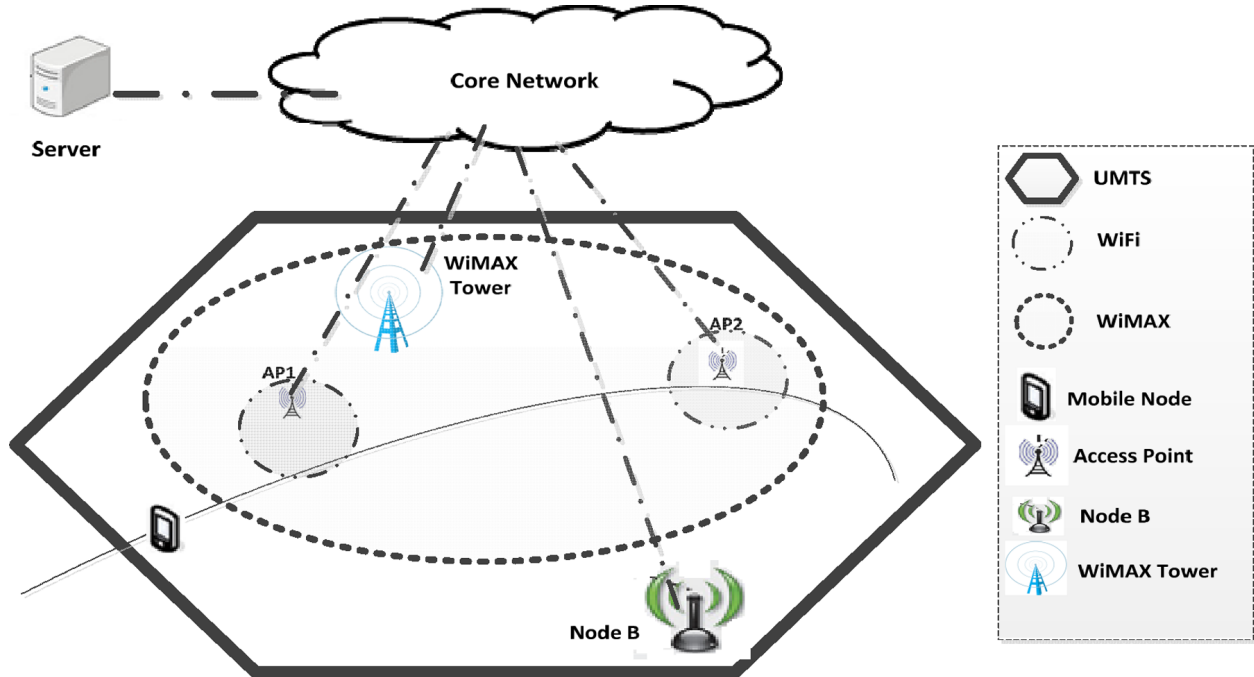


Figure 7 Simulation scenario (see online version for colours)



On the GSAVHO algorithm, we have performed simulations to show the feasibility of proposed scheme by the simulation software QualNet 7.0. Our experiment is for the case of a VH between a 3G network and WLANs. The scenario simulated in QualNet composed of an UMTS BS and IEEE 802.11 APs. The WLAN is based on the IEEE 802.11b standard, and the physical data rate is 11 Mbps. The capacity of a 3G UMTS is 384 kbps. All links except wireless links each have a capacity of 100 Mbps. Ad-hoc on-demand distance vector (AODV) protocol was used as a reactive routing protocol (Royer and Perkins, 1999). This protocol offers quick convergence when the ad hoc

network topology changes (typically, when a node moves in the network).

As shown in Figure 7, the mobile node can be at a given time in the coverage area of an UMTS. However, owing to mobility, it can move into the regions covered by more than one access network, i.e., simultaneously within the coverage areas of, for example, an UMTS BS and an IEEE 802.11 AP. Multiple IEEE 802.11 WLAN coverage areas are usually contained within an UMTS coverage area. Since the WLAN1 has lower coverage range, when the mobile user is moving out of a WLAN1 area, existence of an accurate and timely handoff decision to maintain the connectivity before

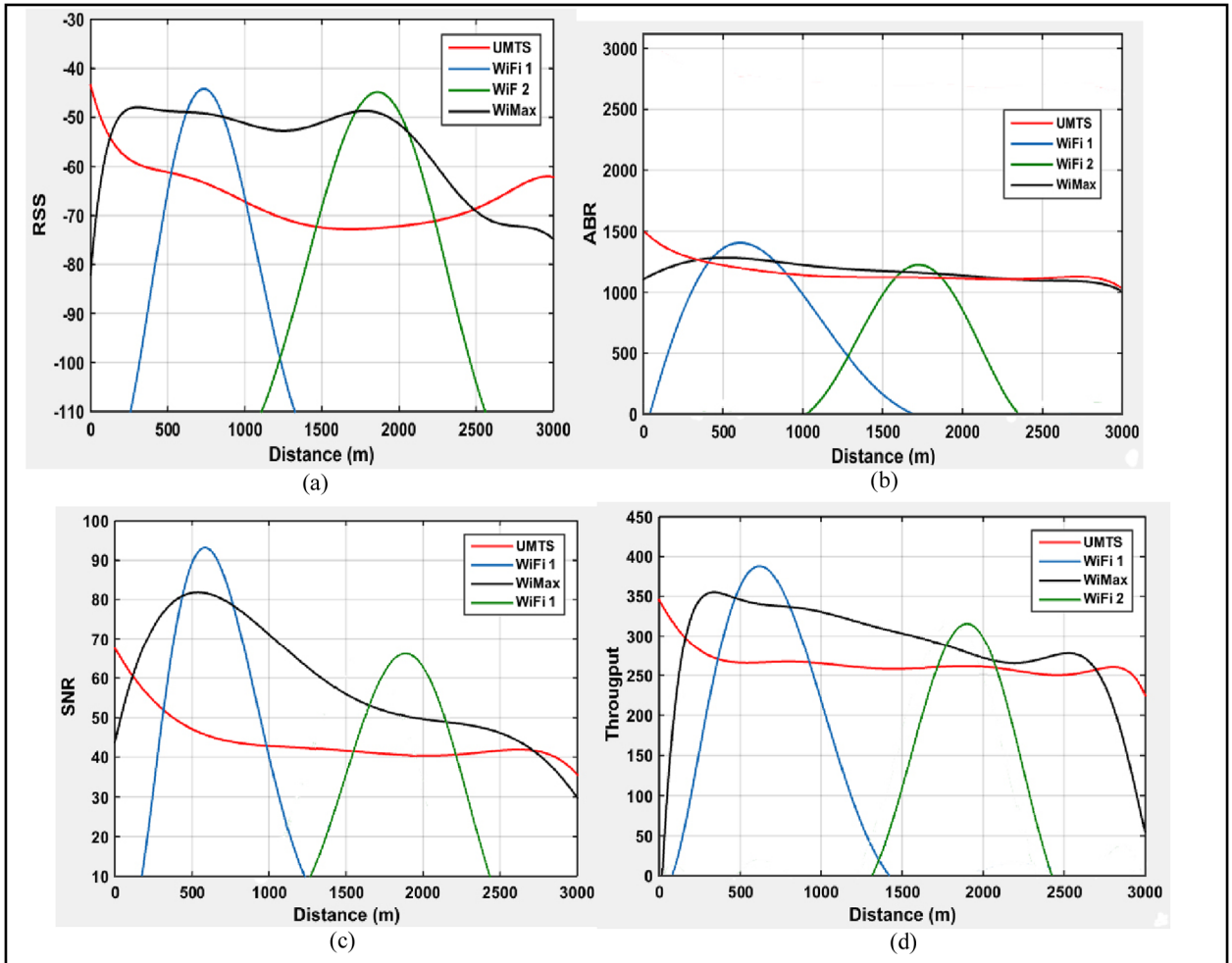
the loss of the WLAN access is necessary. In continue the user move into the regions covered by a UMTS network and then user would like to move into a WLAN2 area because user wants to achieve a higher QoS at a lowest cost. So user changes the connection to the WLAN2. The mobile node associated with UMTS or WLANs monitors and measures the $D_{\text{Bandwidth}}$ is the diversity of available bandwidth between networks and D_{RSS} is the diversity of received signal strength between the networks of nearby WLANs/UMTS too check whether or not a access network with high data rate is offered.

The performance of the proposed algorithm has been assessed in a scenario when the Mobile Node moves with a constant speed along a straight line path from the area covered by WLAN1 to the one covered by UMTS then

roams to the area which covered by WLAN2. Clearly with the increase of distance the average of RSS, ABR and SNR will be reduced and also BER will be increased. By calculating the cost value based on number of iterations, the optimal network can be selected as presented in Figure 8.

Where γ_i is the signal-to-noise ratio, $i = 1, 2$ and 3 for WLANs and UMTS (3G) cell environments, respectively. BER: For non-coherent FSK in a white Gaussian noise channel, the probability of a bit error is given by: $\text{BER}_i(\gamma_i) = (1/2)e^{\gamma_i/2}$, (Reza, 2012). Available bit rate (ABR): $\text{ABR}_i = \text{bandwidth} \times \log_2(1 + \text{SNR}_i)$ $i = 1, 2, 3, 4$ (Pfister et al., 2001). Signal-to-noise ratio (SNR): $\text{SNR}_i = E_b/N_o$ (Reza, 2012), E_b is the received energy per bit and is noise power of the channel.

Figure 8 (a) Average RSS vs. distance; (b) available bit rate (ABR) vs. distance; (c) SNR vs. distance and (d) throughput vs. distance in cellular heterogeneous environment of WLANs and UMTS (see online version for colours)



5.2 Comparison with SEFISA and FMADM

We have performed performance comparisons between our algorithm GSAVHO and other algorithms structured in the literature; namely SEFISA and GAVH (Jaraiz-Simon et al., 2014) and FMADM (Nkansah-Gyekye and Agbinya, 2007). In the study by Jaraiz-Simon et al. (2014), two algorithms

were designed to decide the best network to establish connection in a VH process; SEFISA is based on the SA algorithm and GAVH was developed based on GAs. In addition, the FMADM (Nkansah-Gyekye and Agbinya, 2007) was a fuzzy multiple attribute decision making algorithm that selects a suitable wireless access network during the VH process.

We selected these proposals because this study enabled us to compare our heuristic with similar datasets. To make fair comparisons, our heuristics have five similar QoS parameters in dataset1 and 9 QoS parameters in dataset2. We have considered the following datasets for the weight ranges of QoS parameters, which are shown in Tables 2 and 3. The datasets consist of several networks characterised by the following QoS parameters: B = bandwidth (kbps), E = BER (dB), D = delay (ms), S = (dB), C = cost (eur/MB), L = network latency (ms), J = jitter (ms), R = burst error, A = average retransmissions/packet, P = packet loss (%), G = received signal strength indication RSSI (dBm), N = network coverage area (km), T = reliability, W = battery power requirement (W) and V = mobile terminal velocity (m/s). Among these algorithms, the one based on the hybridisation of GA and SA (GSAVHO) demonstrated the best performance, in terms of precision and cost function.

Table 3 Simulation results corresponding to the best costs for FMADM and GSAVHO for number of networks = 3 and number of QoS = 9 (dataset 2)

		N	L	T	B	W	V	S	G	C
NN=1		1.82	0.54	2.46	1.10	0.44	2.46	2.46	2.46	0.54
NN=2		0.04	0.09	0.04	0.45	0.04	0.04	0.07	0.04	0.21
NN=3		0.46	0.02	0.02	0.46	0.02	0.02	0.02	0.14	0.28
Best cost	FMADM					0.799				
	GSAVHO					4.6861e-06				
Standard deviation	FMADM					N/A				
	GSAVHO					0.0044				

6 Conclusion

Our goal was to achieve an optimised GSAVHO with a small cost function, and for this reason the SA rules were not changed and we proposed an optimisation for GA where only combination and comparison among populations of parents and children were based on the SA rules. The proposed approach performed well in several test problems in terms of the number of cost function evaluations required, the quality of the solutions found, the average of the function evaluation numbers and the average of cost functions. Several experiments were performed for the optimisation of the VHDA-making algorithms with GA and the hybridisation of GA-SA in HWN. The results obtained were analysed and compared, and it was found that the GSAVHO had a lower cost function when compared to the GAVHO. Simulation results indicated that our proposed VHDA was able to minimise the cost function, reduce the number of unnecessary handovers, avoid the 'Ping-Pong' effect and select the best access network that was optimised to network conditions, QoS requirements, mobile terminal conditions, user preferences and service cost.

Table 2 Simulation results corresponding to the best costs for GAVH, SEFISA, GA and GSAVHO for number of networks = 4, number of QoS = 5 (dataset 1)

		B	E	D	S	C
NN = 1		0.44	0.02	0.5	0.02	0.02
NN = 2		0.54	0.36	0.02	0.07	0.01
NN = 3		0.48	0.50	0.04	0.02	0.02
NN = 4		0.34	0.68	0.02	0.04	0.02
Best cost	SEFISA			1.89		
	GAVH			5.77		
	GSAVHO			4.7248e-6		
Standard deviation	SEFISA			N/A		
	GAVH			N/A		
	GSAVHO			0.0430		

References

- Ahmed, A., Boulahia, L.M. and Gaïti, D. (2014) 'Enabling vertical handover decisions in heterogeneous wireless networks: a state-of-the-art and a classification', *Communications Surveys & Tutorials*, IEEE, Vol. 16, No. 2, pp.776–811.
- Ahmed, T., Kyamakya, K. and Ludwig, M. (2006) 'A context-aware vertical handover decision algorithm for multimode mobile terminals and its performance', *Proceedings of IEEE/ACM Euro American Conference on Telematics and Information Systems (EATIS '06)*, Santa Marta, Colombia, February, pp.19–28.
- Çalhan, A. and Çeken, C. (2013a) 'Artificial neural network based vertical handoff algorithm for reducing handoff latency', *Wireless personal communications*, Vol. 71, No. 4, pp.2399–2415.
- Çalhan, A. and Çeken, C. (2013b) 'Case study on handoff strategies for wireless overlay networks', *Computer Standards & Interfaces*, Vol. 35, No. 1, pp.170–178.
- Chandrasekha, P.B. and Behera, P.K. (2010) 'Minimization of number of handoff using genetic algorithm in heterogeneous wireless networks', *International Journal of Latest Trends in Computing*, Vol. 1, No. 2, pp.24–28.
- Dahal, K.P. and Chakpitak, N. (2007) 'Generator maintenance scheduling in power systems using meta heuristic-based hybrid approaches', *Electric Power Systems Research*, Vol. 77, No. 7, pp.771–779.

- Dede, T. and Ayvaz, Y. (2015) 'Combined size and shape optimization of structures with a new meta-heuristic algorithm', *Applied Soft Computing*, Vol. 28, pp.250–258.
- Giupponi, L., Agusti, R., Pérez-Romero, J. and Sallent, O. (2005) 'A novel joint radio resource management approach with reinforcement learning mechanisms', *24th IEEE International on Performance, Computing, and Communications Conference, 2005. IPCCC 2005*, IEEE, April, pp.621–626.
- Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.
- Jaraiz-Simon, M.D, Gomez-Pulido, J.A. and Vega-Rodriguez, M.A. (2014) 'Embedded intelligence for fast QoS-based vertical handoff in heterogeneous wireless access networks', *Pervasive and Mobile Computing*, Vol. 19, pp.141–155.
- Kordos, M. and Duch, W. (2004) 'Variable step search algorithm for MLP training', *8th IASTED Int. Conf. on Artificial Intelligence and Soft Computing*, Marbella, Spain, September, pp.215–221.
- Lee, C.W., Chen, L.M., Chen, M.C. and Sun, Y.S. (2005) 'A framework of handoffs in wireless overlay networks based on mobile IPv6', *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 11, pp.2118–2128.
- Lera, G. and Pinzolas, M. (2002) 'Neighborhood based Levenberg-Marquardt algorithm for neural network training', *IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp.1200–1203.
- Li, X-G. and Wei, X. (2008) 'An improved genetic algorithm-simulated annealing hybrid algorithm for the optimization of multiple reservoirs', *Water Resources Management*, Vol. 22, No. 8, pp.1031–1049.
- Li, Z. and Schonfeld, P. (2015) 'Hybrid simulated annealing and genetic algorithm for optimizing arterial signal timings under oversaturated traffic conditions', *Journal of Advanced Transportation*, Vol. 49, No. 1, pp.153–170.
- Lin, F-T., Kao, C-Y. and Hsu, C-C. (1993) 'Applying the genetic approach to simulated annealing in solving some NP-hard problems', *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 6, pp.1752–1767.
- Liu, X. and Jiang, L-G. (2012) 'A novel vertical handoff algorithm based on fuzzy logic in aid of grey prediction theory in wireless heterogeneous networks', *Journal of Shanghai Jiaotong University (Science)*, Vol. 17, pp.25–30.
- Mangold, S., Choi, S., Hiertz, G.R., Klein, O. and Walke, B. (2003) 'Analysis of IEEE 802.11 e for QoS support in wireless LANs', *Wireless Communications, IEEE*, Vol. 10, No. 6, pp.40–50.
- Movahedi, Z., Ayari, M., Langar, R. and Pujolle, G. (2012) 'A survey of autonomic network architectures and evaluation criteria', *Communications Surveys & Tutorials, IEEE*, Vol. 14, No. 2, pp.464–490.
- Nan, W., Wenxiao, S., Shaoshuai, F. and Shuxiang, L. (2011) 'PSO-FNN-based vertical handoff decision algorithm in heterogeneous wireless networks', *Procedia Environmental Sciences*, Vol. 11, pp.55–62.
- Nkansah-Gyekye, Y. and Agbinya, J. (2007) 'Vertical handoff decision algorithm for UMTS-WLAN', *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, 2007. AusWireless 2007*, IEEE, August, pp.37–37.
- Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Makela, J., Pichna, R. and Vallström, J. (2000) 'Handoff in hybrid mobile data networks', *Personal Communications, IEEE*, Vol. 7, No. 2, pp.34–47.
- Park, H.S., Yoon, S.H., Kim, T.H., Park, J.S., Do, M.S. and Lee, J.Y. (2003) 'Vertical Hando. Procedure and algorithm between IEEE802. 11 WLAN and CDMA cellular network', *Mobile Communications*, Springer Berlin Heidelberg, pp.103–112.
- Paul, S., Pan, J. and Jain, R. (2011) 'Architectures for the future networks and the next generation Internet: a survey', *Computer Communications*, Vol. 34, No. 1, pp.2–42.
- Pfister, H.D., Soriaga, J.B. and Siegel, P.H. (2001) 'On the achievable information rates of finite state ISI channels', *Global Telecommunications Conference, 2001. GLOBECOM'01*, IEEE, Vol. 5, pp.2992–2996.
- Rakesh, J. and Dalal, U. (2010) *A Survey of Mobile WiMax IEEE 802.16 m Standard*, arXiv preprint arXiv:1005.0976.
- Rakovic, V. and Gavrilovska, L. (2010) 'Novel RAT selection mechanism based on Hopfield neural networks', *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, October, pp.210–217.
- Reza, F.R. (2012) 'Optimum ranges for data transmission in mobile communications', *Int. J. Sci. Eng. Res.*, Vol. 3, pp.481–489.
- Royer, E.M. and Perkins, C.E. (1999) 'Multicast operation of the ad-hoc on-demand distance vector routing protocol', *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ACM, August pp.207–218.
- Saaty, T.L. (1988) *What is the Analytic Hierarchy Process?*, Springer Berlin Heidelberg, pp.109–121.
- Shaddad, R.Q., Mohammad, A.B., Al-Gailani, S.A., Al-hetar, A.M. and Elmagzoub, M.A. (2014) 'A survey on access technologies for broadband optical and wireless networks', *Journal of Network and Computer Applications*, Vol. 41, pp.459–472.
- Singh, A.K. and Mishra, B. (2012) 'Comparative study on wireless local area network standards', *International Journal of Applied Engineering and Technology*, Vol. 2, No. 7, July.
- Singhrova, A. and Prakash, N. (2012) 'Vertical handoff decision algorithm for improved quality of service in heterogeneous wireless networks', *IET Communications*, Vol. 6, No. 2, pp.211–223.
- Sourangsu, B. and Rahul, S.C. (2013) 'WiFi & WiMAX: a comparative study', *Journal of Indian Journal of Engineering*, Vol. 2, No. 5 [Preprint], <http://arxiv.org/ftp/arxiv/papers/1302/1302.2247.pdf> (Accessed: 8 July, 2014).
- TalebiFard, P., Wong, T. and Leung, V.C.M. (2010) 'Access and service convergence over the mobile internet – a survey', *Computer Networks*, Vol. 54, No. 4, pp.545–557.
- Tsai, C-C., Huang, H-C. and Chan, C-K. (2011) 'Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation', *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 10, pp.4813–4821.
- Van Laarhoven, P.J. and Aarts, E.H. (1987) *Simulated Annealing: Theory and Applications*, Vol. 37, Springer Science & Business Media.

- Wilson, A., Lenaghan, A. and Malyan, R. (2005) 'Optimising wireless access network selection to maintain qos in heterogeneous wireless environments', *Wireless Personal Multimedia Communications*, September, pp.18–22.
- Yan, X., Şekercioğlu, Y.A. and Narayanan, S. (2010) 'A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks', *Computer Networks*, Vol. 54, No. 11, pp.1848–1863.
- Zameer, A., Mirza, S.M. and Mirza, N.M. (2014) 'Core loading pattern optimization of a typical two-loop 300MWe PWR using simulated annealing (SA), novel crossover genetic algorithms (GA) and hybrid GA (SA) schemes', *Annals of Nuclear Energy*, Vol. 65, pp.122–131.
- Zhang, C., Wang, X. and Huang, M. (2013) 'A multi-objective genetic algorithm based handoff decision scheme with ABC supported', *Intelligent Computing Theories*, Springer Berlin Heidelberg, pp.217–226.

Bibliography

- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.A.M.T. (2002) 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.
- IEEE Computer Society LAN MAN Standards Committee (1997). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*.
- Karaboga, D. (2005) *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Vol. 200, Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Price, K., Storn, R.M. and Lampinen, J.A. (2006) *Differential Evolution: A Practical Approach to Global Optimization*, Springer Science & Business Media.
- Storn, R. and Price, K. (1997) 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *Journal of Global Optimization*, Vol. 11, No. 4, pp.341–359.

Appendix

Appendix A GSAVHO Algorithm

```
function [BestCost nfe AllBestSAGA BestSAGA BandwidthSAGA BERSAGA DelaySAGA SecurityLevelsSAGA
MondetaryCostSAGA]=saga(SelecRandomNet)
CostFunction=@(x) Sphere(x); % Cost Function
SetNetParameter=@(Net) InitializeNetwork(Net);
nVar=9; % Number of Decision variables
VarSize=[1 nVar]; % Variables Martix Size
VarMin=-10; % Variables Lower Bound
VarMax= 10; % Variables Upper Bound
if numel(VarMin)==1 % If VarMin is Scalar
    VarMin= repmat(VarMin,VarSize); % Convert VarMin to Vector
end
if numel(VarMax)==1 % If VarMax is Scalar
    VarMax= repmat(VarMax,VarSize); % Convert VarMax to Vector
end
%% GA Parameters
tic;
MaxIt=500; % Maximum Number of Iterations
MaxSubIt=10; % Maimum Number of Sub-iterations
T0=10; % Initial Temp.
alpha=0.99; % Temp. Reduction Rate
nPop=3; % Population Size
pc=2; % Crossover Percentage
nc=2*round(pc*nPop/2); % Number fo Parents (Offsprings)
pm=1; % Mutation Percentage
nm=round(pm*nPop); % Number of Mutants
gamma=0.2; % Crossover Inflation Rate
mu=0.1; % Mutation Rate
MutationMode='rand'; % Mutation Mode
eta=0.1; % Mutation Step Size Ratio
sigma=eta*(VarMax-VarMin); % Mutation Step Size
%% Initialization
global NFE;
NFE=0;
% Create Empty Structure
empty_individual.Position=[];
empty_individual.Cost=[];
empty_individual.Net=[];
```



```

AllBestSAGA=[];
BestSAGA.Position=[];
BestSAGA.Cost=[];
BestSAGA.Net=[];
BandwidthSAGA=[];
BERSAGA=[];
DelaySAGA=[];
SecurityLevelSAGA=[];
MondetaryCostSAGA=[];

% Create Structre Array to Save Population Data
pop= repmat(empty_individual,nPop,1);
% Initilize Population
for i=1:nPop
% Create Random Solution (Position)
%SelecRandomNet=randi(4);
pop(i).Position=SetNetParameter(SelecRandomNet(i));%unifrnd(VarMin,VarMax,VarSize);
% Evaluate Solutions
    pop(i).Cost=CostFunction(pop(i).Position);
    pop(i).Net=SelecRandomNet(i);
end
%Temporary
temp=pop;
% Sort Population
Costs=[pop.Cost];
[Costs, SortOrder]=sort(Costs);
pop=pop(SortOrder);
% Store Best Solution Ever Found
BestSol=pop(1);
% Create Array to Hold Best Cost Values
BestCost=zeros(MaxIt,1);
% Create Array to Hold NFEs
nfe=zeros(MaxIt,1);
% Initialize Temp.
T=T0;
%% GA Main Loop
for it=1:MaxIt
for subit=1:MaxSubIt
% Perform Crossover
    popc=repmat(empty_individual,nc/2,2);
for k=1:nc/2
% Select First Parent
        i1=randi([1 nPop]);
        p1=pop(i1);
% Select Second Parent
        i2=randi([1 nPop]);
while i1==i2 i2=randi([1 nPop]); end
        p2=pop(i2);
% Perform Crossover
        [popc(k,1).Position, popc(k,2).Position]=...
        ArithmeticCrossover(p1.Position,p2.Position,gamma,VarMin,VarMax);
% Evaluate Offsprings
        popc(k,1).Cost=CostFunction(popc(k,1).Position);
        popc(k,2).Cost=CostFunction(popc(k,2).Position);
%Network
        popc(k,1).Net=p1.Net;
        popc(k,2).Net=p2.Net;
end
        popc=popc(:);
% Perform Mutation
        popm=repmat(empty_individual,nm,1);
for l=1:nm
% Select Parent
        i=randi([1 nPop]);
        p=pop(i);
% Perform Mutation
        popm(l).Position=...
        Mutate(p.Position,mu,MutationMode,sigma,VarMin,VarMax);

```

```

% Evaluate Mutatnt
    popm(1).Cost=CostFunction(popm(1).Position);
    popm(1).Net=p.Net;
end
% Merge Offsprings Population
    newpop=[popc
            popm];
% Sort NEWPOP
    [~, SortOrder]=sort([newpop.Cost]);
    newpop=newpop(SortOrder);
% Compare using SA Rule
for i=1:nPop
    if newpop(i).Cost<=pop(i).Cost
        pop(i)=newpop(i);
    else
        DELTA=(newpop(i).Cost-pop(i).Cost)/pop(i).Cost;
        P=exp(-DELTA/T);
        if rand<=P
            pop(i)=newpop(i);
        end
    end
% Update Best Solution Ever Found
    if pop(i).Cost<=BestSol.Cost
        BestSol=pop(i);
    end
end
% Store Best Cost
    BestCost(it)=BestSol.Cost;
%*****
    BestSAGA(it).Position=BestSol.Position;
    BandwidthSAGA(it)=temp(BestSol.Net).Position(1);
    BERSAGA(it)=temp(BestSol.Net).Position(2);
    DelaySAGA(it)=temp(BestSol.Net).Position(3);
    SecurityLevelsSAGA(it)=temp(BestSol.Net).Position(4);
    MondetaryCostSAGA(it)=temp(BestSol.Net).Position(5);
    BestSAGA(it).Cost=BestSol.Cost;
    BestSAGA(it).Net=BestSol.Net;
    AllBestSAGA(it)=BestSol.Net;
%*****
% Store NFE
    nfe(it)=NFE;
% Show Iteration Information
    disp(['Iteration ' num2str(it) ...
': Best Cost = ' num2str(BestCost(it)) ...
', NFE = ' num2str(nfe(it))]);
% Temp. Reduction
    T=alpha*T;
end
toc;
fprintf('TIC TOC: %g\n', toc);

```
