Received: April 1, 2021. Revised: May 6, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Genetic Algorithms and Particle Swarm Optimization for Interference Minimization in Mobile Network Channel Assignment Problem

Joe Siang Keek[1] **Ser Lee Loh[1]*** **Yan Chiew Wong[1]** **Xiu Juan Woo[1]** **Wei Wen Lee[1]**

[1]*Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia*
* Corresponding author's Email: slloh@utem.edu.my

Abstract: Interference minimization in cellular network has and will always be top priority, whether in current or future generation of cellular technology. Therefore, cellular channel assignment problem (CAP) requires continuous study and research. This paper presents the study and comparison of Genetic Algorithm (GA) and Particle Swam Optimization (PSO) for CAP in minimizing interference. GA with three variants in term of population selection – roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS) were studied, and then compared with classic PSO. Two CAPs were derived and used to comprehensively evaluate the performances of the PSO and GAs. It was found that GA-TS is ~11% and ~7% faster than GA-RWS and GA-SUS, respectively. Although the difference is small, but it allowed GA-TS to run for few more iterations and eventually achieved better interference minimization. Moreover, it was also found that GA-SUS has less noise and produce a more consistent result. On the other hand, PSO is slower than GA-TS, but has higher potential to converge on smaller minimum value.

Keywords: Channel assignment problem, Spectrum sharing, Genetic algorithm, Roulette wheel selection, Tournament selection, Stochastic universal sampling, Particle swarm optimization.

1. Introduction

A specified range of radio frequency spectrum is reserved and allocated for cellular network and communication purposes. The allocated spectrum is then segmented into smaller bands known as channels. Data transmission then happens within each channel for the nodes that are tuned with that particular channel frequency range. The channel available for the cellular communication is already limited, yet with the need to accommodate increasing population over time, channel assignment problem (CAP) is foreseen to become more and more challenging. Therefore, CAP is often an important research topic that requires continuous study and research from time to time. The number of users in cellular network is predicted to increase by ten times within five years [1].

Traditionally, channel assignment (CA) can be classified into types of static, dynamic and hybrid [2]. Static CA involves channels that are fixed over a region. Whereas, the access of the channels in dynamic system is user-demand dependent. Dynamic CA is overall less efficient than static CA in handling heavy traffic [3, 4]. Hybrid CA is a combination of static and dynamic CAs, where a central pool of static and dynamic channels exists [5, 6]. Now. in the era of fifth generation telecommunication technology (5G), channel assignment is way beyond classical static, dynamic and hybrid system. Unlike previous generations, the 5G network, which is known as heterogeneous network (HetNet), is a complex cellular network that has the highest densifications over space and spectrum than ever before [1, 7-9]. HetNet is a cognitive network in such a way that all available radio spectrum or channel that coexist within a region will be fully utilized for both primary and secondary users [10-12], including television signal white space [13-15]. Such spectrum sharing or borrowing can happen among licensed and unlicensed networks [16-19]. In 5G, millimetre wave (mmWave) is newly introduced as well as part of the architecture. mmWave signal has much higher transmission speed, but has low penetration power

[20-22]. Therefore, more network cells such as picocell and femtocell are anticipated at the higher level of cellular network hierarchy [21-23]. All and all, these densifications imply exponential growth of data transmission within a more stringent bandwidth, which will surely increase the vulnerability of end-user towards interference.

Telecommunication signal interference can be divided into three types, namely co-channel interference, adjacent-channel interference and co-site interference [24]. Co-site interference happens between the channels under one particular cell. Whereas, adjacent-channel interference and co-channel interference are interference happens with adjacent cells and cells beyond, respectively. The total interference will then determine the quality of all calls within the cell and network. Whether 1G, 5G or even upcoming generations, optimizing the channel assignment to minimize interference will always be the elementary.

Many optimization methods had been proposed in the past for interference minimization: ant colony optimization (ACO) [25], genetic algorithm (GA) [1, 26-29], particle swarm optimization (PSO) [1, 30-32], tabu search (TS) [33-35] et cetera. Evolutionary computing algorithm such as GA is said to be effective for CAP and is more computational-friendlier [27, 36].

The organization of this paper is as follows: general characteristics and performances of GA and PSO in CAP were first discussed, then followed by GA with three different population selection methods, namely roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS). The main contribution of this paper is to provide an insight on these selection methods on their performances in CAP. At the final section of this paper, GA-TS was evaluated with PSO in CAP1 and CAP2. To make the evaluation more realistic, CAP2 is derived based on real-life graphical data and situation.

2. Related work

Channel assignment problem (CAP), which is also known as network selection or spectrum allocation, can be abundantly found in the literature. The parameters to be optimized can be total interference, data speed, price, signal strength et cetera [10-12, 31, 37, 38]. Optimizing these parameters at the same time is a multi-objective optimization process. Most of the time, optimizing total interference is essential, whereas the other parameters are optional, depending on the intention or objective of respective authors.

[1] et al. presented GA and PSO in minimizing interference and price that has to be paid by secondary user, under the constraint of data transmission speed as well. The results showed that the GA outperform the PSO at the end of the evaluation. Anyhow, a repair process was introduced by the authors, in which when infeasible assignment such as clashing of channels, out-of-range allocation etc. happen, a repeat step is triggered to regenerate new position for the infeasible chromosome or particle. This step is repeat until feasible solution is found.

[37] et al. integrated auxiliary scheme – nonlinear one-leader-multiple follower (OLMF) and nonlinear bilevel OLMF with PSO in optimizing pricing during the spectrum sharing process. [32] et al. presented PSO to optimize parameters such as signal-to-noise ratio, spectral efficiency and power consumption in a dynamic CAP. The authors mentioned that higher number of particle and iteration will improve the overall fitness. However, and logically, such statement is reasonable but up to certain extend, that is before the variables – number of particle and iteration become saturated. In the paper, no comparison was made with the proposed PSO.

Next, [27] et al. proposed GA with roulette wheel selection (RWS) in optimizing spectrum allocation under the constraint or requirement of quality of service (QoS) for end-user. No comparison is done between the proposed GA with other optimization method. In [29], the authors proposed GA for fixed, dynamic and hybrid CAPs. The performance of the GA in these three CAPs were analysed and discussed.

In this paper, two complex fixed CAPs were defined and used. One of the CAPs is based on real-life channel allocation situation in Helsinki, Finland [41], so that the evaluation produced is more convincing. GA with three types of population selection methods – roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS) were analysed and discussed in this paper. The best GA variant was then selected to compare with classic PSO. In CAP, the occurrence of infeasible channel allocation is very common among the PSO and GA. This is due to the fact that channel interference minimization process does not has continuous descending gradient. In fact, there are a few infeasible channel allocation 'loopholes' at the descending slope that will endlessly trap the algorithm if no repair action is done. Moreover, due the randomness in PSO and GA, the particle or chromosome may accidentally have repeating channel allocation or allocating channel that is out of the feasible range. Therefore, in this paper, a repair action is proposed to overcome this problem. The repair action is a random sampling without replacement method that will generate new position and new gene for the particle and chromosome respectively. Complete detail on the formulations of the PSO and GA are discussed in upcoming section.

3. Methodology

In this section, the formulation of the channel assignment problem (CAP) used in this paper is first presented, then followed by the optimization methods: particle swarm optimization (PSO) and genetic algorithm (GA). GA with different population selection methods variants: roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS) are also discussed in this section

3.1 Problem formulation

We consider that a network has N cells. Each cell has M available channels that can be supplied to users based on demand D. Notation D is therefore a one-dimensional (1 × N) matrix. Non-interference constraint C is a symmetrical N × N matrix that expresses the minimum frequency separation between the channels that were occupied by users. For example, C12 = 3 denotes that the channels in cell 1 and cell 2 have to be at least 3 steps apart or else, the interference will be added to the fitness function. The fitness function is as shown in Eq. (1).

$$F(X) = \left(\sum_{j=1}^{N} \sum_{k=1}^{M} X_{j,k} \sum_{i=1}^{N} \sum_{l=1}^{M} P_{j,i,(m+1)} X_{i,l} \right) \Big/ 2 \quad (1)$$

where $m = |k - l|$. Notation Xj,k denotes binary variable that has properties as shown in Eq. (2),

$$X_{j,k} = \begin{cases} 1, \text{if channel } k \text{ of cell } j \text{ is allocated to user} \\ 0, \text{otherwise} \end{cases} \quad (2)$$

where $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, M$. Next, Pj,i,(m+1) denotes cost tensor which indicates the severity of the interference based on non-interference constraint Cj,i. The formula of the cost tensor is as shown in Eq. (3).

$$P_{j,i,(m+1)} = \max(0, C_{j,i} - m) \quad (3)$$

Fitness function F(X) sums up the total interference caused by the overall channel allocations; higher value indicates higher severity in overall interference. The pseudo-code of the fitness function can be found in Algorithm 1.

Algorithm 1: Fitness function

Result: Return penalty or computed *F*(*X*) value
for *particle's position / chromosome's gene* **do**
  if *similar position / gene* **then**
    return penalty value
end
Compute fitness value
return fitness value

As shown in Algorithm 1, infeasible channel allocation will lead to return of penalty value by the function. The penalty value serves as an indicator for the optimization algorithm to carry out repair action. Otherwise, the computation of fitness value will be as usual, that is based on Eq. (1).

3.2 Particle swarm optimization

Particle swarm optimization (PSO) involves a group or particles that work together in searching for global minimum position [39]. Each particle has respective position. The position is an array or list, and its size is depending on the dimension of the search problem. In channel assignment problem (CAP), the dimension of the search problem is based on total number of calls or users in the previous-mentioned one-dimensional matrix demand, D. At every iteration, the action of each particle is affected by the last action taken, personal best position and global best position. Each of these factors are weighted by respective gain or known as hyperparameter. Randomness is also added to create arbitrary action as an effort for the particle to explore. The formula of particle's velocity (action) at each iteration is as shown in Eq. (4).

$$\begin{aligned} v_k^n = w v_{k-1}^n + c_1 r_1 (p_{k-1}^n - x_{k-1}^n) \\ + c_2 r_2 (g_{k-1} - x_{k-1}^n) \end{aligned} \quad (4)$$

Notation v_k^n denotes the velocity of particle n at timestep k. Understandably, timestep $k - 1$ denotes one step earlier than current iteration. Notations w, c1 and c2 are the gains, whereas r1 and r2 are random decimal value ranges from 0 to 1. Notation x_{k-1}^n denotes the position of particle n at timestep $k - 1$. Finally, p_{k-1}^n denotes the particle's personal best position so far, whereas g_{k-1} denotes global best record, in other words, best position among all particles. By the end of each iteration, these best records will be updated. All particles are also updated with new position using formula as shown in Eq. (5).

$$p_k^n = p_{k-1}^n + v_k^n \quad (5)$$

Received: April 1, 2021. Revised: May 6, 2021.

PSO is prominent for its simplicity. Due to the factor of global best position, convergence of PSO is often promising. However, PSO consumes higher computation power due to a few stages of calculation, and yet they involve decimal. Besides that, assignment of channel is in integer form, therefore the position and velocity of the particle will be rounded-off to nearest integer. Comparing PSO with genetic algorithm, the latter is less computational heavy as it involves only binary value. The pseudo-code of PSO is as shown in Algorithm 2.

Algorithm 2: Particle swarm optimization

Result: Return global best position; channel allocation with least interference.
Initialize particles and their positions
Store initial personal and global best records
for *less than maximum iteration* **do**
    for *each particle* **do**
        Obtain fitness value from current position
        if *fitness value == penalty value* **then**
            Repair action
        if *fitness value < personal best record* **then**
            Update to personal best record
        if *fitness value < global best record* **then**
            Update to global best record
        Compute velocity
        Compute new position with the velocity
    end
end

For the repair action mentioned in Algorithm 2, the action intends to repair infeasible channel allocation such as particle going beyond the range of available channels and occupation of channel by more than one user. The repair action is generally made up of random sampling function and floor-and-ceiling function. They ensure that the newly generated particle's position is feasible.

3.3 Genetic algorithm

Genetic algorithm (GA) is an evolutionary algorithm that enables chromosomes to evolve and become better in upcoming generations. GA does not involve formula but instead, it involves a series of operation or step – ranking, selection, breeding and mutation. Each chromosome is made up of gene, whereas each gene consists of binary blocks. The amount of gene within a chromosome is depending on the total number of calls, which is based on the demand, D. Fig. 1 shows an example of chromosome.

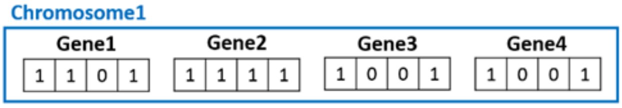

Figure. 1 A chromosome for channel assignment problem

Based on Fig. 1, the chromosome is applicable to CAP that has total of four demands, gene has four bits and can therefore support up to $2^4 = 16$ of channels per cell. Since GA involves binary values whereas CAP involves integers, therefore, encoding and decoding are needed during the computation of fitness value.

In GA, breeding and mutation happen directly on the most fundamental elements, which are the bits and binary blocks. The algorithm of GA is as shown as Algorithm 3.

Algorithm 3: Genetic algorithm

Result: Return global best chromosome; channel allocation with least interference.
Initialize chromosomes and their genes
for *less than maximum generation* **do**
    for *each chromosome* **do**
        if *fitness value == penalty value* **then**
            Repair action
        Compute and store fitness value
    end
    Rank population based on their fitness values
    New population selection via RWS / TS / SUS
    Breeding via crossover
    Mutation
end

Elitism is used for the GA of this paper. For example, if elite size, e equals to 3, the top three best chromosomes (top three lowest interference) will be retained throughout the generation. In other words, well-performed chromosomes are protected from being altered due to crossover and mutation. Elitism ensures better convergence for the optimization. Next, breeding and mutation are two important actions in searching for global minimum. These two actions introduce randomness as an exploring effort in the search space. Fig. 2 illustrated the concept of crossover. Based on the figure, Chromosome1 and Chromosome 2 are parents whereas Chromosome3 are their child after the crossover. The crossover points are selected randomly each time. Also, prior to the crossover process, all chromosomes in the population are randomly rearranged to create a randomized mating pool.
Once a new population of new-born chromosomes are created from the crossover, the population will then pass through the mutation stage. Mutation flips

Received: April 1, 2021. Revised: May 6, 2021.

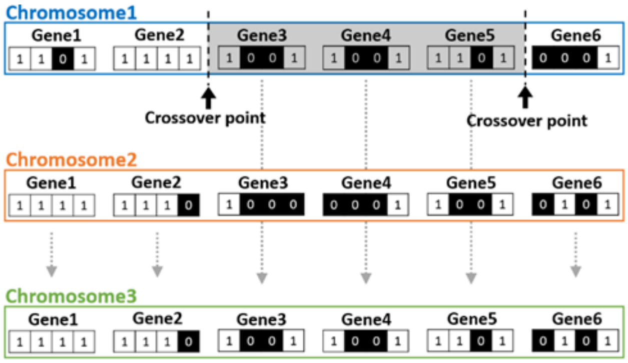

Figure. 2 The concept of crossover in GA

the binary value of the gene. Mutation rate, m is another hyperparameter for GA to control the likelihood of mutation. The parameter should have decimal value within 0 and 1. The closer the value of m towards 1, the higher the number of genes undergo mutation. Mutation plays a vital role in GA. Even though good convergence is met, mutation tends to continue the searching for better result. It also serves to prevent the population from completely trapped in local minimum.

In GA, new generation will have new population. Similar to the lives on Earth, only living things that fulfil the survival requirements will survive. Therefore, in the population selection in Algorithm 3, the selection will be based on certain 'survival requirement' as well. Firstly, the top e (elite size) best chromosomes will first advance to new population. The remaining non-elite chromosomes will be selected based on desired selection method. In this paper, three selection methods will be discussed and analysed – roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS). The properties and features of RWS, TS and SUS are presented in upcoming sub-section.

3.3.1. Roulette wheel selection

Roulette wheel selection (RWS) is also known as fitness proportionate selection. Generally, in RWS, chromosome with lower interference has higher chance of advancing to new population. The selection happens in RWS is based on cumulative percentile, which is also the probability of being selected. The formula of calculating cumulative percentile of each chromosome is as shown as Eq. (6).

$$CP_k^n = \frac{(1/F_k^n)+CP_k^{n-1}}{\sum_{n=1}^{N}(1/F_k^n)} \times 100\% \tag{6}$$

Notation CP_k^n denotes cumulative percentile of n^{th} chromosome, whereas CP_k^{n-1} denotes cumulative sum of $(n-1)^{\text{th}}$ chromosome, both at generation k.

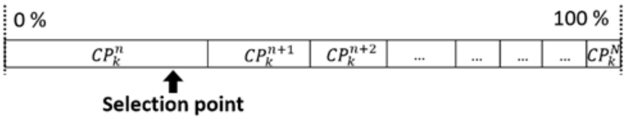

Figure. 3 The concept of roulette wheel selection

The population has a total of N chromosomes. The denominator, $\sum_{n=0}^{N}(1/F_k^n)$ is actually equals to CP_k^N. Since CAP is a minimization problem, fitness inverse, $1/F_k^n$ is needed in Eq. (6) instead of fitness, F_k^n. If not, the infeasible channel allocation (penalty value or infinity large value) will cause the CP_k^N to become infinity, and eventually, CP_k^n becomes 0 for all chromosomes. Anyhow, by using fitness inverse, zero interference $F_k^n = 0$ will cause invalid calculation as well. In order to this issue, whenever zero interference channel allocation is achieved by the chromosome, inverse fitness will become 2. Whereas for chromosome with non-zero interference and feasible channel allocation, its fitness value will be within 0 and 1. It is important to mention that such approach is applied to the cases of TS and SUS as well. Fig. 3 illustrates the concept of RWS.

Based on Fig. 3, chromosome with lower interference has higher chance of being selected for the new population. The position of the 'selection point' is decided randomly, therefore, chromosome with higher interference still has chance to being selected.

3.3.2. Tournament selection

Generally, as what the name implies, tournament selection (TS) creates a tournament for a group randomly picked candidates to compete with each other. If one chromosome is picked more than once, repick will be done. The size of the group for the tournament, or the number of candidates, K is a hyperparameter for the TS. Prior to the tournament, the candidates will first be ranked in term of their fitness. Hyperparameter selection pressure, S decides the winning probability of the first candidate *viz.* candidate with best fitness among others. The winning probability of the candidates is as shown as Eq. (7). It is noteworthy that the first candidate has winning probability of equals to selection pressure, S. The probability of winning decreases for successive candidates [40].

$$\alpha_k^n = S((1-S)^{n-1}), \text{for } n = 1,2, \dots K \tag{7}$$

Anyhow, there is a problem when tournament size K is small. Table 1. shows winning probabilities of candidates when $K = 7$ and $K = 3$, let S equals to 0.5.

Received: April 1, 2021. Revised: May 6, 2021.

Table 1. Example situation when $K = 7$ and $K = 3$

n	α_k^n	
1	0.5	0.5
2	0.25	0.25
3	0.125	0.125
4	0.0625	–
5	0.03125	–
6	0.01525	–
7	0.0078125	–
$\sum_{n=1}^{K} \alpha_k^n$	0.992	0.875

Based on Table 1, let us say a random number 0.92 is generated, for situation where $K = 7$, candidate $n = 4$ is selected to win the tournament. However, for situation where $K = 3$, no candidate wins because the total probability is 0.875, which is less than the random number. To solve this problem, a cumulative and normalization method as shown in Eq. (8) is introduced to ensure that no matter the size of K, the summation will always equal to 1. Table 2 shows the outcome of using Eq. (8) on the data in Table 1.

$$CP_k^n = \frac{\alpha_k^n + \alpha_k^{n-1}}{\sum_{n=1}^{K} \alpha_k^n}, \text{for } n = 1,2, \dots K \tag{8}$$

As TS randomly picks candidates to participate in the tournament, it helps the GA from merely relying on good chromosomes which may cause premature convergence or trapping at local minimum. In additional with the implementation of selection pressure S, more variation and randomness are introduced which possible in helping the GA to broadly explore possible better minimum. In comparison with RWS, TS is more computational power friendly as it does not iterate through the whole population but instead, it focuses on creating a tournament with lesser chromosomes. Comparison of RWS and TS are presented and discussed in the next section.

3.3.3. Stochastic universal sampling

Generally, the selection method by stochastic universal sampling (SUS) is quite similar with RWS, as both of them utilizes cumulative percentile, CP_k^n. The major difference is that SUS uses multiple selection points (also known as pointers) instead of only one. The concept of SUS can be explained clearly via illustration, as shown in Fig. 4.

Based on Fig. 4, it assumes that the GA requires to select eight chromosomes from the population to form a new population. Therefore, 100 % divided by 8, which is 12.5 % represented by notation d.

Table 2. Similar situation as previous but using Eq. (8)

n	CP_k^n	
1	0.504	0.571
2	0.756	0.857
3	0.882	1.000
4	0.945	–
5	0.977	–
6	0.992	–
7	1.000	–

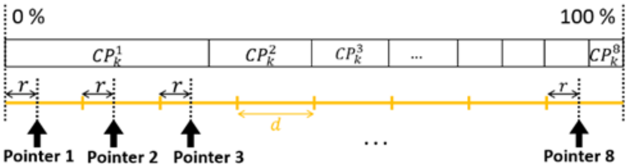

Figure. 4 The concept of stochastic universal sampling

Whereas, notation r is a random number generated at each generation. SUS may be faster than RWS because the former iterates the population once and does not require to continuously generate new pointer. However, based on the figure, CP_k^1 is selected three times into the new population, which is reasonable because chromosome $n = 1$ has the most promising fitness than others. This may cause the GA to end up converging towards local minimum.

4. Simulation results

For all the simulation results in this paper, one similar computer is used. To ensure the results obtained are at utmost reliability and accuracy, the authors ensured that the computer has no other background processes running each time during the simulations. Anyhow, the computer used for the experiments in this paper is nowhere comparable to a high-end computer or supercomputer; only a common laptop with Intel Core i5 processor and 8GB RAM. Therefore, comparison between the optimization methods and the results in this paper with the ones in the literature in terms of computational time and performance should be unrealistic.

4.1 Simulation settings for channel assignment problem

In this paper, two channel assignment problems (CAPs) i.e. CAP1 and CAP2 are defined and implemented to evaluate the performance of the optimization methods i.e. particle swarm optimization (PSO) and genetic algorithm (GA). The complete detail of CAP1 is as shown in Table 3.

Based on Table 3, CAP1 is considered to have small N and M, but has challenging D and C. Based on C_{CAP1}, it can be noticed that the co-site non-

Table 3. Setting of CAP1

Parameters	Values
Number of cells, N_{CAP1}	8
Number of channels, M_{CAP1}	30
Demand, D_{CAP1}	[5 4 3 4 3 4 2 0]
Non-interference constraint, C_{CAP1}	$\begin{bmatrix} 6&5&4&3&2&1&0&0 \\ 5&6&5&4&3&2&1&0 \\ 4&5&6&5&4&3&2&1 \\ 3&4&5&6&5&4&3&2 \\ 2&3&4&5&6&5&4&3 \\ 1&2&3&4&5&6&5&4 \\ 0&1&2&3&4&5&6&5 \\ 0&0&1&2&3&4&5&6 \end{bmatrix}$

interference constraint (diagonal term) is stringent. Overall, finding global minimum for CAP1 should be challenging and sufficient to validate the performance of the optimization method.

Next, CAP2 is derived based on real-life graphical data and situation i.e. 24 km × 21 km area around Helsinki, Finland [41]. The parameters of CAP2 is as shown in Table 4.

Based on Table 4, it is significant that CAP2 is much more challenging in terms of N, M and D. High computation power consumption is expected even though C_{CAP2} is less stringent than C_{CAP1}. These characteristics make CAP2 different from CAP1, thus making the evaluation in this paper to be more comprehensive. Moreover, since CAP2 is based on real-life situation, the simulation result is more convincing.

Table 4. Setting of CAP2

Parameters	Values
Number of cells, N_{CAP2}	15
Number of channels, M_{CAP2}	44
Demand, D_{CAP2}	[10, 11, 9, 5, 9, 4, 5, 7, 4, 8, 8, 9, 10, 7, 7]
Non-interference constraint, C_{CAP2}	$\begin{bmatrix} 2&1&1&0&1&0&1&1&1&1&0&1&1&1&1 \\ 1&2&1&0&1&0&1&1&0&1&0&1&1&1&1 \\ 1&1&2&1&1&1&1&1&1&1&1&1&1&1&0 \\ 0&0&1&2&0&0&1&1&1&1&1&1&1&0&0 \\ 1&1&1&0&2&0&0&0&0&1&1&1&1&1&1 \\ 0&0&1&0&0&2&1&1&1&1&0&0&0&0&0 \\ 1&1&1&1&0&1&2&1&1&1&1&1&1&0&0 \\ 1&1&1&1&0&1&1&2&1&1&1&1&1&0&0 \\ 1&0&1&1&0&1&1&1&2&1&1&1&0&0&0 \\ 1&1&1&1&1&1&1&1&1&2&1&1&1&1&1 \\ 0&0&1&1&1&0&1&1&1&1&2&0&1&1&1 \\ 1&1&1&1&1&0&1&1&1&1&0&2&1&1&0 \\ 1&1&1&1&1&0&1&1&0&1&1&1&2&1&1 \\ 1&1&1&0&1&0&0&0&0&1&1&1&1&2&1 \\ 1&1&0&0&1&0&0&0&0&1&1&0&1&1&2 \end{bmatrix}$

4.2 Preliminary simulation results of GAs on CAP1

This subsection presents, analyses and discusses the performance of genetic algorithm (GA) with variants roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS). The settings of the hyperparameters is as shown in Table 5.

Since M_{CAP2} and M_{CAP2} are 30 and 44 respectively, the bit sizes per gene in CAP1 and CAP2 are then 5 and 6, respectively. Based on Table 5, simulation was done on CAP1. The results are tabulated in Tables 6, 7, and 8. Figs. 5 and 6 show the results in plots.

Based on Fig. 5, among three of the variants, GA-SUS displays most consistent 'final best fitness' value, with variance and standard deviation of only 4.46 and 2.11 respectively, which are lowest among the two other variants. Anyhow, GA-SUS did not manage to attain lowest 'final best fitness' value in ten runs, with only having value of 91 as its lowest.

Table 5. Hyperparameter value of GAs on solving CAP1

Hyperparameter	Population selection variants		
	RWS	TS	SUS
Population size	40	40	40
Maximum generation	20	20	20
Elite size, e	4	4	4
Mutation rate, m	0.02	0.02	0.02
Tournament size, K	–	5	–
Selection pressure, S	–	0.8	–

Table 6. Result of GA-RWS for solving CAP1

Experiment no.	GA-RWS			
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	106	89	17	148.89
2	112	93	19	149.27
3	124	98	26	142.73
4	131	89	42	145.01
5	110	95	15	134.37
6	125	95	30	136.57
7	121	98	23	143.06
8	108	93	15	143.31
9	120	94	26	143.91
10	120	96	24	145.72
Variance	–	10	–	22.38
Std. dev.	–	3.16	–	4.73
Mean	–	94	–	143.28

Table 7. Result of GA-TS for solving CAP1

Experiment no.	GA-TS			
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	128	99	29	126.75
2	118	96	22	128.05
3	113	94	19	139.72
4	121	91	30	133.89
5	117	94	23	127.77
6	141	99	42	120.13
7	116	89	27	141.50
8	125	95	30	125.54
9	117	92	25	123.44
10	128	90	38	126.45
Variance	–	12.1	–	47.73
Std. dev.	–	3.48	–	6.91
Mean	–	93.9	–	129.32

Table 8. Result of GA-SUS for solving CAP1

Experiment no.	GA-SUS			
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	115	98	17	138.77
2	119	93	26	140.66
3	125	94	31	141.96
4	121	91	30	141.24
5	140	93	47	131.95
6	122	96	26	138.44
7	119	92	27	131.85
8	120	93	27	142.98
9	124	95	29	140.92
10	126	92	34	136.36
Variance	–	4.46	–	15.76
Std. dev.	–	2.11	–	3.97
Mean	–	93.7	–	138.51

Whereas, both GA-RWS and GA-TS successfully reached ‘final best fitness’ of 89. GA-RWS has two 89s whereas GA-TS has one 89 and one 90. The variance, standard deviation and mean of GA-RWS and GA-TS are also very much similar.

Even though the ‘final best fitness’ performances of GA-RWS and GA-TS are quite similar, but in term of ‘time taken’ for 20 generations, GA-TS significantly achieved the best among all as shown in Fig. 6, with average ‘time taken’ of 129.32s. This indicates that GA-TS has plenty of space to accommodate larger population size and longer generation to improve its ‘final best fitness’. Therefore, based on this ideology, the hyperparameters of GA-TS - population size, maximum generation, tournament size, K and selection pressure, S are fined tuned so that its average ‘time taken’ is comparable with GA-RWS.

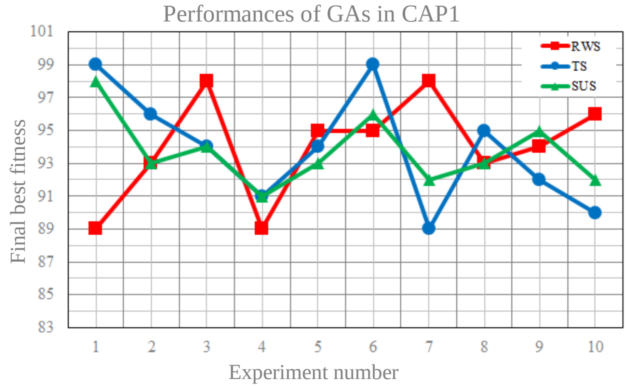

Figure. 5 Performances of GAs in ‘final best fitness’

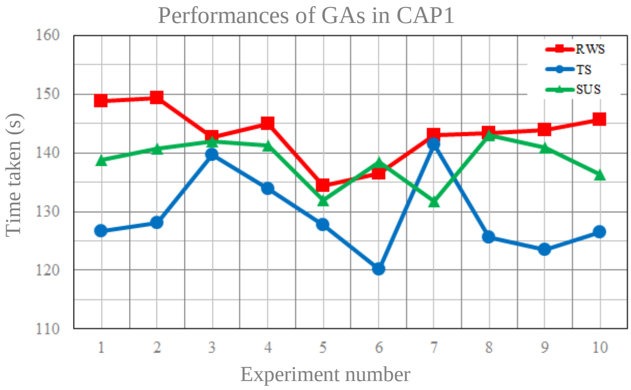

Figure. 6 Performances of GAs in computational time

The new setting for the GA-TS is as shown in Table 9. Even though GA-SUS has slightly lower ‘time taken’ than GA-RWS, but it was eliminated for next round of evaluation because its ‘final best fitness’ is not competitive with others. Anyhow, it was found that GA-SUS produced significant consistent result in both ‘final best fitness’ and ‘time taken’. This shows that SUS certainly has reduced stochastic noise during the optimization. By comparing Tables 5 and 9, the values of maximum generation, K and S had been changed. Meanwhile, it was found that increasing the population did not help to improve the final best fitness. Also, K is reduced to 3 so that the its ‘time taken’ is comparable with GA-RWS. It was also found that by reducing S to 0.5, a more consistent ‘final best fitness’ result is obtained. The simulation result of GA-TS-new on solving CAP1 is as shown in Table 10, Figs. 7 and 8. Based on Fig. 7, GA-TS-new significantly has better performance than GA-RWS and GA-TS in overall. In terms of variance, standard deviation and mean of ‘final best fitness’, GA-TS-new has the lowest among GA-TS and GA-RWS. The ‘time taken’ of GA-TS-new is quite similar with GA-RWS, i.e. 141.27s and 143.28s, respectively. Therefore, this indicates that under a similar ‘time taken’ range, GA-TS-new managed to get lower ‘final best fitness than GA-RWS.

Table 9. Hyperparameter value of GA-TS-new for solving CAP1

Hyperparameter	GA-TS-new
Population size	40
Maximum generation	25
Elite size, e	4
Mutation rate, m	0.02
Tournament size, K	3
Selection pressure, S	0.5

Table 10. Result of GA-TS-new for solving CAP1.

Experiment no.	GA-TS-new Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	131	91	40	146.89
2	115	90	25	149.76
3	101	92	9	147.32
4	117	92	25	133.56
5	120	90	30	141.57
6	135	90	45	144.04
7	133	93	40	142.19
8	113	83	30	128.43
9	122	92	30	126.32
10	127	92	35	152.58
Variance	–	8.06	–	80.51
Std. dev.	–	2.84	–	8.97
Mean	–	90.5	–	141.27

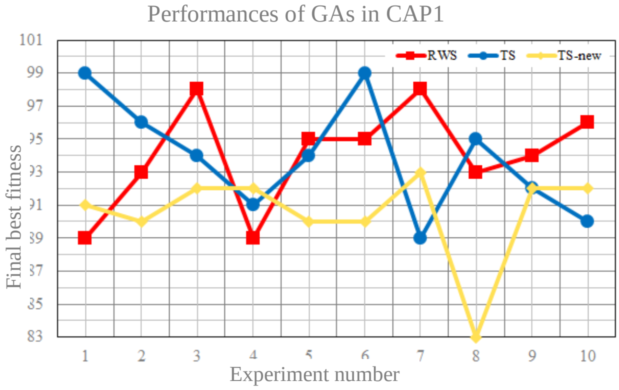

Figure. 7 Performances of GAs in 'final best fitness'

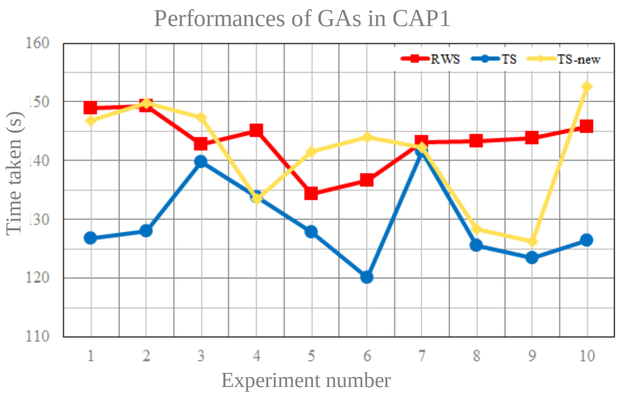

Figure. 8 Performances of GAs in computational time

4.3 Simulation results of GA-TS and PSO on CAP1

Based on the last subsection, it was found that GA-TS with K = 3 and S = 0.5 produces better result. Thus, the values will be retained to compare with particle swarm optimization (PSO) in this subsection. Tables 11 and 12 show the setting of PSO and GA-TS respectively for solving CAP1.

The values of w, c_1 and c_2 shown in Table 11 were obtained through trial-and-error. Intuitively, global best gain c_2 should be larger than personal best gain c_1 so that the particles will converge. However, due to the fact that CAP is a complex optimization problem where multiple global minimum may exist, it was found that having similar value for c_1 and c_2 tend to improve the searching effort of the particles and reduce the chances of premature convergence. Tables 13 and 14 show the performances of the PSO and GA-TS respectively on solving CAP1. A new column i.e. 'converged' is introduced to the tables to indicates whether the optimization process is converged or not – if identical fitness value in the final five iterations / generations, the optimization is considered converged (value of 1), and vice-versa.

Based on the results in Tables 13 and 14, both PSO and GA-TS had almost similar 'final best fitness', with mean value of 89.6 and 89.0, respectively. However, GA-TS has the lowest 'final fitness value' i.e. 85. Other than that, it is important to take note that seven out of ten of the GA-TS had converged, whereas for the PSO, only 20% had converged. Therefore, the PSO was then tested with 60 iterations in order to obtain its converged result. The result is as shown in Table 15.

Table 11. Hyperparameter value of PSO for solving

Hyperparameter	PSO
Number of particles	15
Maximum iteration	40
Velocity gain, w	0.7
Personal best gain, c_1	1.50
Global best gain, c_2	1.50

Table 12. Hyperparameter value of GA-TS in CAP1

Hyperparameter	GA-TS
Population size	40
Maximum generation	50
Elite size, e	4
Mutation rate, m	0.02
Tournament size, K	3
Selection pressure, S	0.5

Table 13. Result of PSO for solving CAP1 with maximum iteration of 40

Experiment no.	PSO				
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)	Converged
1	154	91	63	319.99	1
2	114	88	26	284.08	0
3	133	87	46	306.25	0
4	137	86	51	289.56	0
5	151	87	64	290.89	0
6	121	89	32	291.65	0
7	141	94	47	295.51	1
8	133	91	42	286.64	0
9	147	93	54	256.46	0
10	117	90	27	301.94	0
Var.	–	7.16	–	273.18	–
S.D.	–	2.67	–	16.53	–
Mean	–	89.6	–	292.30	–

Table 14. Result of GA-TS for solving CAP1

Experiment no.	GA-TS				
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)	Converged
1	115	92	23	294.26	1
2	131	92	39	246.29	0
3	121	90	31	274.62	1
4	119	94	25	285.76	1
5	122	85	37	310.69	1
6	124	87	37	300.45	0
7	112	86	26	293.97	1
8	102	86	16	317.00	1
9	124	90	34	278.66	1
10	107	88	19	314.93	0
Var.	–	9.33	–	465.30	–
S. D.	–	3.06	–	21.57	–
Mean	–	89.0	–	291.66	–

Table 15. Result of PSO for solving CAP1 with maximum iteration of 60

Experiment no.	PSO				
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)	Converged
1	123	87	36	464.08	0
2	122	86	36	472.31	0
3	117	90	27	469.30	1
4	138	84	54	469.58	1
5	126	89	37	445.64	1
6	112	87	25	490.95	1
7	136	87	49	444.11	1
8	136	87	49	441.60	1
9	123	87	36	457.58	0
10	126	88	38	450.11	1
Var.	–	2.62	–	244.72	–
S. D.	–	1.62	–	15.64	–
Mean	–	87.2	–	460.53	–

4.4 Simulation results of GA-TS and PSO in CAP2

As mentioned, CAP2 has significantly much larger search space than CAP1. Moreover, the total demand in CAP2 is not reduced but increased. Although non-interference constraint in CAP2 is less stringent, but CAP2 is still expected to be more challenging than CAP1. Therefore, evaluation by using CAP2 serves different purpose than CAP1.

Due to high computation needed in CAP2 and limited processing power by our simulator, both the PSO and GA-TS were running for 10 iterations only.

Received: April 1, 2021. Revised: May 6, 2021.

Table 16. Result of PSO for solving CAP2

Experiment no.	PSO			
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	141	100	41	989.72
2	128	94	34	998.29
3	114	98	16	977.08
4	120	97	23	976.23
5	124	101	23	976.58
6	109	100	9	980.22
7	139	91	48	976.67
8	138	96	42	991.21
9	118	101	17	977.64
10	126	100	26	980.21
Variance	–	11.07	–	61.00
Std. dev.	–	3.33	–	7.81
Mean	–	97.8	–	982.39

Table 17. Result of GA-TS for solving CAP2

Experiment no.	GA-TS			
	Initial best fitness	Final best fitness	Fitness delta	Time taken (s)
1	122	95	27	998.91
2	112	93	19	999.03
3	123	96	27	997.56
4	136	93	43	1029.71
5	120	98	22	991.29
6	113	98	15	997.03
7	133	95	38	970.32
8	127	94	33	998.75
9	107	90	17	974.26
10	108	96	12	992.47
Variance	–	5.96	–	257.82
Std. dev.	–	2.44	–	16.06
Mean	–	94.8	–	994.93

The PSO and GA-TS were set to have 25 and 60 of population sizes respectively so that by the end of 10^{th} iteration, both of them have similar 'time take' which is around 1000s. Other than that, the remaining settings for the PSO and GA-TS in solving the CAP2 in this subsection are similar with previous, which is as shown in Tables 11 and 12. The results are shown in Tables 16 and 17.

Based on the results in Tables 16 and 17, GA-TS has better 'final best fitness' than PSO, with average value of 94.8 and 97.8, respectively. Both PSO and GA-TS achieves lowest 'final best fitness' of 91 and 90 respectively, which is very close to each other. However, for PSO, most of the 'final best fitness' values stay above 95, which is the opposite of GA-TS.

5. Discussion and conclusion

In this paper, the importance of solving channel assignment even in current fifth generation of telecommunication technology (5G) was first discussed. Then, three types of variants for genetic algorithm (GA) were presented, namely roulette wheel selection (RWS), tournament selection (TS) and stochastic universal sampling (SUS). The performances of these population selection methods were analysed by using CAP1. It was found that GA-TS is more computational power friendly than the others, thus allowing a more effective searching and eventually converged at lower minimum value. Anyhow, it was also found that the result by GA-SUS is significantly less noisy among others.

Next, the performance of GA-TS was compared with particle swarm optimization (PSO) in CAP1 and CAP2. CAP1 involved a small network but has stringent non-interference constraint, whereas CAP2 is vice-versa. In CAP1, it was found that GA-TS is more computation-friendly than PSO. However, after a longer iteration, the result showed that GA-TS tends to converge faster and ended up with larger 'final best fitness' value than PSO. Moreover, in the ten repeated runs, PSO has also showed consistency, which means significantly smaller standard deviation. Anyhow, such consistency is not repeating in CAP2, where PSO displayed large and inconsistent 'final best fitness' values.

As a conclusion, both PSO and GAs has respective pros and cons. In this case, selection of optimization method has to be based on requirements, such as allowing faster convergence rate but with acceptable interference or the other way around. Anyway, as both the PSO and GAs in this paper is introductory and for performance analysis, there are plenty of rooms for improved versions of PSO and GAs in future work.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, JSK and SLL; methodology, JSK; software, JSK, XJW and WWL; validation, SLL and YCW; formal analysis, JSK, SLL and YCW; investigation, JSK, XJW and WWL; resources, JSK and SLL; data curation, JSK; writing—original draft preparation, JSK; writing—review and editing, JSK; visualization, JSK; supervision, SLL; project administration, SLL; funding acquisition, SLL.

Acknowledgments

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM). This research was funded by Fundamental Research Grant Scheme (FRGS) by Ministry of Higher Education (MOHE) of Malaysia, grant number FRGS/2018/FKE-CERIA/F00352.

References

[1] N. U. Hasan, W. Ejaz, N. Ejaz, H. S. Kim, A. Anpalagan, and M. Jo, "Network Selection and Channel Allocation for Spectrum Sharing in 5G Heterogeneous Networks", *IEEE Access*, Vol. 4, No. 2016, pp. 980-992, 2016.

[2] M. P. Mishra and P. C. Saxena, "Survey of Channel Allocation Algorithms Research for Cellular System", *Int. J. Networks Commun.*, Vol. 2, No. 5, pp. 75-104, 2012.

[3] I. Katzela and M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication Systems - A Comprehensive Survey", *IEEE Personal Communications Magazine*, 1996.

[4] D. Corne, M. J. Oates, and G. D. Smith, *Telecom Optimisation : Heuristic and Adaptive Techniques*. John Wiley & Sons, 2000.

[5] S. Alagu and T. Meyyappan, "Efficient utilization of channels using dynamic guard channel allocation with channel borrowing strategy in handoffs", in *International Conf. CCSEA*, pp. 2658-2663, 2012.

[6] P. Shukla and S. Pancholi, "Hybrid Channel Allocation in Wireless Cellular Network", *Int. J. Commun. Netw. Secur.*, Vol. 1, No. 4, pp. 51-54, 2012.

[7] M. Iwamura, "NGMN view on 5G architecture", in *IEEE 81st Vehicular Technology Conf. (VTC Spring)*, pp. 1–5, 2015.

[8] H. Droste, G. Zimmermann, M. Stamatelatos, N. Lindqvist, O. Bulakci, J. Einchinger, V. Venkatasubramanian, U. Dotsch, and H. Tullberg, "The METIS 5G Architecture: A Summary of METIS Work on 5G Architectures," in *IEEE 81st Vehicular Technology Conf. (VTC Spring)*, pp. 1-4, 2015.

[9] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture", *IEEE Commun. Mag.*, Vol. 52, No. 11, pp. 66-75, 2014.

[10] W. Ejaz, N. ul Hasan, and H. S. Kim, "Distributed cooperative spectrum sensing in cognitive radio for ad hoc networks", *Comput. Commun.*, Vol. 36, No. 12, pp. 1341-1349, 2013.

[11] M. Ju and K. M. Kang, "Cognitive Radio Networks With Secondary Network Selection", *EEE Trans. Veh. Technol.*, Vol. 65, No. 2, pp. 966-972, 2016.

[12] S. Andreev, M. Gerasimenko, O. Galinina, Y. Koucheryavy, N. Himayat, S. P. Yeh, and S. Talwar, "Intelligent access network selection in converged multi-radio heterogeneous networks", *IEEE Wirel. Commun.*, Vol. 21, No. 6, pp. 86-96, 2014.

[13] L. Yin, K. Wu, S. Yin, S. Li, and L. M. Ni, "Reuse of GSM White Space Spectrum for Cognitive Femtocell Access", in *IEEE 18th International Conf. on Parallel and Distributed Systems*, pp. 1-7, 2012.

[14] T. Zahir, K. Arshad, A. Nakata, and K. Moessner, "Interference Management in Femtocells", *EEE Commun. Surv. Tutorials*, Vol. 15, No. 1, pp. 293-311, 2013.

[15] A. Adhikary, V. Ntranos, and G. Caire, "Cognitive femtocells: Breaking the spatial reuse barrier of cellular systems", in *Information Theory and Applications Workshop*, pp. 1–10, 2011.

[16] I. Parvez and A. I. Sarwat, "A Spectrum Sharing based Metering Infrastructure for Smart Grid Utilizing LTE and WiFi", *Adv. Sci. Technol. Eng. Syst. J.*, Vol. 4, No. 2, pp. 70-77, 2019.

[17] W. Lee, J. Kang, and J. Kang, "Joint Resource Allocation for Throughput Enhancement in Cognitive Radio Femtocell Networks", *IEEE Wirel. Commun. Lett.*, Vol. 4, No. 2, pp. 181-184, 2015.

[18] Y. S. Liang, W. H. Chung, G. K. Ni, I. Y. Chen, H. Zhang, and S. Y. Kuo, "Resource Allocation with Interference Avoidance in OFDMA Femtocell Networks", *IEEE Trans. Veh. Technol.*, Vol. 61, No. 5, pp. 2243-2255, 2012.

[19] L. Huang, G. Zhu, and X. Du, "Cognitive femtocell networks: an opportunistic spectrum access for future indoor wireless coverage", *IEEE Wirel. Commun.*, Vol. 20, No. 2, pp. 44-51, 2013.

[20] W. Roh, J. Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results", *IEEE Commun. Mag.*, Vol. 52, No. 2, pp. 106-113, 2014.

[21] B. S. Ramanjaneyulu and K. Annapurna, "Femtocell channel allocations that reduce interferences and optimize bandwidths," in *International Conf. on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 482-485, 2016.

[22] C. K. Agubor, I. Akwukwuegbu, M. Olubiwe, C. O. Nosiri, A. Ehinomen, A. A. Olukunle, S. O. Okozi, L. Ezema, and B. C. Okeke, "A Comprehensive Review on the Feasibility and Chall

enges of Millimeter Wave in Emerging 5G Mobile Communication", *Adv. Sci. Technol. Eng. Syst. J.*, Vol. 4, No. 3, pp. 138-144, 2019.

[23] G. Kendall and M. Mohamad, "Solving the Fixed Channel Assignment Problem in Cellular Communications Using An Adaptive Local Search", 2004.

[24] Y. Peng, L. Wang, and B. H. Soon, "Optimal channel assignment in cellular systems using tabu search", in *4th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, pp. 1-15, 2003,

[25] S. Latif, S. Akraam, and M. A. Saleem, "Joint Optimization of Interference and Cost in Cognitive Radio Heterogeneous Network Using Fuzzy Logic Powered Ants", *Wirel. Commun. Mob. Comput.*, Vol. 2017, pp. 1-11, 2017.

[26] W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization", *IEEE Trans. Veh. Technol.*, Vol. 45, No. 1, pp. 91-96, 1996.

[27] Y. El Morabit, F. Mrabti, and E. H. Abarkan, "Spectrum allocation using genetic algorithm in cognitive radio networks", in *3rd International Workshop on RFID And Adaptive Wireless Sensor Networks*, pp. 90-93, 2015.

[28] L. Wang, S. Li, S. C. Lay, W. H. Yu, and C. Wan, "Genetic Algorithms for Optimal Channel Assignments in Mobile Communications", in *9th International Conf. on Neural Information Processing*, pp. 1-33, 2002.

[29] S. N. Ohatkar and D. S. Bormane, "Channel Allocation Technique with Genetic Algorithm for Interference Reduction in Cellular Network", in *Annual IEEE on India Conf.*, pp. 1–6, 2013.

[30] N. U. Hasan, W. Ejaz, H. S. Kim, and J. H. Kim, "Particle swarm optimization based methodology for solving network selection problem in cognitive radio networks", in *IEEE Frontiers of Info. Technol.*, pp. 230–234, 2011.

[31] T. M. Shami, A. A. El-Saleh, and A. M. Kareem, "On the detection performance of cooperative spectrum sensing using particle swarm optimization algorithms", in *IEEE 2nd International Symposium on Telecom Technol.*, pp. 110-114, 2014.

[32] S. B. Behera and D. D. Seth, "Resource allocation for cognitive radio network using particle swarm optimization", in *IEEE 2nd International Conf. Electronics Communication Systems (ICECS)*, pp. 665-667, 2015.

[33] S. L. Loh, N. A. Ali, and S. Fam, "OPTIMIZATION OF CHANNEL ASSIGNMENT FOR MOBILE COMMUNICATION USING TABU SEARCH", *J. Adv. Manuf. Technol.*, Vol. 11, No. 1, pp. 165-178, 2018.

[34] Y. Peng, L. Wang, and B. H. Soong, "Optimal Channel Assignment in Cellular Systems using Tabu Search", in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, pp. 31-35, 2003.

[35] G. Didem, G. Denc, and C. Ersoy, "Channel Assignment Problem in Cellular Networks: A Reactive Tabu Search Approach", in *24th International Symposium on Computer and Information Sciences*, pp. 298-303, 2009.

[36] A. Zhou, B. Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art", *Swarm Evol. Comput.*, Vol. 1, No. 1, pp. 32-49, 2011.

[37] M. D. Weng, B. H. Lee, and J. M. Chen, "Two novel price-based algorithms for spectrum sharing in cognitive radio networks", in *EURASIP Journal on Wireless Communications and Networking*, pp. 265, 2013.

[38] R. A. Rashid, A. H. F. B. A. Hamid, N. Fisal, S. K. Syed-Yusof, H. Hosseini, A. Lo, and A. Farzamnia, "Efficient In-Band Spectrum Sensing Using Swarm Intelligence for Cognitive Radio Network", *Can. J. Electr. Comput. Eng.*, Vol. 38, No. 2, pp. 106-115, 2015.

[39] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", in *IEEE International Conf. on Evolutionary Computation*, pp. 69-73, 1998.

[40] M. Brad and G. David, "Genetic Algorithms, Tournament Selection, and the Effects of Noise", *Complex Syst.*, Vol. 9, No. 1, pp. 193-212, 1995.

[41] D. Kunz, "Channel assignment for cellular radio using neural networks", *IEEE Trans. Veh. Technol.*, Vol. 40, No. 1, pp. 188-193, 1991.