Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Multiuser detection: Comparative analysis of heuristic approach

M. F. Sohail [1, 2], Sheraz Alam [1, 2], Asad Hussain [1], Sajjad A. Ghauri [2, *], Mubashar Sarfraz [1], M. A. Ahmed [1]

[1]Department of Electrical Engineering, National University of Modern Languages, Islamabad, Pakistan
[2]Department of Electronics Engineering, International Islamic University, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 16 November 2016
Received in revised form
14 March 2017
Accepted 15 April 2017

Keywords:
Multiuser detection
Code division multiple access
Multiple access interference
Maximum likelihood detector
Evolutionary algorithms

ABSTRACT

Multiple users simultaneously accessing the resource give rise to interference. In such a scenario maximum likelihood multiple user detector is optimum but at the cost of increased computational complexity. Detectors based on evolutionary techniques have been discussed in this paper that evidently reduces complexity by heuristically finding the solution in the search space. We present genetic algorithm (GA), differential evolution (DE) and variants of PSO to minimize the average bit error rate (BER) against number of iterations and SNR in synchronous transmission for the discrete problem. Analysis of each algorithm is discussed separately and comparison is presented for performance analysis. The result has been generated using Monte Carlo simulations and show superiority of Soft PSO in noise prone scenarios.

1. Introduction

The principle idea behind MUD techniques is to cancel the interference produced by the other users. Multiuser detectors have the aptitude to appreciably improve the capacity and performance of a CDMA system. A specific code is assigned to every user, which is mutually orthogonal with other users. With the increase in number of users, it causes multiple access interference (MAI) which limits the capacity and performance of the system. When multiple user access the system at the same time it causes interference. This interference is the result of the random time offsets between signals, which make it impossible to design the code waveforms to be completely orthogonal (Moshavi, 1996). However, in CDMA systems multiple access interference (MAI) arises because of the non-ideal cross correlation properties of the spreading signals and multipath propagation (Ravindrababu et al., 2014). By reducing the interference it will gives the rise in capacity.

There are numerous research articles based on MUD where authors have provided solutions to various aspects such as interference and bit error rate (BER) minimization, capacity enhancement etc. using evolutionary algorithms. An analysis of different MUD strategies is given in (Moshavi, 1996).

Evolutionary algorithms are based on nature inspired computational techniques. In the last few years the literature has collected many solutions based on heuristic algorithms, particularly the evolutionary ones, for inherent problems of the multiple access communication, few of them are presented as: optimum detection problem (optimum performance) (Tan, 2001; Yen and Hanzo, 2004), sequences selection (Jeszensky and Stolfi, 1998; Kuramoto et al., 2004), parameters estimation, especially the channel coefficients estimation (Yen and Hanzo, 2001), power control problem and the rate allocation and throughput optimization (Moustafa et al., 2004), performance and capacity enhancement of CDMA communication (Ciriaco et al., 2006).

Heuristic techniques have been prominently utilized for different problems present in CDMA however lack of collective comparison analysis of these algorithms motivate us to implement and present a performance analysis for the problem of multi user detection.

We use Differential Evolution (DE), Genetic Algorithm (GA), Hard Particle Swarm Optimization (HPSO) and Soft Particle Swarm Optimization (SPSO) for discrete problems in our analysis. We have compared the results of all these evolutionary algorithms for average BER against number of iterations and SNR.

In the next section we introduce the system model; section 3 includes flowcharts and pseudo codes for the algorithms. Section 4 contains the simulation results of the evolutionary algorithms

* Corresponding Author.
Email Address: sajjad.ghauri@iiu.edu.pk (S. A. Ghauri)
https://doi.org/10.21833/ijaas.2017.06.016

and their comparison will be discussed in this section.

2. System model

There are numerous detection techniques namely minimum mean square error (MMSE) and maximum likelihood detection (MLD). Maximum-likelihood multiuser detector is finest among these detectors i.e. it offers the least probability of error equally identifying signal for all subscribers. Receiver selecting the most possible sequence of bits b_k during the period $0 \leq t \leq T_b$ is called optimal MLD (Maximum likelihood detector). Our system model is for synchronous transmission mode and given in (Proakis and Salehi, 2008).

All subscribers generate just one representation in synchronous transmission that intrudes the preferred symbol, representing the received signal as (Eq. 1):

$$r(t) = \sum_{k=1}^{K} A_K b_k g_k(t) + n(t) \quad 0 \leq t \leq Tb \tag{1}$$

This detector calculates the function of log-likelihood a (Eq. 2):

$$A(b) = \int_0^{T_b} (r(t) - \sum_{k=2}^{K} A_K b_k g_k(t))^2 \, dt \tag{2}$$

and it selects the information sequence b_k which decreases A(b). When Eq. 2 is expanded it becomes (Eq. 3):

$$A(b) = \int_0^{T_b} r^2 - 2 \sum_{k=1}^{K} A_k B_k r_k - \sum_{j=1}^{K} \sum_{k=1}^{K} A_j A_k b_j b_k \rho_{jk} \tag{3}$$

It is quite clear that the integration terms that consist of r^2 (t) are of no importance. Hence we can ignore them. So it should be better to maximize the following correlation matrix (Eq. 4)

$$C(\boldsymbol{r}, \boldsymbol{b}) = \sum_{k=1}^{K} A_k B_k r_k - \sum_{j=1}^{K} \sum_{k=1}^{K} A_j A_k b_j b_k \rho_{jk} \tag{4}$$

where, $\sum_{j=1}^{K} \sum_{k=1}^{K} A_j A_k b_j b_k \rho_{jk}$ is the multiple access interference (MAI) and received signals cross-correlation is represented by r_k (Eq. 5)

$$r_k = \int_0^{T_b} r(t) g_k(t) dt \qquad 1 \leq k \leq K \tag{5}$$

the correlation metric ρ_{jk} may be expressed as (Eqs. 6, 7, and 8)

$$C(r, b) = 2b^T r - b^T R b \tag{6}$$
$$\boldsymbol{r} = (r_1, r_2, \ldots, r_K)^T \tag{7}$$
$$\boldsymbol{b} = (A_1 b_1, A_2 b_2, \ldots, A_k b_k)^T \tag{8}$$

where, r is the received signal, b represents the symbol for kth user and correlation matrix R is expressed as (Eq. 9):

$$\mathbf{R} = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1k} \\ \vdots & \ddots & \vdots \\ \rho_{k1} & \cdots & \rho_{kk} \end{bmatrix} \tag{9}$$

3. Algorithms and flowcharts

In the beginning, we will discuss differential evolution then genetic algorithm and PSO variants. Also, to better represent the algorithms, the flow charts for genetic algorithm, particle swarm optimization and differential equation are presented in Figs. 1, 2, and 3 respectively.

Genetic Algorithm: Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Genetic algorithms (GA) steps are given below:

Initialization: Initialize parents (initial population) randomly by defining upper and lower limits.

Roulette Wheel Selection: Produce offspring using roulette wheel selection method according to probabilities of the parents, parents with greater probability have greater chance to be selected to produce offspring (Eq. 10).

$$P_i = \sum_{i=1}^{n} \frac{fitness(i)}{Sum(fitness)} \tag{10}$$

Crossover: Generate new generation from offspring's and parents (initial population) using crossover.

Mutation: Change the bit of the last 20% population if the solution is not converging.

Condition: If the condition satisfies then terminate otherwise repeat the process.

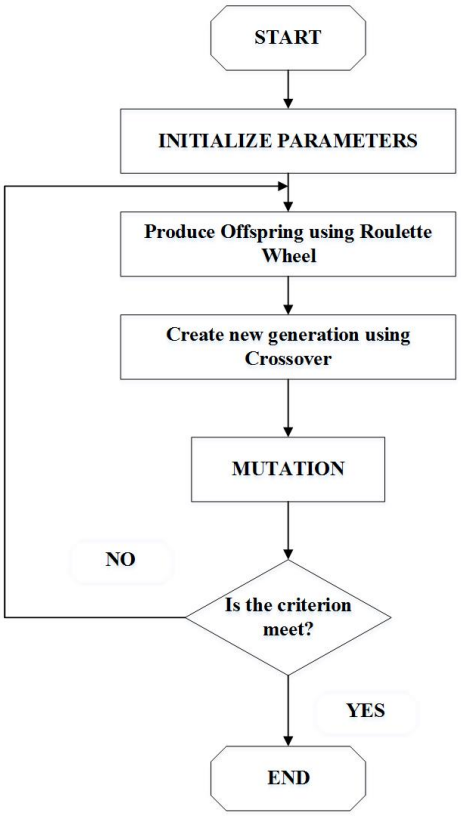

Fig. 1: Flow chart of the GA

Discrete Particle Swarm Optimization: PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals

(referred to as particles) adapts by returning stochastically toward previously successful regions (Kennedy and Eberhart, 2001). Algorithms of discrete PSO variants given below:

Initialization: Initialize the swarm and velocities of the swarm.

Update velocities: Update velocities of the swarm for hard PSO by using the Eq. 11

$$V(k,:) = V(k,:) + rand() * (Pb(k,:) - B(k,:)) + rand() * (Gb - B(k,:)) \quad (11)$$

update velocities of swarm for Soft PSO as (Eq. 12)

$$V(k,:) = V(k,:) + rand() * (1 - rand) * (Pb(k,:) - B(k,:)) + rand() * rand * (Gb - B(k,:)) \quad (12)$$

Update swarm: Update swarm for Hard PSO by using (Eq. 13)

$$S(a,b) = 1/(1 + exp(-V(a,b))) \quad (13)$$
$$If\ (S(a,b) > rand())$$
$$B(a,b) = 1$$
$$Else$$
$$B(a,b) = -1$$

update swarm for Soft PSO by using (Eq. 14)

$$S(a,b) = 1/(1 + exp(-V(a,b))) \quad (14)$$
$$If\ (S(a,b) > rand())$$
$$B(a,b) = sign(S(a,b))$$
$$Else$$
$$B(a,b) = sign(-1 + S(a,b))$$

where, S is the updating swarm, B is the initial swarm, P is the local best of the swarm and V is the velocity of the swarm.

Condition: Check the condition if the criteria meet then terminate otherwise repeat the process.

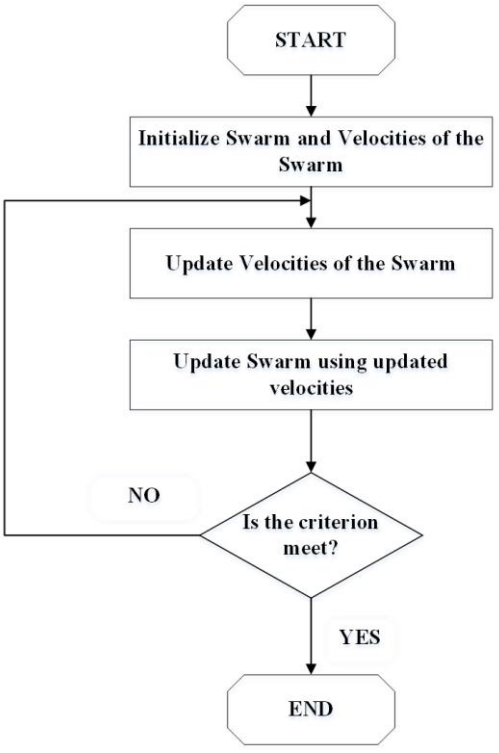

Fig. 2: Flow chart of discrete PSO

Discrete Differential Evolution: It is a stochastic, population-based optimization algorithm for solving nonlinear optimization problem. Discrete Differential Evolution (DE) algorithm steps and flow chart are given below:

Initialization: Initialize X_{m*n} randomly by defining upper and lower bound. Calculate fitness of the X.

Mutation: Randomly select x_{r1}, x_{r2} and x_{r3} from X (initial population) and all should be distinct and create a new population by using the equation given below (Eq. 15):

$$Y_{i,G+1} = x_{r1,G} + F*(x_{r2,G} - x_{r3}) \quad (15)$$

Recombination: Select the new population Z from Y and X by using the method:

$$If\ (j = Jrand\ or\ rand \leq CR)$$
$$Z(i,j) = Y(i,j)$$
$$Else$$
$$Z(i,j) = X(i,j)$$

where, CR is the convergence rate, Y is the mutated population and Z is the new population created after recombination of initial population X and the mutated population Z. Evaluate the fitness of the newly created population.

Greedy Selection: Pick up the best ones from newly created population and the initial population according to their fitness by using greedy selection.

Condition: Check if the condition meet then terminate otherwise repeat the process.

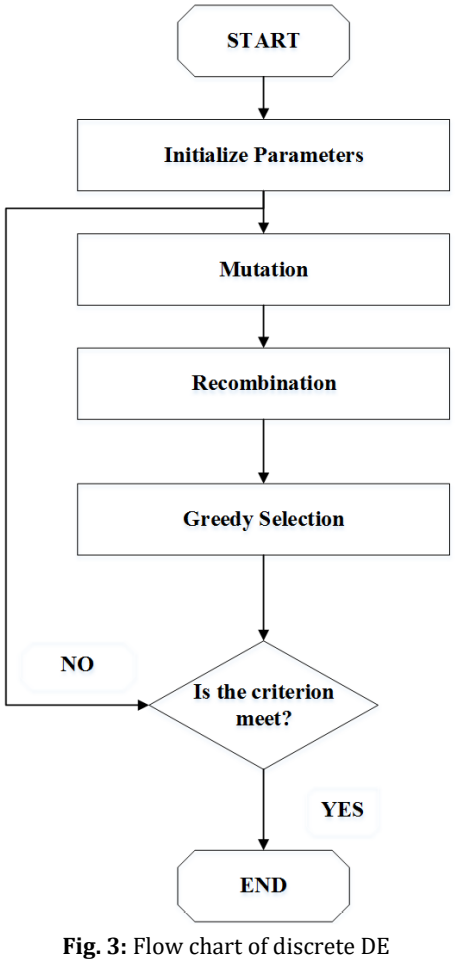

Fig. 3: Flow chart of discrete DE

4. Simulation results

In this section, we present the simulation results of the evolutionary algorithms for average BER against number of iterations and SNR. Performance of Differential Evolution (DE), Genetic Algorithm (GA), Hard Particle Swarm Optimization (HPSO) and Soft Particle Swarm Optimization (SPSO) (for discrete problems) is compared for both by changing number of iterations and SNR for the population of 30 with 20db SNR using Monte Carlo simulations.

Figs. 4 and 5 show the simulation results of the genetic algorithm (GA). The average BER starts decreasing from 3 to 0 (for iterations) and almost from 5 to 0.15 (for SNR).

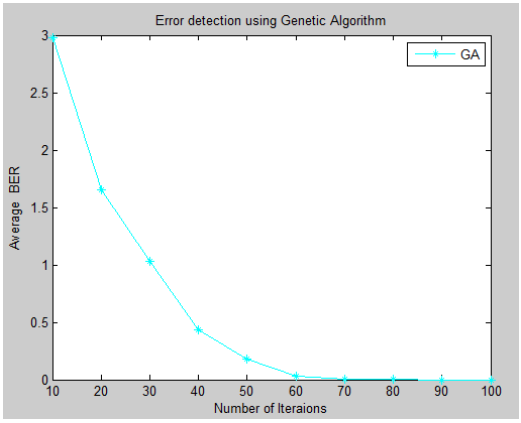

Fig. 4: Average BER against number of Iterations for Genetic Algorithm (GA)

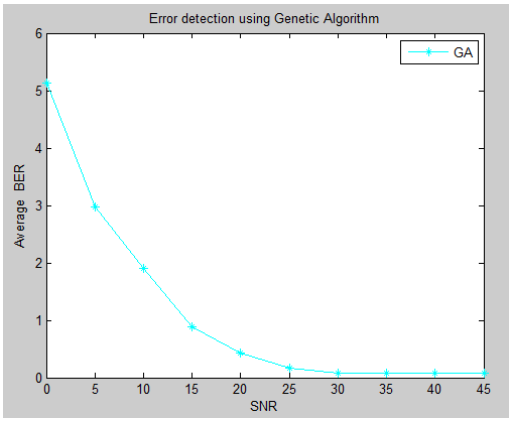

Fig. 5: Average BER against SNR for Genetic Algorithm (GA)

Figs. 6 and 7 represents the simulation results of hard particle swarm optimization (HPSO) for average BER against number of iterations and SNR. Average BER almost decreases from 2.65 to 0 (for number of iterations) and decreases from 5.2 to 0.2 (for SNR).

Figs. 8 and 9 show the simulation results of Soft PSO. Average BER decreases from 4 to 0 (for iterations) and decreases from 5.2 to 0.2 (for SNR).

Figs. 10 and 11 represent the simulation results of the differential evolution (DE). The average BER is almost starts decreasing from 3.4 to 0.1 (by varying number of iterations) and almost starts decreasing from 7.4 to 0.08 (by varying SNR).

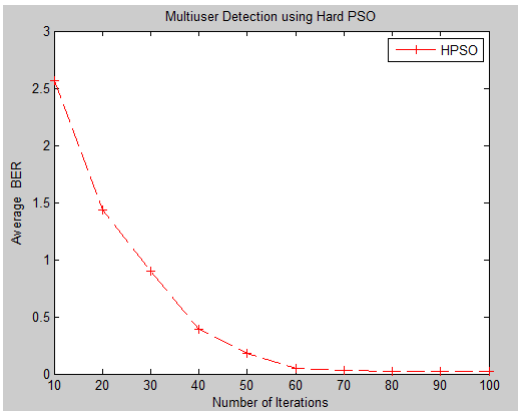

Fig. 6: Average BER for number of iterations for hard particle swarm optimization (HPSO)

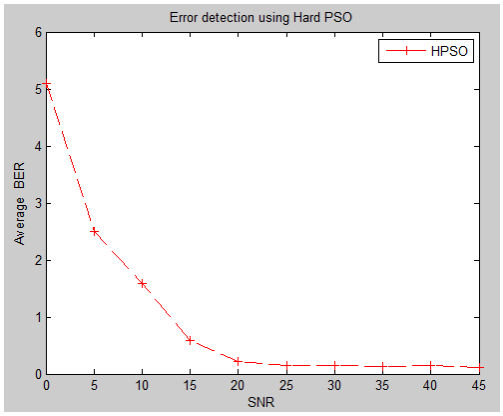

Fig. 7: Average BER by varying SNR values for hard particle swarm optimization (HPSO)

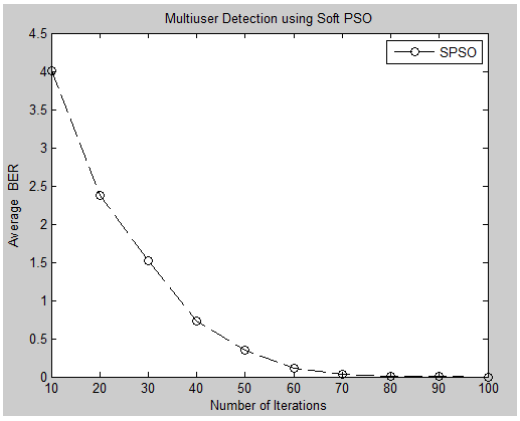

Fig. 8: Average BER against number of iterations for Soft Particle Swarm Optimization (SPSO)

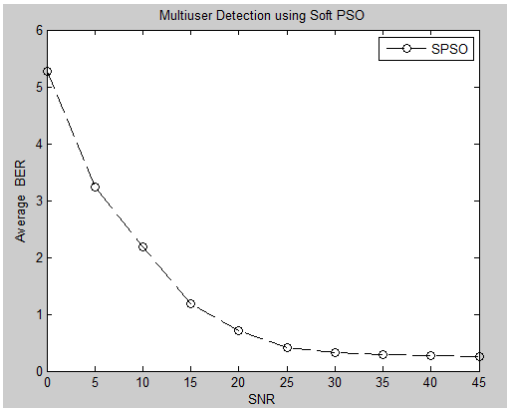

Fig. 9: Average BER against SNR for Soft Particle Swarm Optimization (SPSO)

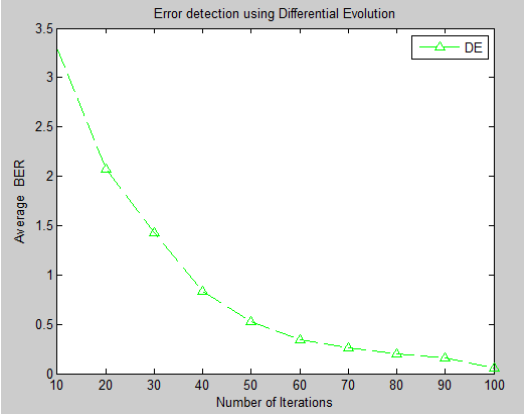

Fig. 10: Average BER against number of iterations for Differential Evolution (DE)

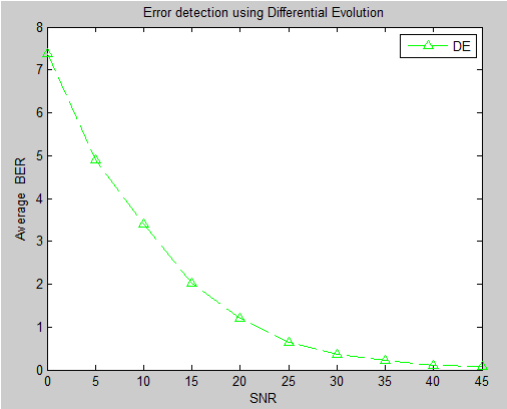

Fig. 11: Average BER against SNR for Differential Evolution (DE)

Fig. 12 shows the simulation result of performance comparison of DE, GA, HPSO and SPSO by varying number of iteration. The result shows that SPSO and GA gives the best result among all because the error decreases to zero at 60 and onward iterations. HPSO starts converging (average BER comes at 0) after 70 iterations and the same case for GA but DE gives almost 0.1 of error.

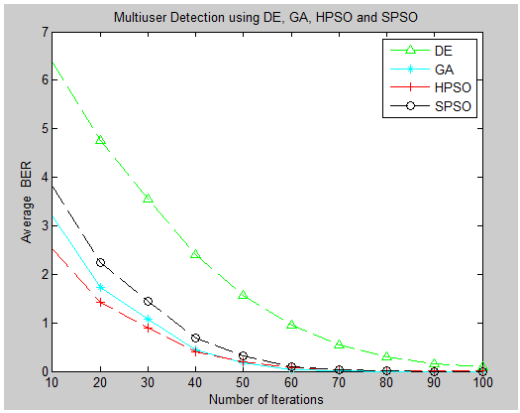

Fig. 12: Performance comparison of DE, GA, HPSO and SPSO for average BER against number of iterations

Fig. 13 shows the simulation result of performance comparison of DE, GA, HPSO and SPSO by varying SNR. By varying SNR, all the algorithms start converging and the error is in between 0 and 0.2. The model for varying SNR gives the desired results. GA gives the optimum result for increasing SNR as compared to others.

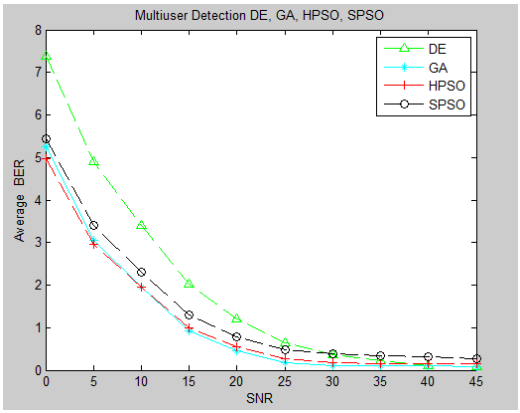

Fig. 13: Performance comparison of DE, GA, HPSO and SPSO for Average BER against SNR

5. Conclusion

On the basis of above analysis, the results show that the average BER decreases with the increase in number of cycles and SNR for all suboptimal detectors but soft particle swarm optimization (SPSO) and genetic algorithm (GA) offers the best optimum results for number of iterations as compared to others (hard PSO and DE) whereas GA converges fast for increasing SNR. We observe that these suboptimal detectors converge and achieve near optimum results with minimized computational complexity. In future, bacterial foraging optimization (BFO), firefly optimization (FFO) and cat swarm optimization (CSO) can be used to improve the results.

References

Ciriaco F, Abrão T, and Jeszensky PJE (2006). DS/CDMA multiuser detection with evolutionary algorithms. Journal of Universal Computer Science, 12(4): 450-480.

Jeszensky PJE and Stolfi G (1998). CDMA systems sequences optimization by simulated annealing. In the 5th International Symposium on Spread Spectrum Techniques and Applications, IEEE, Sun City, South Africa, 1: 38-40. https://doi.org/10.1109/ISSSTA.1998.726191

Kennedy J and Eberhart R (2001). Swarm intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Kuramoto ASR, Jeszensky PJE, and Abr˜ao T (2004). Projeto de seq¨u^encias para sistemas QS-CDMA multitaxa mpg. In the XXI Simp´osio Brasileiro de Telecomunica,c˜oes, Bel´em, PA, Brazil.

Moshavi S (1996). Multi-user detection for DS-CDMA communications. IEEE Communications Magazine, 34(10): 124-136.

Moustafa M, Habib I, and Naghshineh MN (2004). Efficient radio resource control in wireless networks. IEEE Transactions on Wireless Communications, 3(6): 2385-2395.

Proakis JG and Salehi M (2008). Digital Communication. 5th Edition, Tata McGraw-Hill India Education, New Delhi, India.

Ravindrababu J, Krishna REV, and Raja RY (2014). Interference cancellation and complexity reduction in multi stage multi-user detection. WSEAS Transactions on Communications, 13: 62-70.

Tan PH (2001). Multiuser detection in CDMA-combinatorial optimization methods. PhD Dissertation, Chalmers University of Technology.

Yen K and Hanzo L (2001). Genetic algorithm assisted joint multiuser symbol detection and fading channel estimation for synchronous CDMA systems. IEEE Journal on Selected Areas in Communications, 19(6): 985-998.

Yen K and Hanzo L (2004). Genetic-algorithm-assisted multiuser detection in asynchronous CDMA communications. IEEE Transactions on Vehicular Technology, 53(5): 1413-1422.