Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 1476 – 1483

doi:10.3906/elk-1206-108

Multiuser detection using soft particle swarm optimization along with radial basis function

Muhammad ZUBAIR[1,*], **Muhammad Aamer Saleem CHOUDHRY**[2], **Ijaz Mansoor QURESHI**[3]
[1]Department of Electronic Engineering, International Islamic University, Islamabad, Pakistan
[2]Department of Electrical Engineering, University of Lahore, Islamabad, Pakistan
[3]Department of Electrical Engineering, Air University, Islamabad, Pakistan

Received: 25.06.2012 • **Accepted:** 25.12.2012 • **Published Online:** 07.11.2014 • **Printed:** 28.11.2014

Abstract: The multiuser detection (MUD) problem was addressed as a pattern classification problem. Due to their strength in solving nonlinear separable problems, radial basis functions, aided by soft particle swarm optimization, were proposed to perform MUD for a synchronous direct sequence code division multiple access system. The proposed solution was shown to exhibit performance better than a number of other suboptimum detectors including the genetic algorithm and the classical particle swarm optimization algorithm.

Key words: Multiuser detection, radial basis functions, soft particle swarm optimization, DS-CDMA

1. Introduction

Code division multiple access (CDMA) systems offer a soft user capacity but at the same time suffer from multiple access interference (MAI), which is one of the key issues that restrict capacity. Thus, one of the major challenges in CDMA systems has been to mitigate the effects of MAI. For this, optimum as well as a number of suboptimum detectors has been proposed. The optimum detector [1] uses the principle of maximum likelihood and serves as an ideal or an upper bound for other suboptimal detectors. However, it cannot be used practically due to its enormous computational cost. Suboptimum detectors can be classified in various categories like linear, nonlinear, and evolutionary classes.

The most famous linear detectors are the decorrelating detector [2] and linear minimum mean square error (LMMSE) detector [3], which face the problem of noise enhancement and the requirement of channel estimates at the receiver. The successive interference cancellation [4] technique in the nonlinear category of detectors faces the problem of error propagation, while the performance of parallel interference cancellation (PIC) [5] heavily depends on the initial bit estimates. Researchers have also used radial basis functions (RBFs) for the multiuser detection (MUD) of CDMA systems [6]. There has been increased interest in using evolutionary techniques for the MUD problem recently. Researchers have proposed the genetic algorithm (GA) [7–9], ant colony optimization [10], evolutionary strategies [11], evolutionary programming [12] and particle swarm optimization (PSO) [13,14]. This paper is an extension of our previous contribution [14], in which the hard PSO (HPSO) along with RBF for MUD was used to carry out MUD. In this paper, we employ soft PSO (SPSO) for the same problem. The SPSO proceeds towards the optimum solution in a gradual manner, thus providing better results compared to other algorithms.

*Correspondence: muhammad.zubair@iiu.edu.pk

The remainder of the paper is ordered as follows: The system model is described in section 2. Section 3 explains the application of the PSO algorithm to the MUD problem. Section 4 contains the simulations and their results, while section 5 presents conclusions and future work.

2. System model

We consider the direct sequence CDMA (DS-CDMA) system with K users, employing a binary phase shift keying with a bit interval of T_b. The spreading code $\mathbf{g}^{(k)} = \left[g_1^k g_2^k \cdots g_N^k\right]$ for the kth user consists of N chips, with each chip having an interval of T_c. As shown in Figure 1, the composite signal of K users thus transmitting in a synchronous DS-CDMA system is given as

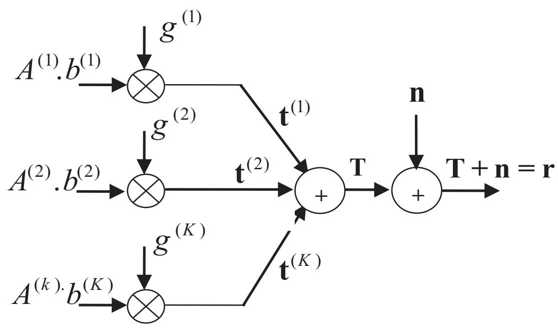

Figure 1. System model of the transmitter.

$$\mathbf{T} = \sum_{j=1}^{K} t^{(k)} = \sum_{j=1}^{K} A^{(j)} b^{(j)} \mathbf{g}^{(j)}, \tag{1}$$

where the transmitted bit has an amplitude of $A^{(k)}$. The received vector is given as:

$$\mathbf{r}\text{=}\mathbf{T}\text{+}\mathbf{n}. \tag{2}$$

The channel has been assumed to be memoryless and nondispersive Rayleigh and $\mathbf{n}$ is the noise vector of the zero mean. The channel state is assumed to be known at the receiver. Figure 2 shows the proposed detector. The soft output $\mathbf{r}$ from the matched filter is passed through tanh(.), whose output is employed for center selection using the SPSO. For the composite received signal $\mathbf{r}$ and vector center c_j, the jth RBF is given by:

$$\varphi_j(r) = \exp\left[-\frac{\|\mathbf{r} - \mathbf{c}_j\|^2}{2\sigma_j^2}\right] j = 1, 2, \cdots, 2^K, \tag{3}$$

where σ_j^2 represents the spread of the RBF. The 2^K noiseless channel output vectors serve as candidates for the RBF centers [9]. The weighted output from the RBF network is

$$\mathbf{z} = \left[z^{(1)} z^{(2)} \dots z^{(j)} \dots z^{(K)}\right],$$

where

$$z^{(j)} = \sum_{i=1}^{M} w_{i,j} \varphi_i(r) \tag{4}$$

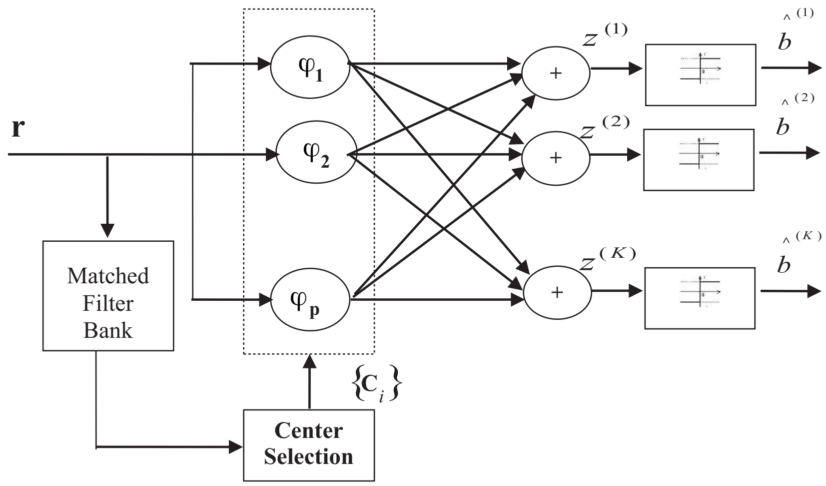

Figure 2. SPSO-RBF–assisted receiver.

or

$$\mathbf{y}{=}\mathbf{W}\Phi. \tag{5}$$

The weight $w_{i.j}$ from any ith RBF to any jth summer is either 1 or –1, following the majority decision, as follows:

$$w_{ij} = \begin{cases} 1, \; b_i^{(j)} = 1 \\ -1, \; b_i^{(j)} = -1 \end{cases}. \tag{6}$$

Owing to binary signaling, we have 2^K possible combinations of bits, and hence 2^K possible RBF centers, given as follows:

$$\mathbf{c}_i = \sum_{j=1}^{K} A^{(j)} b_i^{(j)} g^{(j)} \text{ where } j = 1, 2, \cdots 2^K. \tag{7}$$

The optimum approach requires us to employ all of the 2^K combinations, which turns out to be computationally intensive, especially when the values of K go up. This problem has been attempted previously using the GA [9] and PSO [14]. We suggest SPSO, which performs better than PSO [14].

3. HPSO and SPSO

Like any other evolutionary algorithm, the PSO algorithm operates on a population, also referred to as a swarm. In the following, we describe the steps of the SPSO as well as the difference between the hard and SPSO. There are 4 differences between HPSO and SPSO. It is important to note that the number of steps in HPSO and SPSO are the same. The 2 algorithms differ only in terms of the details of each step.

Step 1. Initialization: The first difference lies in the initialization step. In the HPSO or the classical discrete PSO algorithm, the particles are initialized at hard values of +1 and –1. The rest of the population is generated by flipping the first particle in a random manner. In the case of SPSO, the algorithm starts with soft values between +1 and –1. In order to tilt the particles towards +1 and –1, these soft values are passed through the tanh() function. The rest of the whole population is generated by adding a small random number (between 0 and 1) to random locations of the first particle. The velocity of each particle is initialized at small random values, in the case of both SPSO and HPSO.

Step 2. Fitness evaluation: The fitness of each particle is evaluated using a problem-specific fitness function.

Step 3. Update the velocity: The HPSO utilizes the following equation to update the velocity of the particles:

$$v_i^{(m)}(n) = v_i^{(m)}(n-1) + \varphi_1(p_i^{(m)} - b_i^{(m)}(n-1)) + \varphi_2(p_g^{(m)} - b_i^{(m)}(n-1)) \ . \tag{8}$$

The second term in the above equation takes into account the current position of the particle and the best position achieved so far. This is referred to as personal intelligence. The global intelligence is contributed by the third term, which involves the global best particle p_g. It is important to note that HPSO is proposed to have a fixed and equal contribution of local and global intelligence for updating the velocity of the particle. On the other hand, the SPSO updates the velocity as follows:

$$v_i^{(m)}(n) = v_i^{(m)}(n-1) + \beta\varphi_1(p_i^{(m)} - b_i^{(m)}(n-1)) + (1-\beta)\varphi_2(p_g^{(m)} - b_i^{(m)}(n-1)) \ . \tag{9}$$

Here, an adaptive approach is proposed by incorporating the dynamic parameter β, which gives more importance to local intelligence in the initial stages and more importance to collective intelligence at the later stages of the algorithm.

Step 4. Update the position: HPSO employs the following equation to update the particle position:

$$if(rand() < S(v_i^{(m)})),$$

then

$$b_i^{(m)} = -1 \tag{10}$$

else

$$b_i^{(m)} = 1$$

where

$$S(v_i^{(m)}) = \frac{1}{1+\exp(-v_i^{(m)})}. \tag{11}$$

On the other hand, SPSO proceeds as

$$if(rand() < S(v_i^{(m)})),$$

then

$$b_i^{(m)} = -1 + S(v_i^{(m)}) \tag{12}$$

else

$$b_i^{(m)} = S(v_i^{(m)}),$$

where

$$S(v_i^{(m)}) = \frac{1}{1+\exp(-\gamma v_i^{(m)})}. \tag{13}$$

The changes in Eqs. (12) and (13) make discrete PSO look like continuous PSO. The parameter γ in Eq. (13) makes $S(v_i^{(m)})$ an adaptive function. To start with, $S(v_i^{(m)})$ is kept fairly linear due to the small value for γ, and then γ keeps on increasing with the number of cycles, thus making $S(v_i^{(m)})$ more and more nonlinear, and, at the end, makes it very close to a hard sigmoid function. This is obvious in Figure 3.

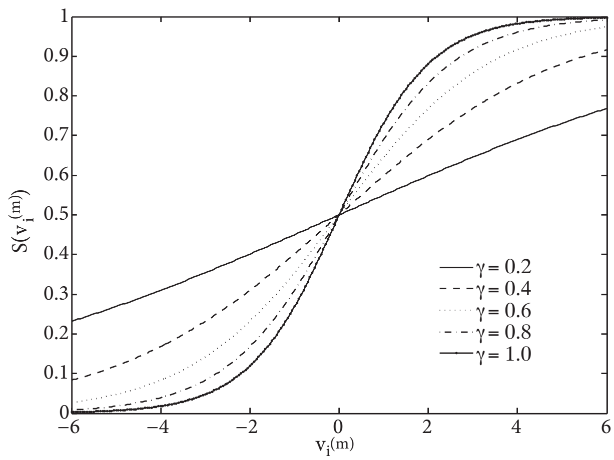

Figure 3. Effect of γ on the softness of the SPSO.

Step 5. Go back to step 2, until the stopping condition is met.

The complete algorithm is shown in the form of a flowchart in Figure 4.

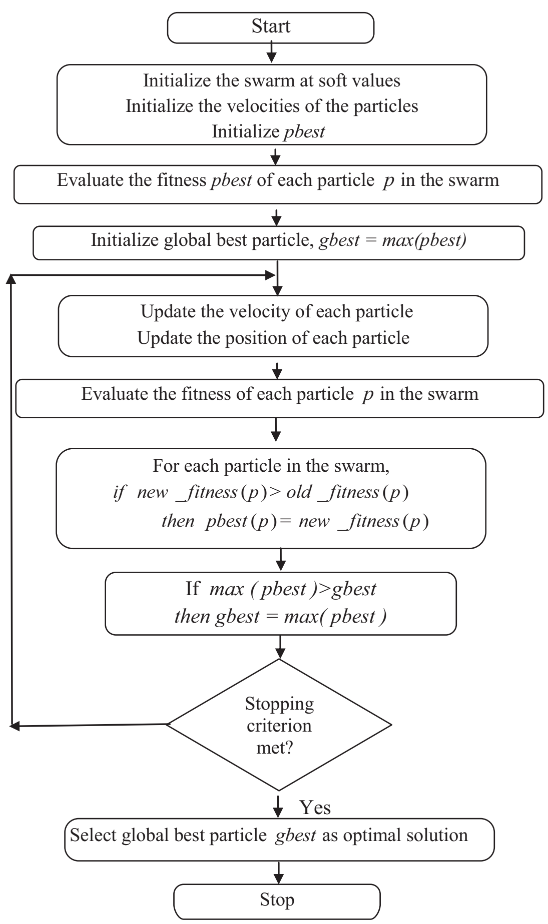

Figure 4. Flowchart of the PSO.

4. SPSO- and RBF-assisted MUD

The optimal detection requires us to maximize the objective function (Eq. (3)), which, in turn, requires the selection of optimal RBF centers. The RBF centers, closer to the vector $\mathbf{r}$, will result in optimal detection. Any jth RBF center c_j will be represented as a particle b_j in the PSO, as follows:

$$\mathbf{b}_j = \left[b_j^{(1)} b_j^{(2)}, \cdots, b_j^{(m)}, \cdots b_j^{(K)}\right], \tag{14}$$

where K is the number of users, and b_j^m is the position of the jth particle and mth user. Each particle has a corresponding velocity, given by

$$\mathbf{v}_j = \left[v_j^{(1)} v_j^{(2)}, \cdots, v_j^{(m)}, \cdots v_j^{(K)}\right]. \tag{15}$$

The best value of a particle is maintained by a particle, called personal best, and represented as $\mathbf{p}_i = \left[p_i^{(1)} \cdots p_i^{(m)} \cdots p_i^{(K)}\right]$. Similarly, the best particle of the whole swarm is stored and denoted as $\mathbf{p}_g = \left[p^{(1)} p^{(2)} \cdots p^{(m)} \cdots p^{(K)}\right]$. Both the personal and global best particles are updated. The PSO algorithm proceeds as follows:

Step 1: As described earlier, the output from the group of matched filters is passed through $\tanh(.)$ to get the first particle of the swarm. The rest of the swarm is created by randomly perturbing the values of the first particle.

Step 2: For the fitness evaluation of a particle in the swarm, we use Eq. (3) as a fitness function. The personal as well as the global best particles are selected and recorded on the basis of this fitness evaluation.

Step 3: In our previous paper, the value of β was fixed at 0.7, while in this paper, we change the value of b gradually, such that by the end of the algorithm, the global intelligence of the particles plays a dominant role compared to the individual intelligence.

$$v_i^{(m)}(n) = v_i^{(m)}(n-1) + \beta\varphi_1(p_i^{(m)} - b_i^{(m)}(n-1)) + (1-\beta)\varphi_2(p_g^{(m)} - b_i^{(m)}(n-1)) \tag{16}$$

The velocity $v_i^{(m)}$ of a particle is bounded as follows:

$$\text{if } v_i^{(m)} > V_{\max}, v_i^{(m)} = V_{\max}$$

and

$$\text{if } v_i^{(m)} < V_{\max}, v_i^{(m)} = -V_{\max}.$$

The strength of SPSO lies in the fact that as the algorithm runs the particles move towards +1 or −1 gradually, under the influence of the factor g in Eq. (13). A relatively high value of $v_i^{(m)}$ pulls a particle towards +1 and vice versa.

Step 4: The steps of the algorithm are repeated to a finite number of iterations. At the end of the algorithm, a hard decision is applied on the particle to get the estimated data bits. **Simulations and results**

The proposed detector is simulated using MATLAB. We investigate the bit error rate (BER) performance of SPSO against a number of other suboptimal detectors, like the decorrelating detector, LMMSE detector, and partial PIC detector, for systems with various numbers of users. It is worth mentioning that the optimum detector, which is based on the principle of maximum likelihood, is not practically used because of its high computational complexity, which increases exponentially with the number of users. We employ the 31-chip Gold code for spreading. The computational complexity of SPSO is close to that suggested in [9]. It is a given by product of the number of cycles and number of chromosomes. We also consider the same metric. Figure 5 shows the performance comparison of HSPO and SPSO for a system with 10 users for 2 computational complexities, i.e. 10 cycles with 10 particles and 10 cycles with 20 particles. SPSO shows better BER performance than HPSO.

Figure 6 shows the performance comparison of HPSO, GA, and SPSO for a system with 15 users. After 8 dB, SPSO takes over HPSO and is far better than the GA-based detector. The superiority of the performance of SPSO is mainly because of its faster convergence and its strength to avoid local minima. Figure 7 presents the performance comparison of SPSO with a decorrelating detector, LMMSE detector, partial PIC detector, and HPSO for a system with 20 users. Here, SPSO outperforms the competitors by a clear margin. We use 100 particles and 50 iterations for HPSO, and 100 particles and 10 iterations for SPSO. Figure 8 presents a comparison of the BER performance for a 32-user system. No graphs were available for comparison with the suboptimal detectors for systems with 32 users; therefore, the last graph presents a self-comparison of SPSO

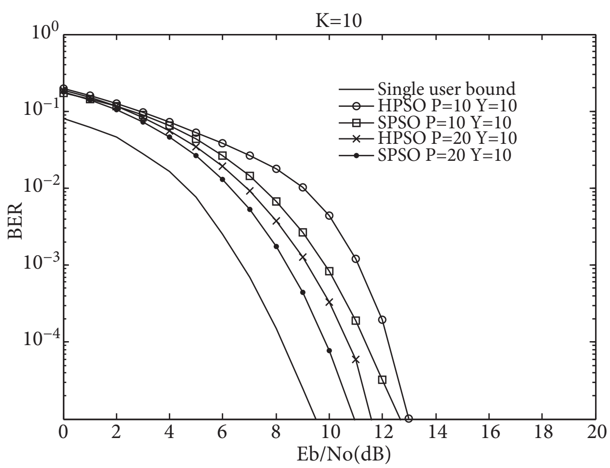

Figure 5. Performance comparison of HPSO-RBF and SPSO-RBF (K = 10).

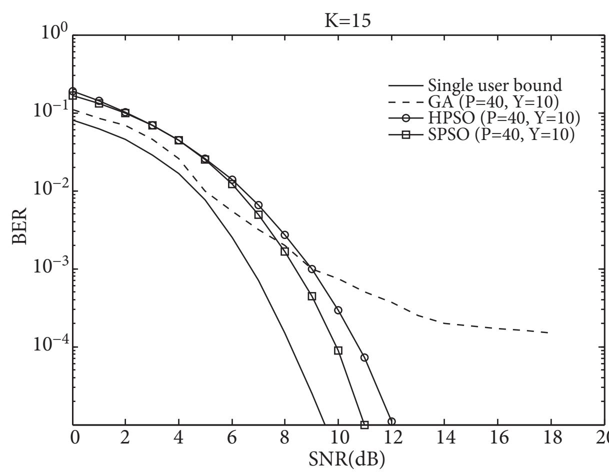

Figure 6. Performance comparison of GA-RBF, HPSO-RBF, and SPSO-RBF (K = 15).

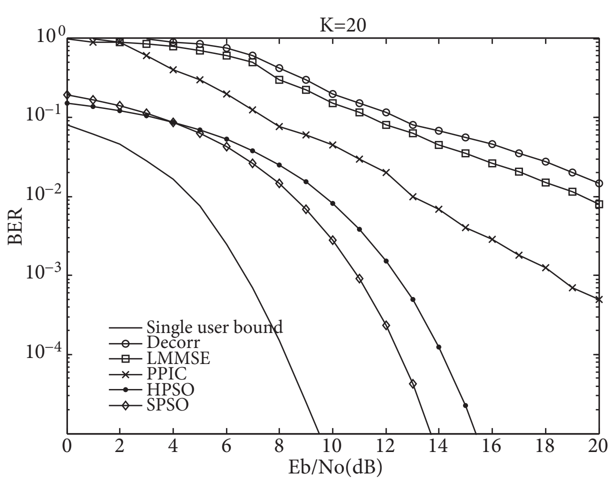

Figure 7. Performance comparison of various suboptimal detectors with the SPSO-RBF detector (K = 20).

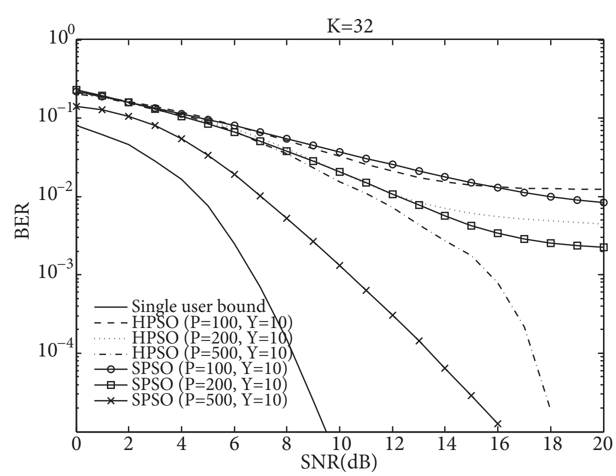

Figure 8. Performance comparison of HPSO-RBF and SPSO-RBF for a 32-user system.

for various complexities. It is obvious that as the number of particles is increased, SPSO starts taking over by a clear margin. It is important to mention that with 64 users, the optimum detector would require a huge computational requirement, while the suboptimum detector, like SPSO, does the job with a very nominal computational cost. The 4 differences highlighted in the previous section are the major reasons for the improved performance of SPSO over HPSO.

5. Conclusion

We investigated the performance of SPSO-RBF–assisted MUD for synchronous CDMA systems against a number of other suboptimal detectors. We used a nondispersive additive white Gaussian noise channel. It was seen that SPSO performs better than its hard counterpart, i.e. HPSO. This improvement in performance is mainly because of the manipulation of soft values instead of hard values. In future, we aim to explore the strength of PSO for asynchronous cases and for dispersive channels.

References

[1] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels", IEEE Transactions on Information Theory, Vol. 32, pp. 85–96, 1986.

[2] Z. Zvonar, D. Brady, "Linear multipath-decorrelating receivers for CDMA frequency-selective fading channels", IEEE Transactions on Communications, Vol. 44, pp. 650–653, 1996.

[3] R. Lupas, S. Verdú, "Linear multiuser detectors for asynchronous code-division multiple-access channels", IEEE Transactions on Information Theory, Vol. 35, pp. 123–136, 1989.

[4] P. Patel, J. Holtzman, "Analysis of a simple successive interference cancellation scheme in DS/CDMA system", IEEE Journal on Selected Areas in Communications, Vol. 12, pp. 796–807, 1994.

[5] T. Hesse, W. Schulz, "Parallel interference cancellation applied to asynchronous MC-CDMA system", Proceedings of the IEEE Vehicular Technology Conference, Vol. 2, pp. 689–693, 2002.

[6] R. Tanner, D.G.M. Cruickshank, "RBF based receivers for DS-CDMA with reduced complexity", Proceedings of the IEEE 5th International Symposium on Spread Spectrum Techniques and Applications, Vol. 2, pp. 647–651, 1998.

[7] M.J. Juntti, T. Schlosser, J.O. Lilleberg, "Genetic algorithms for multiuser detection in synchronous CDMA", Proceedings of the IEEE International Symposium on Information Theory, pp. 492, 1997.

[8] H. Wei, L. Hanzo, "Reduced-complexity near-optimum genetic algorithm assisted multiuser detection for synchronous multicarrier CDMA", Proceedings of the IEEE Vehicular Technology Conference, Vol. 3, pp. 1717–1721, 2004.

[9] H. Wei, L. Hanzo, "Reduced-complexity genetic algorithm aided and radial basis function assisted multiuser detection for synchronous CDMA", Proceedings of the European Signal Processing Conference, pp. 157–160, 2004.

[10] S.L. Hijazi, A. Best, B. Natarajan, S. Das, "Novel low-complexity DS-CDMA multiuser detector based on ant colony optimization", Proceedings of the 60th Vehicular IEEE Technology Conference, Vol. 3, pp. 1939–1943, 2000.

[11] S. Wang, Q. Zhu, L. Kang, "(1+ λ) evolution strategy method for asynchronous DS-CDMA multiuser detection", IEEE Communication Letters, Vol. 10, pp. 423–425, 2006.

[12] H.S. Lim, M.V.C. Rao, A.W.C. Tan, H.T. Chuah, "Multiuser detection for DS-CDMA systems using evolutionary programming", IEEE Communications Letters, Vol. 7, pp. 101–103, 2003.

[13] M.A.S. Choudhry, M. Zubair, A. Naveed, I.M. Qureshi, "Near optimum detector for DS-CDMA system using particle swarm optimization", IEICE Transactions on Communications, Vol. 90, pp. 3278–3282, 2007.

[14] M. Zubair, M.A.S. Choudhry, A.N. Malik, I.M. Qureshi, "Particle swarm optimization assisted multiuser detection along with radial basis function", IEICE Transactions on Communications, Vol. 90, pp. 1861–1863, 2007.