Multiuser Detection Using a Genetic Algorithm in CDMA Communications Systems

Cem Ergün and Kadri Hacioglu

***Abstract*—In this study, a hybrid approach that employs a genetic algorithm (GA) and a multistage detector (MSD) for the multiuser detection problem in code-division multiple-access communications system is proposed. Using this approach: 1) the GA is used as the first stage of the MSD to provide a good initial point for successive stages of the MSD and 2) the MSD is embedded into the GA as a "genetic operator" to improve further the fitness of the population at each generation. Such a hybridization of the GA with the MSD reduces its computational complexity by providing faster convergence. In addition, a better initial data estimate supplied by the GA improves the performance of the MSD, and the embedded MSD improves the performance of the GA. Simulation results for the synchronous and asynchronous cases are provided to show that the approach is promising.**

***Index Terms*—Code-division multiple access, genetic algorithms, multiple-access interference, multiuser detection, near–far problem.**

I. Introduction

CODE-DIVISION multiple access (CDMA) is one of several methods of multiplexing that has taken a significant role in cellular and personal communications [1]–[5]. Direct-sequence CDMA is the most popular among several types of CDMA. Here, each user's signal is assigned a different signature waveform, and the received signal is the superposition of the signals transmitted by each user.

The conventional detection approach is to pass the received signal through a bank of filters matched to the users' signature waveforms, and then decide on the information bits based on the sign of the output. Each user is detected separately considering the others as interference or noise. Due to multiple-access interference (MAI), this single-user detection strategy creates a problem, called the near–far problem: the performance drops when the power of the transmitting users are dissimilar. Thus, cancellation of MAI is necessary to increase the performance. This requires the so-called multiuser strategy [6]. Here, information of multiple users are used jointly in detecting the information related to a particular user.

Paper approved by B. Aazhang, the Editor for Spread Spectrum Networks of the IEEE Communications Society. Manuscript received January 14, 1998; revised March 8, 1999. This paper was presented in part at the Ninth Mediterranean Electrochemical Conference (Melecon'98), Tel-Aviv, Israel, May 1998, and in part at the Third European Mobile Communications Conference (EPMCC'99), Paris, France, March 1999.

C. Ergün is with the Department of Electrical and Electronics Engineering, Eastern Mediterranean University, Mersin 10, Turkey.

K. Hacioglu is with the Center for Spoken Language Research, University of Colorado, Boulder, CO 80309-0594 USA, on leave from the Department of Electrical and Electronics Engineering, Eastern Mediterranean University, Mersin 10, Turkey (e-mail: hacioglu@glottis.colorado.edu).

Publisher Item Identifier S 0090-6778(00)07091-4.

The study of the optimum multiuser detection has shown that while significantly superior performance over the CD is possible, it can be obtained only with a marked increase in computational complexity, which is exponential in the number of users [6]. Since CDMA systems could potentially have a large number of users, this solution may prove, in a number of situations, to be impractical and too expensive to implement. Therefore, research efforts have recently concentrated on the development of suboptimal receivers that are near–far resistant, have reasonable computational complexity, and their performance is comparable to that of the optimum receiver. Two types of receiver whose complexity is proportional to the number of users have been mainly considered. One is the decorrelating detector [7], [8] and the other is the multistage detector (MSD) [9]. The decorrelating detector is linear and near–far resistant. It provides significant improvement over the CD at conditions where the MAI is dominant, without requiring knowledge of the user powers. On the other hand, the MSD is a nonlinear detector that might achieve near-optimal performance in some cases. In contrast to the decorrelating detector, it requires exact knowledge of the user powers. It relies on improving each stage's estimate by subtracting the estimate of the MAI obtained by the previous stage.

The decorrelating detector suffers from noise enhancement and the MSD performance depends heavily on the initial estimate, which is commonly provided either by the conventional detector (CD) or the decorrelating detector. Thus, under several conditions, their performances might fall short of the optimum performance motivating research for other near–far resistant and near-optimal methodologies.

In this paper, we consider the multiuser detection from a combinatorial optimization viewpoint. Driven by demand for algorithms with significantly lower computational complexity than the optimum algorithm but slightly poorer performance, not to mention curiosity, we propose a genetic algorithm (GA) and a GA hybridized with the MSD to seek the optimal solution efficiently (for the sake of lower computational complexity). It is worth mentioning that the MSD can be viewed as an optimizer that searches for a better solution over one dimension at a time instead of over all dimensions, as in the optimal detector (OD).

The GA multiuser detector operates on a population of potential solutions (called generation) and uses the likelihood function as the objective function to produce better and better approximations to a solution. At each iteration, a new set of possible solutions is created by selecting individuals based on the fitness level and applying the so-called genetic operators; crossover, and mutation [10]. The computation time is quite long, but GA's can be easily and effectively parallelized. Direct

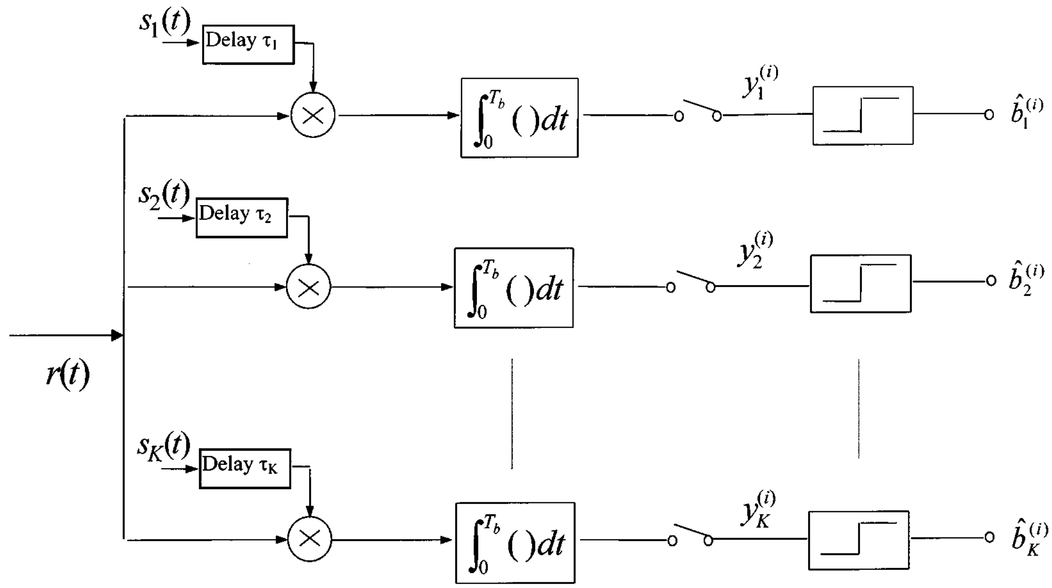


Fig. 1. Conventional multiuser CDMA detector.

application of GA, i.e., full power (or pure) GA, requires fine tuning of its related parameters, and extensive computation, yet as a multiuser detector it might have lower computational complexity/bit than the optimum detector based either on the exhaustive search in the synchronous case or the Viterbi algorithm (VA) in the asynchronous case, but at slightly lower performance. As will be shown, the computational complexity per bit decision of the GA is order of K^2, for large enough K, whereas the optimum detector has a computational complexity on the order of 2^K in either case, where K is the number of the users in the system. However, it seems most likely that in order to be more useful in the context of CDMA communications, the GA must be hybridized with some other technique to hopefully find a good solution quickly. For this reason, a radically different approach of using combination of the GA with the MSD is introduced. Here: 1) the GA is used as the first stage of MSD to provide a good initial point and let the rest be done by successive stages and 2) the MSD is used as a genetic operator to further improve the average fitness of the population. So, a lower complexity GA (with a smaller population size and number of generations) can be employed in both cases. This, in turn, reduces the sensitivity of the GA to parameter selection, such as the crossover and mutation probabilities, etc.

The paper is organized as follows. The general CDMA communication model and the conventional, optimum, decorrelating, and MSD's are presented in Section II. In Section III, the basic GA is explained and the detectors, which employ the hybrid GA/MSD approach are presented. Section IV compares the computational complexity of the detectors. Computer simulation results are presented in Section V. Finally, in Section VI, concluding remarks are made.

II. Multiuser CDMA System Model

Let us assume that there are K users in the system and each user has its own signature waveform, and each employs the binary phase-shift keying modulation. The baseband received signal can be written as

$$r(t) = S(t, \mathbf{b}) + n(t) \tag{1}$$

where

$$S(t, \mathbf{b}) = \sum_{i=1}^{P} \sum_{k=1}^{K} b_k^{(i)} s_k \left(t - iT_b - \tau_k\right) \tag{2}$$

$\mathbf{b} = [[b_1^{(1)}, b_2^{(1)}, \ldots, b_K^{(1)}], [b_1^{(2)}, b_2^{(2)}, \ldots, b_K^{(2)}], \ldots, [b_1^{(P)}, b_2^{(P)}, \ldots, b_K^{(P)}]]^T$, $b_k^{(i)} \in \{-1, 1\}$ is the ith transmitted bit of the kth user, $s_k(t)$ is the signature waveform of the kth user, T_b is the bit interval duration, $\tau_k \in [0, T_b)$ is the time delay of the kth user, P is the packet size, and $n(t)$ is the white Gaussian noise with two-sided power spectral density $N_o/2$. For simplicity and without loss of generality, we suppose that the users are numbered such that $0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_k < T_b$. The signature waveform is time limited to $[0, T_b]$ and it can be expressed as

$$s_k(t) = \sum_{l=0}^{L-1} \sqrt{e_k} a_l^{(k)} p_{T_c} \left(t - lT_c\right) \tag{3}$$

where e_k is the energy of the kth user, $p_{T_c}(t)$ is the unit rectangular pulse of chip duration T_c, and $\{a_l^{(k)} \in \{-1, 1\}\}$ is the lth bit of the kth user's spreading sequence. Note that the length of the signature waveform is $L = T_b/T_c$. In the following, the energies and time delays of all users are assumed to be known at the receiver.

A. Conventional Detector

The CD consists of a bank of filters matched to the signature waveforms of each user, and threshold devices that produce the estimate of transmitted bits as shown in Fig. 1. Mathematically

$$\hat{\mathbf{b}}_{\text{CD}}^{(i)} = \text{sign}\left(\mathbf{y}^{(i)}\right), \qquad 1 \le i \le P \tag{4}$$

where $\mathbf{y}^{(i)} = [y_1^{(i)} \quad y_2^{(i)} \quad \cdots \quad y_K^{(i)}]$, $\hat{\mathbf{b}}_{\text{CD}}^{(i)} = [\hat{b}_1^{(i)} \quad \hat{b}_2^{(i)} \quad \cdots \quad \hat{b}_K^{(i)}]$, and

$$y_k^{(i)} = \int_{iT_b+\tau_k}^{(i+1)T_b+\tau_k} r(t) s_k(t - iT_b - \tau_k)\, dt. \tag{5}$$

In matrix form

$$\mathbf{y}^{(i)} = \mathbf{H}(+1)\mathbf{b}^{(i-1)} + \mathbf{H}(0)\mathbf{b}^{(i)} + \mathbf{H}(-1)\mathbf{b}^{(i+1)} + \mathbf{n}^{(i)} \tag{6}$$

and

$$\mathbf{y} = \mathbf{Hb} + \mathbf{n} \tag{7}$$

where $\mathbf{y} = [[y_1^{(1)}, y_2^{(1)}, \ldots, y_K^{(1)}], [y_1^{(2)}, y_2^{(2)}, \ldots, y_K^{(2)}], \ldots, [y_1^{(P)}, y_2^{(P)}, \ldots, y_K^{(P)}]]^T$. The elements of the matrix $\mathbf{H}(v) \in \Re^{K\times K}$ are given by

$$h_{kl}(v) = \int_{\tau_k}^{T_b+\tau_k} s_k(t-\tau_k)\, s_l(t + vT_b - \tau_l)\, dt, \quad 1 \le k, l \le K. \tag{8}$$

Since the signature waveforms are zero outside the interval $[0, T_b]$, the following two conditions are satisfied:

$$h_{kl}(v) = 0 \qquad \forall |v| > 1 \tag{9}$$

$$h_{kl}(-v) = h_{kl}^T(v). \tag{10}$$

Therefore, the matrix $\mathbf{H} \in \Re^{PK\times PK}$ is shown at the bottom of the page.

In a synchronous scenario, for which $\tau_i = 0, \ \forall i$ and $P = 1$, $\mathbf{H} = \mathbf{H}(0)$.

B. Optimum Multiuser Detector (OMD)

The OMD selects the data sequence that minimizes [6]

$$\Lambda(\mathbf{b}) = \int_{T_b+\tau_1}^{PT_b+\tau_K} [r(t) - \mathbf{S}(t, \mathbf{b})]^2\, dt. \tag{11}$$

The minimization of (11) results in the following detection:

$$\hat{\mathbf{b}}_{\text{OMD}} = \arg\left\{ \max_{\mathbf{b}\in\{-1,+1\}^{PK}} \left(2\mathbf{y}^T\mathbf{b} - \mathbf{b}^T\mathbf{Hb}\right)\right\}. \tag{12}$$

Note that (12) dictates a search over the 2^{PK} possible combinations of the components of the vector b, if not implemented using the VA or sequential search techniques [6]. It is clear that the computational complexity increases with the number of users.

C. Decorrelating Detector

The decorrelating detector applies the inverse of the correlation matrix $\mathbf{H}^{-1}$ to the matched filter output in order to decouple the received signal. From (4), the result is

$$\hat{\mathbf{b}}_{\text{dec}} = \text{sign}\left[\mathbf{H}^{-1}\mathbf{y}\right] = \text{sign}\left(\mathbf{b} + \mathbf{n}_c\right) \tag{13}$$

which is the sign of decoupled data plus a noise term. It can be seen that (13) does not depend on the power of the users. Thus, it is optimal in the sense of near–far resistance, but not optimal in the sense of minimum bit-error rate (BER) since additive noise is colored with autocorrelation $N_o\mathbf{H}^{-1}$.

D. MSD

The MSD is one of several suboptimal methods that have been developed over the last decade [9]. The MSD utilizes previously made decisions of other users to cancel interference present in the signal of the desired user. The $(m+1)$th stage of this detector uses decisions of the mth stage to cancel the MAI present in the received signal. The MSD use the following instead of (12):

$$\hat{\mathbf{b}}_k^{(i)}(m+1) = \arg\left\{ \max_{\substack{b_k^{(i)}\in\{-1,+1\} \\ b_l^{(i)}=\hat{b}_l^{(i)}(m),\, l\neq k}} \left[2\mathbf{y}^T\mathbf{b} - \mathbf{b}^T\mathbf{Hb}\right]\right\}. \tag{14}$$

Note that the maximization is over one bit instead of all bits as in (11). It can be easily shown that the $(m+1)$th stage estimate of the ith transmitted vector $\hat{\mathbf{b}}^{(i)}$, in matrix form, is [9]

$$\hat{\mathbf{b}}^{(i)}(m+1) = \text{sign}\left[\mathbf{z}^{(i)}(m)\right], \qquad m \ge 1 \tag{15}$$

where

$$\mathbf{z}^{(i)}(m) = \Big[\mathbf{y}^{(i)} - \Big[\mathbf{H}(+1)\hat{\mathbf{b}}^{(i-1)}(m) + [\mathbf{H}(0) - \mathbf{E}]\hat{\mathbf{b}}^{(i)}(m) + \mathbf{H}(-1)\hat{\mathbf{b}}^{(i+1)}(m)\Big]\Big] \tag{16}$$

and the energy matrix $\mathbf{E}$ be defined such that $\mathbf{E} = \text{diag}([e_1, e_2, \ldots, e_K])$. The performance of the MSD depends heavily on the initial data estimate $(\hat{\mathbf{b}}^{(i)}(1))$. The initial data estimate can be the output of the CD as

$$\hat{\mathbf{b}}^{(i)}(1) = \text{sign}\left[\mathbf{y}^{(i)}\right] \tag{17}$$

or can be the output of the decorrelator detector as

$$\hat{\mathbf{b}}^{(i)}(1) = \mathbf{b}_{\text{dec}}^{(i)}. \tag{18}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}(0) & \mathbf{H}(-1) & 0 & \cdots & 0 & 0 \\ \mathbf{H}(+1) & \mathbf{H}(0) & \mathbf{H}(-1) & & \cdot & 0 \\ 0 & \mathbf{H}(+1) & \mathbf{H}(0) & & \vdots & \\ \vdots & \cdot & \cdot & \vdots & \vdots & \vdots \\ \vdots & & & & 0 & \\ 0 & & & \mathbf{H}(+1) & \mathbf{H}(0) & \mathbf{H}(-1) \\ 0 & 0 & & 0 & \mathbf{H}(+1) & \mathbf{H}(0) \end{bmatrix}$$

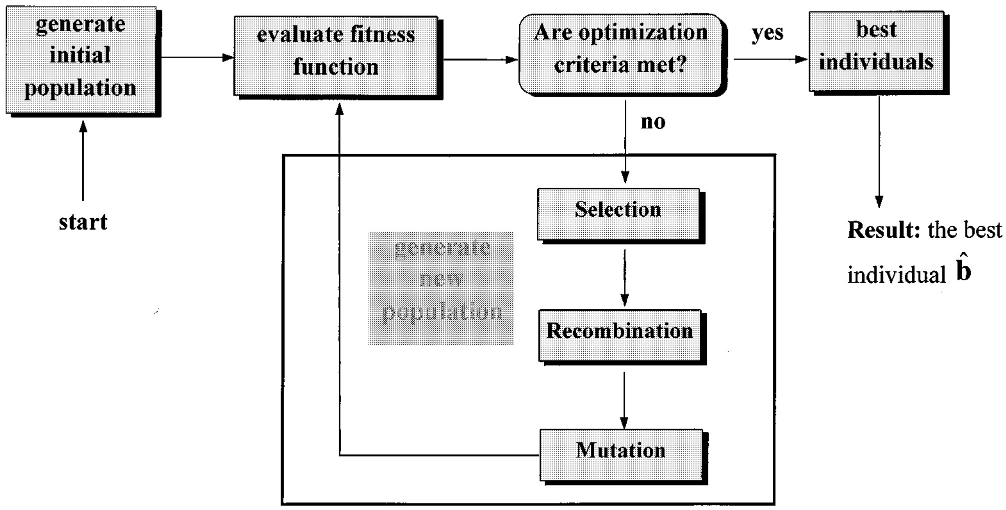


Fig. 2. Simple GA procedure.

Thus, the selection of the first stage is very important. The decorrelating detector has been found as a good choice for the first stage rather than the CD [11].

III. Genetic Algorithm Based Multiuser Detection

A. Basic GA

The GA is a stochastic search method that mimics the metaphor of natural biological evaluation and is an efficient tool in searching for the global optimum [10]. Fig. 2 shows the structure of a simple GA.

An implementation of a GA begins with a population of random chromosomes which can be represented by binary strings. This population is generally of fixed size N_p which does not change over time. Nevertheless, the population size and chromosome length are problem dependent. The members of the population are usually strings of symbols that represent possible solutions to the problem to be solved. Each member of the population at each generation is evaluated, and according to its fitness value, it is assigned a probability to be selected for reproduction. After selection, genetic recombination (crossover) is applied to pairs of parents to create offsprings, which will be mutated through the mutation operator and then inserted into a new population, forming the next generation of individuals.

The crossover operates on two selected members of the population and combines their respective chromosomes to create offsprings. The mutation operator selects a member of the population and changes some part of its chromosome. The members of the populations with the worst fitness values are replaced by the new individuals. The algorithm continues until good results are obtained in terms of the objective function.

Summarizing, the following issues must be carefully addressed for the problem under consideration:

- representation of solutions as chromosomes;
- initialization of the population;
- determination of the fitness measure;
- selection of genetic operators;
- adjustment of GA parameters (population size, crossover and mutation probabilities, number of generations, etc.).

B. Representation of the Problem

Each chromosome in the population is a trial solution to the problem. The chromosome is a string of variables. Binary encoding of variables, in which each variable takes one of two possible values, is the most common in GA applications due to its mathematical tractability and ease of implementation. In multiuser detection, the solution vector $\hat{\mathbf{b}}$ is already in binary form ($+1$ and -1) and, thus, requires no effort for encoding.

C. Initialization of the Population

In many GA applications, it is common to select the initial population randomly from the solution space. In this study, a different approach is adopted. The CD output is taken as the input chromosome, and the initial population is generated by perturbing the input chromosome. The population size N_p is kept constant throughout the optimization.

D. Fitness Function

The goal of the fitness function is to evaluate the status of each chromosome. The fitness function is required to be nonnegative. In the multiuser detection, the objective is the maximization of the cost (or objective) function

$$c(\mathbf{b}) = 2\mathbf{y}^T\mathbf{b} - \mathbf{b}^T\mathbf{H}\mathbf{b}. \tag{19}$$

The objective function $c(\mathbf{b})$ is not guaranteed to be nonnegative in all cases. For this reason, it is necessary to map the objective value to a fitness value. The following objective-to-fitness transformation is used [12]:

$$f(\mathbf{b}) = K + (c(\mathbf{b}) - c_w)\,. \tag{20}$$

Here, c_w is the objective value of the worst chromosome in the current population and K is a positive constant.

E. Genetic Operators

1) Parent Selection: Selection is an operator which uses the fitness value to select the fittest chromosome. It emulates the survival of the fittest mechanism in nature. Most commonly

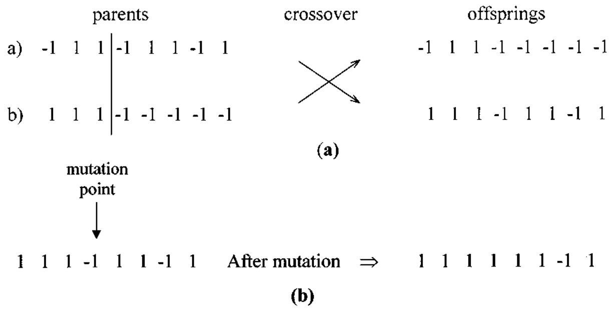


Fig. 3. (a) Single point crossover in the GA's. (b) Bit mutation on the fourth bit of the old chromosome.

used operators among several are the proportionate and tournament schemes. An example of the proportionate scheme is the roulette wheel selection, which operates as follows [13]:

1) sum the fitness values of all population members

$$f_T = \sum_{i=1}^{N_p} f_i \tag{21}$$

2) generate a random number n uniformly distributed on $(0, f_T)$;
3) select the kth member which satisfies

$$\sum_{i=1}^{k} f_i \geq n. \tag{22}$$

Here, the chromosomes with the largest fitness scores are selected one or more times while the chromosomes with low fitness scores are removed for the next generation.

2) Crossover Operator: The crossover operator basically creates two offsprings by combining subparts of the bit strings of two selected (as mentioned above) members (called parents) of the population. There are various ways of implementing the crossover operator. The simplest form is the one-point crossover. The crossover point is randomly selected along the chromosome and portions up to that point are exchanged between two parents. This scheme illustrated in Fig. 3(a). The probability of crossover p_c is user controlled and usually set to a high value (e.g., 0.9). If the crossover is not allowed, the parents are placed into the next generation unchanged.

3) Mutation Operator: The mutation operator simply alters each bit of the bitstring randomly with an user controlled probability p_m. In contrast to the crossover, the probability of the mutation is set to a small value (e.g., 0.01). An example of the mutation on the fourth bit of the second offspring is shown in Fig. 3(b).

F. Setting GA Parameters

The optimal setting of GA parameters is strongly dependent on the problem. The parameters to be set are

- population size N_p;
- selection mechanism;
- type of crossover;
- crossover probability p_c.
- mutation probability p_m.
- replacement strategy.

Although there is no universally accepted guide for parameter selection, there are several choices available. The best, but time-consuming way, is to try each possibility and choose the one that performs the best. However, the following choices are not uncommon in many GA implementations:

- to vary the population size from a relatively small value to a relatively large one;
- to use the roulette-wheel selection method;
- to employ the one-point crossover operator with p_c close to 1;
- to apply the mutation operator to all bits with p_m close to 0;
- to use the elitist policy for replacement.

The most challenging issue is to fix the population size. Indeed, there is no fixed optimal value. Its choice is highly problem dependent and one must consider several factors. A GA with a relatively large population size might converge fast with high probability of finding the global optimum, but at an extremely large computational complexity. On the other hand, a GA with relatively small population size has reasonable computational complexity but less chance to catch the global optimum. Besides, a premature convergence to a very poor solution is very common. Here, the tradeoff is between the computational complexity and probability of finding the global optimum. The popularity of the aforementioned genetic operators is due to their simplicity. In the elitist policy, the fittest member of the present population is kept in the next generation. This assures that the best solution obtained to date cannot be lost due to an unfortunate crossover or mutation operations.

G. GA with MSD

In this study, the following three structures are considered:

a) GA only;
b) GA followed by MSD;
c) MSD embedded in GA.

1) Pure GA Multiuser Detector: In this detector, the basic GA algorithm is used, as detailed in the preceding sections. The best chromosome in the population after convergence is selected as the decision $\hat{\mathbf{b}}_{\text{GEN}}$ (see Fig. 2). In the following, for the sake of clarity, we summed up how the basic GA operates as a multiuser detector.

The initial population (of size N_p) is created using the decision provided by the CD. The CD bits are randomly flipped (or not) from $+1$ to -1, or vice versa, until N_p solution candidates are created. Each member of the population is then evaluated using the fitness function derived from the maximum likelihood function. This is followed by the roulette wheel selection to select pairs for the crossover and mutation. All the members of the present population but the fittest (elitism) are deleted to make room for the new members which are created after the crossover and mutation operations. Using the crossover operator we hope that the (frequent) exchange of substrings of good solutions will create a new solution with a better performance. Using the mutation we (rarely) extend the research into previously unexplored areas. The cycle (selection, crossover/mutation, replacement) is repeated until an optimization criterion is met. The maximum number of iterations (or generations), convergence or reaching

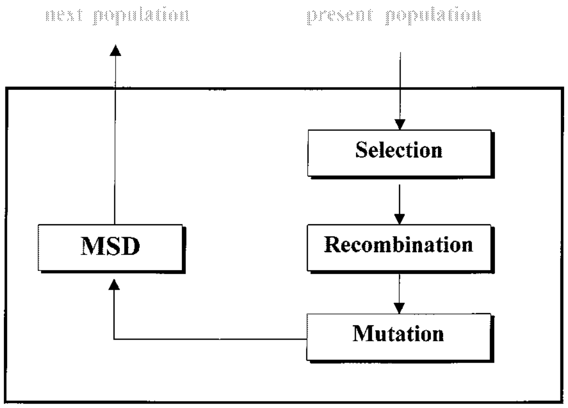


Fig. 4. MSD as a genetic operator.

an acceptable fitness level (if known) can be given as examples of optimization criteria. The former is adopted in this study.

2) Cascaded GA/MSD Multiuser Detector: In this detector, the outcome of the GA algorithm is used as the initial estimate of the MSD. That is, $\hat{\mathbf{b}}(1) = \hat{\mathbf{b}}_{\mathrm{GEN}}$. As is well known, the initial guess influences the MSD much, and the MSD with initial decisions from the decorrelating detector performs better than the MSD with initial decisions from the CD. So, we expect the MSD to perform better if GA provides initial estimates better than those provided by the CD and decorrelating detector. It is worth mentioning that the GA itself is capable of finding a solution that cannot be further improved by the MSD. So, why should we use the MSD after the GA? Indeed, the pure GA can be used in cases where its slow convergence, high computational complexity, and fine tuning are acceptable. Otherwise, the MSD would be helpful to set a small size population and run the GA for a few number of generations without any fine tuning. Then, the unidimensional search by the MSD would improve the premature candidate provided by that low complexity GA. We provide simulation results that support this.

3) Embedded GA/MSD Multiuser Detector: In this section, instead of applying the MSD to the fittest member of the population after a number of generations, we propose to apply the MSD at each generation to all or a portion of the members of the population. This suggests the use of the MSD as a "genetic operator." This embedding of the MSD in to the GA is illustrated in Fig. 4. The basic GA is fairly fast to locate the appropriate region(s) for a good solution in the search space, but fairly slow to fine tune the optimum solution using the genetic operators that are random in nature. However, in the proposed structure the MSD provides a systematic, local, and computationally efficient unidimensional search that would significantly reduce the time wasted by the GA searching randomly along incorrect directions, particularly for relatively large search spaces. Summarizing, we expect this approach to overcome the problems of GA (slow convergence, high computational complexity, and fine tuning of parameters) and MSD (poor initial estimates).

IV. Computational Complexity of the Detectors

In this section, first we consider (12) in the context of combinatorial optimization. We then define the computational complexity in terms of the number of users K. Finally, we determine the computational complexities of the detectors.

The optimal multiuser detection, as mentioned in (12), is NP-hard [6]. That is, unfortunately, there exists no algorithm that exactly solves the problem in polynomial time. The dimension of the search space is PK and the size (the number of possible solutions) is 2^{PK}. The straightforward approach is to evaluate (12) for each possible solution and select the one that gives the maximum. Nevertheless, this exhaustive search is impractical for typical numbers of users and packet sizes. There exists another algorithm based on dynamic programming (DP) that provides the exact solution [6]. This DP algorithm, also known as the VA, has a computational complexity independent of P. The implementation of this algorithm is also out of question even for moderate values of K, otherwise the problem would not be NP-hard. So, we need approximate (or heuristic) algorithms that provide quite fast and sufficiently good solutions. The GA can be considered as an optimization algorithm that meets this requirement [15]. One may also consider the MSD as an approximate optimization algorithm based on local search. A local search algorithm starts from an initial solution and replaces it with a better solution in its properly defined neighborhood until no better solution is found in the neighborhood. The simplicity brings the risk of being trapped at a local optimum solution with a relatively worse performance. That is why the MSD performance strongly depends on the choice of an initial solution. In this context, the proposed detectors (or optimization algorithms), namely cascade GA and MSD detector (C-GA/MSD) and embedded GA and MSD detector (E-GA/MSD), can be viewed as the GA with local search. Intuitively, the GA with local search will be more effective than the GA itself. Put differently, the GA with local search is expected to yield a good solution in much shorter time.

Let us now consider the computational complexities. We say that the computational complexity/bit is $O(g(K))$, for large enough K, if there exists a constant, say $c > 0$, such that the computational complexity/bit of $C(K) \leq cg(K)$. As was shown in [6], each $\mathbf{b}$ in (12) defines a path through a trellis of length PK with 2^{K-1} states, and each state has two incoming and outgoing paths. Thus, the objective value of a particular $\mathbf{b}$ is the summation of PK path metrics. The VA computes $2 \times 2^{K-1}$ path metrics for each received bit. Thus, the VA has a computational complexity/bit of $O(2^K)$ [6]. On the other hand, the GA has a population size N_p and runs for N_g generations. Since each member requires PK path metric computations to determine its fitness value, a total of N_gN_pPK computations are performed. The computational complexity/bit is then $O(N_gN_p)$. Additional operations required by the GA are the generation of random numbers (to realize the desired probabilities), exchange of bit strings (crossover), and inversion of bits (mutation). For each generation, we generate N_p random numbers for selection, $N_p/2$ for crossover, and N_pPK for mutation. The average number of bit-string exchanges is $p_cN_p/2$, and the average number of bit inversions is p_mN_pPK. To obtain the corresponding computational complexity/bit, we add all, multiply by N_g, and divide by PK; the result is $O(N_gN_p)$. It is worth mentioning that N_p and N_g implicitly depend on K and P. The more complex the search space is, the larger the population size and the number of generations should be. Although the choice of N_g and N_p is strongly problem dependent, in our case, we conjecture that the choice of $N_p = aPK$ and $N_g = bPK$ works well, where the values of a and b strongly depend on the type of the GA algorithm used. Thus, for a given P, the computational complexity of the GA is $O(K^2)$. This corresponds to significant computational savings over the VA for $K \geq 10$. The implicit dependency on P is not a big problem. The sliding window approach allows a data length of $P' \ll P$ (typically 4–6, see [16]) to be used for the maximum-likelihood optimization with little loss compared to that of the full-size maximum-likelihood optimization.

The computational complexity/bit of the MSD is $O(K)$ [9]. Therefore, the computational complexity/bit of the C-GA/MSD is $O(K^2)$ and the computational complexity/bit of the E-GA/MSD is $O(K^2) + N_g N_p O(K)$. The latter corresponds to the case where the MSD is applied to all the members of the population. However, it is enough to apply the MSD to a small fraction of the population, as simulations show. In that case, the computational complexity/bit of the E-GA/MSD is $O(K^2) + \alpha N_g N_p O(K)$, where $0 \leq \alpha \leq 1$. For the values of α close to zero, the computational complexity is $O(K^2)$ and for the values of α close to 1, it is $O(K^3)$. So, α plays an important role in controlling the computational complexity. Simulations indicate that $\alpha = 0.1$ is a good choice. In conclusion, the GA-based approach offers methods with acceptable levels of complexity for the multiuser detection with a large user population. It should also be kept in mind that the GA-based algorithms can take advantage of parallel hardware, thanks to the intrinsic parallelism of the GA.

V. Simulations

A ten-user CDMA system is considered. The signature waveforms are derived from Gold sequences of length 15. The largest correlation among selected sequences is 7/15. A justification for this choice of sequences can be found in [11].

The first set of simulations are carried out for the synchronous case (i.e., $\tau_i = 0, \forall i$) using the following detectors:

- optimal detector (OD);
- conventional detector (CD);
- decorrelating detector (DEC);
- multistage detector (MSD);
- genetic algorithm (GA) detector;
- cascade GA and MSD detector (C-GA/MSD);
- embedded GA and MSD detector (E-GA/MSD).

The BER's of the first user with $\mathrm{SNR}(1) = E_1/N_o = 6$ dB are exhibited in Fig. 5. In these figures, the interfering signals strength relative to the desired user strength is increasing from left to right. The MSD is operated for three stages with its initial $\hat{\mathbf{b}}(1)$ taken from the CD. The default GA settings are summarized in Table I.

Fig. 5 clearly shows that the E-GA/MSD achieved near-optimal performance in all near–far ratios. Although the MSD improves the performance over the DEC, it is unable in some cases (here for $E_i/E_1 \leq 0$) to reach the optimal solution due to the unidimensional optimization As was observed elsewhere, it showed good performance when the interfering signals are stronger. With the settings given in Table I, both the GA and C-GA/MSD do not have near-optimal performance. It should be noted that the worse performance of the pure GA is due to the limited complexity used here ($N_p = 10$; $N_g = 10$). Its potential for the optimal performance is illustrated in Fig. 6 with the BER plotted against N_g for several values of N_p. The convergence and performance improve as we increase N_p. This clearly shows that when N_p is too small, the search over the simulation space is poor and slow, and when N_p is too large, the search good and fast but with too much time required to evaluate each generation. Here, the choice of $a = 3$ and $b = 1$ (see Section IV) provides a good compromise between the performance and complexity.

To illustrate the performance gain offered by the hybrid approach, we provide simulation results in terms of BER versus number of generations in Fig. 7(a) and (b) for $N_p = 10$ and $N_p = 30$, respectively, when all user powers are equal. The performances of the OD and MSD are also included to show the target gap in performance. With $N_p = 10$, none of

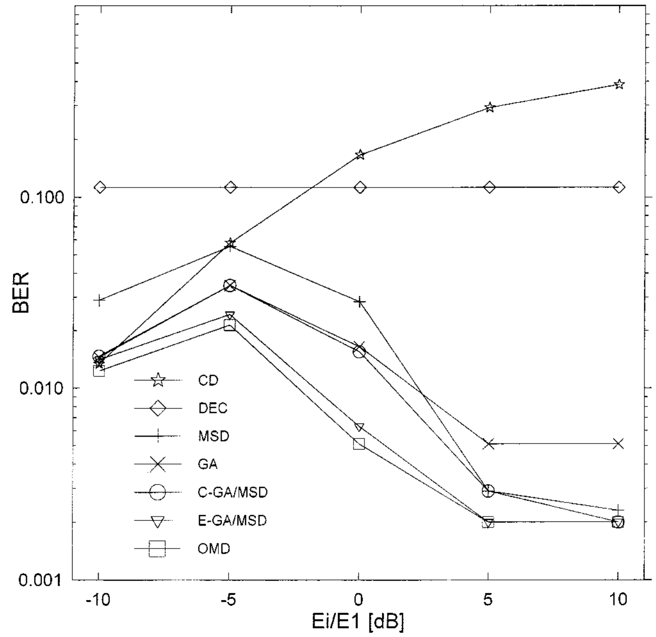


Fig. 5. BER versus E_i/E_1 of genetic-based multiuser detectors ($E_1/N_o = 6$ dB, $K = 10$, $L = 15$).

TABLE I
GA Parameters

Parameter	Value
Population Size, N_p	10
Crossover	1-point
Crossover probability, p_c	0.95
Mutation probability p_m	0.05
Replacement	Elitist
Generations, N_g	10
Selection	Roulette Wheel

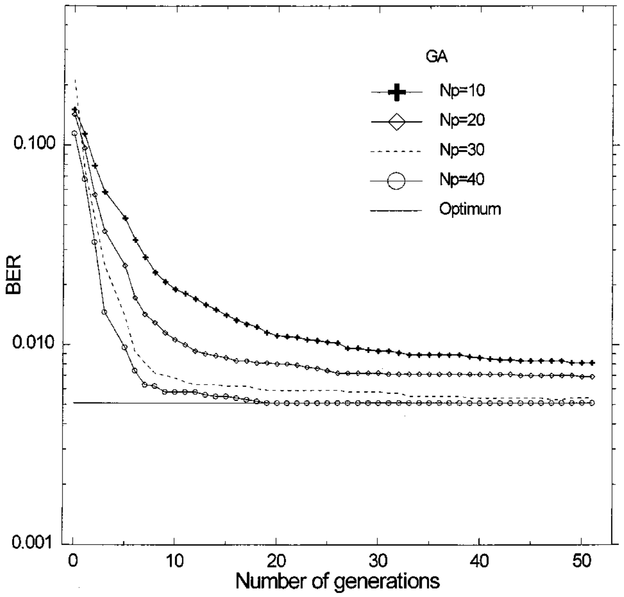
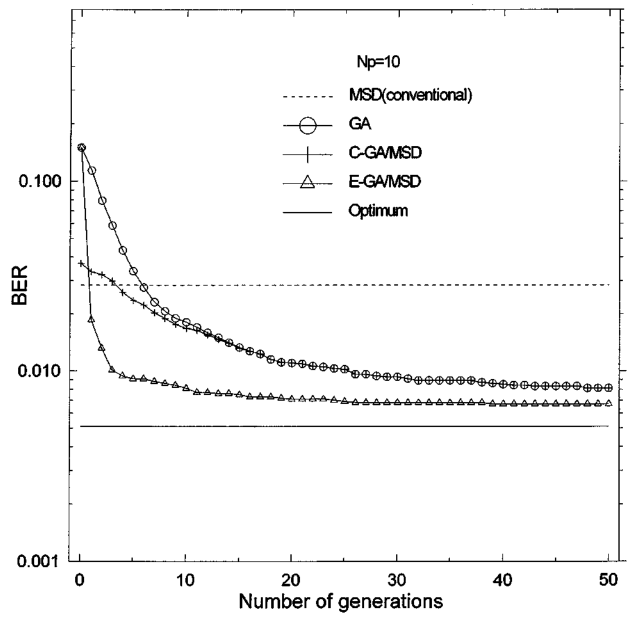


Fig. 6. BER performance of the GA detector with respect to number of generations for variation values of population $N_p(E_i/E_1 = 0$ dB, $E_i/N_o = 6$ dB, $K = 10$, $L = 15)$.

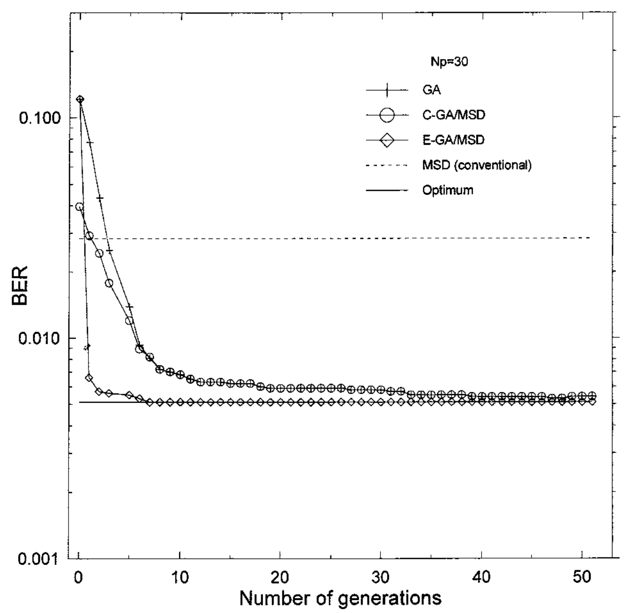
the GA-based detectors reached the optimum performance. However, with $N_p = 30$, all detectors were found capable of attaining the optimum performance. The fast convergence of the E-GA/MSD is obvious. After a few number of generations (typically 5–7), it has reached the performance that the others reached at the 50th generation. So, the E-GA/MSD, that is designed as a GA with a nonrandom "local" search method (MSD here), is very fast to locate the optimum solution. It should be noted that the C-GA/MSD can only perform better than the GA at early stages of evolution. That is, after a number of generations, the GA provides an initial decision that cannot be improved further by the MSD. All these results confirm our expectation that the local search around all candidates at each generation is very powerful compared to the local search around the fittest member after a number of generations.

In the simulations presented above, the MSD in the embedded detector is applied to the all members of the population at each iteration. However, it is possible to reduce the computational complexity by restricting the application of the MSD to $M < N_p$ members. Fig. 8 shows BER performance with respect to M at $E_1/N_o = 6$ dB and for E_i/E_1 equal to 0 and 5 dB. The results (also with those not shown) illustrate that the value of M can be fixed as a small percentage of the total number of members in the population.

The last set of simulations were carried out for the asynchronous case. Fig. 9 shows BER's of the CD, DEC, MSD, and GA-based detectors with respect to E_i/E_1 at $E_1/N_o = 6$ dB. The packet-size P was set to 4. Even with this packet size, the dimension of the optimization problem is very high, i.e., $PK = 40$. Because of the prohibitively long computation time, we were unable to present the OD performance. The results are an average of ten random trials of delays τ_k. The parameters were used as in Table I, except $N_p = 50$ and $N_g = 50$. The value of M was taken as 10% of the population, i.e., $\alpha = 0.1$ and $M = 5$. Fig. 9 again shows the superior performance of



(a)



(b)

Fig. 7. (a) BER comparison of the GA-based detector versus number of generations for $N_p = 10(E_i/E_1 = 0$ dB, $E_i/N_o = 6$ dB, $K = 10$, $L = 15)$. (b) BER comparison of GA-based detector versus number of generations for $N_p = 30(E_i/E_1 = 0$ dB, $E_i/N_o = 6$ dB, $K = 10$, $L = 15)$.

E-GA/MSD. It should be noted that the MSD as used here has failed to show a satisfactory performance. The pure GA has shown poor performance due to the fact that the population size and the maximum number of generations set here are relatively small for such a large optimization problem. This is clearly illustrated by the evolution curves exhibited in Fig. 10(a) and (b) for the worst and best cases of the ten trials. We note that it is very early to stop the pure GA after 50 iterations and apparently it will continue to reduce BER if allowed to run for more generations. As expected, the C-GA/MSD significantly improves the premature solution provided by the GA (in cases where its performance falls short of the performance of the MSD) at

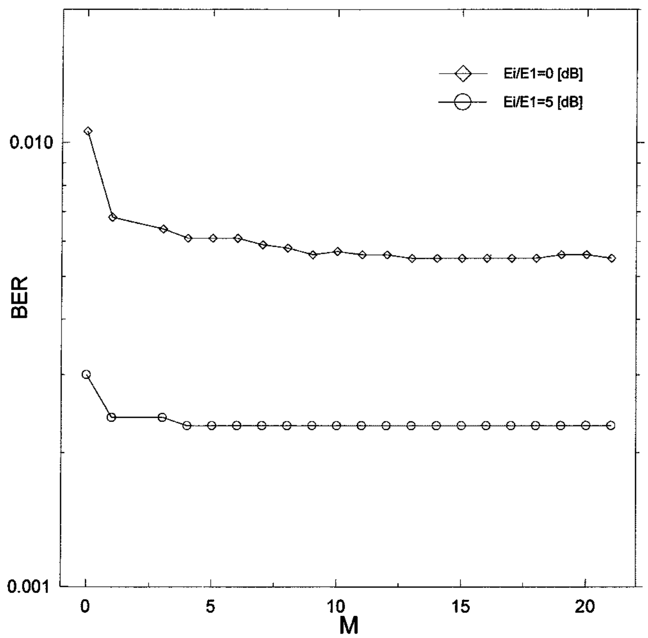


Fig. 8. BER performance of the E-/GA/MSD detector with respect to number of members to which the MSD is applied ($E_1/N_o = 6$ dB, $K = 10$, $L = 15$).

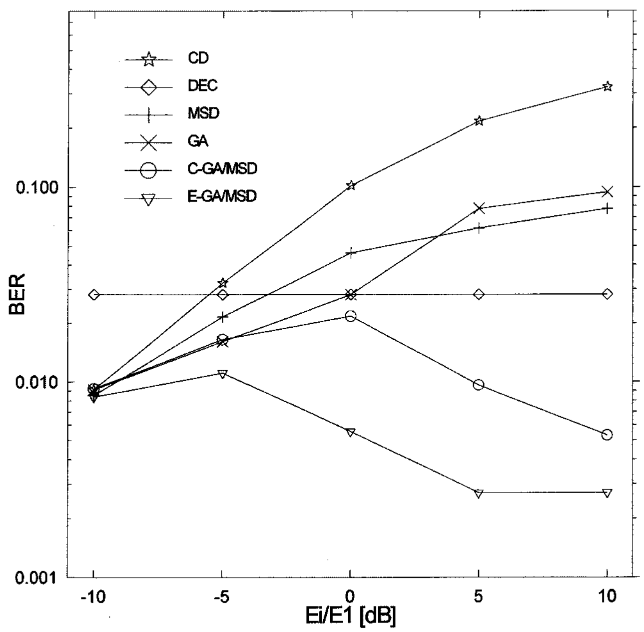
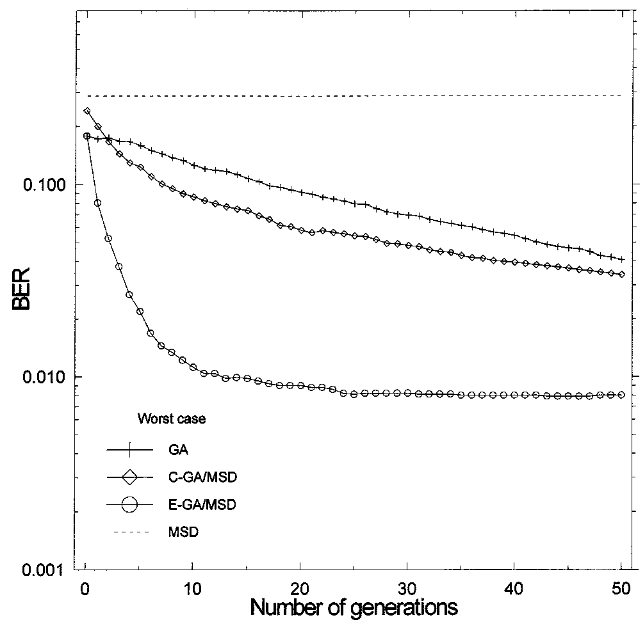
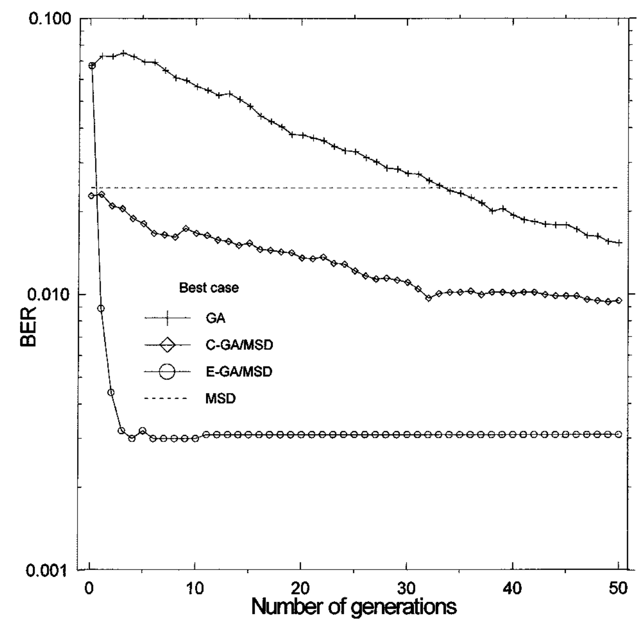


Fig. 9. BER versus E_i/E_1 of genetic-based multiuser detectors for asynchronous case ($E_i/N_o = 6$ dB, $K = 10$, $L = 15$, $P = 4$).



(a)



(b)

Fig. 10. (a) BER comparison of the GA-based detector versus number of generations for $N_p = 50$ in worse asynchronous case ($E_i/E_1 = 0$ dB, $E_1/N_o = 6$ dB, $K = 10$, $P = 4$, $L = 15$). (b) BER comparison of the GA-based detector versus number of generations for $N_p = 50$ in best asynchronous case ($E_i/E_1 = 0$ dB, $E_1/N_o = 6$ dB, $K = 10$, $P = 4$, $L = 15$).

the early stages of the evolution and this improvement reduces as the number of generations increases. The behavior of the E-GA/MSD is very impressive. It has converged quite fast to a quite good solution; hopefully, the global optimum. Occasional small increases in performance are due to the fact that the evolution curves are plotted for the desired user's best BER instead of the best fitness value, otherwise the curves will be monotonically nondecreasing due to the elitist replacement.

The results are in parallel with those obtained in the synchronous case. This is expected since the two cases differ only in the size of the search space. Nevertheless, the effectiveness and efficiency of the proposed E-GA/MSD becomes more apparent as the dimension of the search space increases.

Simulations have shown that the values of a and b required in the E-GA/MSD are noticeably smaller than those required in the pure GA detector. We found that $a = 1.25$ and $b = 0.25$ works very well in the former. The significance of this observation becomes more apparent when we detail the computational complexities of the pure GA detector and E-GA/MSD as $O(abg(K))$. The factor ab is considerably above one in the pure GA detector and well below one in the E-GA/MSD.

VI. Conclusions

The multiuser detection has been viewed as an optimization problem and the pure GA is used to solve the problem. Although pure GA has been shown to have the ability to find the optimal solution, a hybrid approach has been developed to reduce its computational requirements. The MSD, which can be viewed as a unidimensional optimizer, is hybridized with the GA in two different structures, namely, the cascaded and embedded GA/MSD. Simulation results have shown that the E-GA/MSD is an efficient and effective method for multiuser detection in both synchronous and asynchronous cases. It can achieve near-optimal performance and outperform the decorrelating and MSD's.

There is no doubt that advances in parallel hardware would make GA's (or its hybridized versions) more attractive and practical for various signal processing applications in communications [17], [18]. Otherwise, a GA-based solution may serve as a benchmark in a case where the exact solution is computationally untractable.

References

[1] A. J. Viterbi, "The orthogonal-random waveform dichotomy for digital mobile personal communications," *IEEE Pers. Commun.*, pp. 18–24, 1st qtr. 1994.

[2] W. C. Y. Lee, "Overview of cellular CDMA," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 291–302, May 1991.

[3] K. S. Gilhousen *et al.*, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303–312, May 1991.

[4] R. L. Pickholtz, L. B. Milstein, and D. L. Schilling, "Spread spectrum for mobile communications," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 313–322, May 1991.

[5] R. Kohno, R. Median, and L. B. Milstein, "Spread spectrum access methods for wireless communications," *IEEE Commun. Mag.*, pp. 58–67, Jan. 1995.

[6] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.

[7] R. Lupas and S. Verdu, "Linear multi-user detectors for synchronous code-division multiple access channels," *IEEE Trans. Inform. Theory*, vol. 35, pp. 123–136, Jan. 1989.

[8] ——, "Near–far resistance of multi-user detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. 38, pp. 496–508, Apr. 1989.

[9] M. K. Varansi and B. Aazhang, "Multi-stage detection in asynchronous code-division multiple access communications," *IEEE Trans. Commun.*, vol. 38, pp. 509–519, Apr. 1990.

[10] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[11] M. K. Varansi and B. Aazhang, "Near optimum detection in synchronous code-division multiple access systems," *IEEE Trans. Commun.*, vol. 39, pp. 725–735, May 1991.

[12] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Mag.*, pp. 22–37, Nov. 1996.

[13] P. Husbands, "Advances in parallel algorithms," in *Genetic Algorithms in Optimization and Adaptation*, L. Kronsjö and D. Shumsheruddin, Eds. Oxford, U.K.: Blackwell, 1992, pp. 227–269.

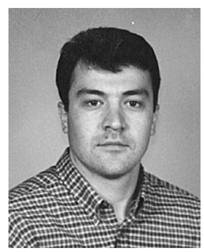
[14] C. Ergün and K. Hacioglu, "Application of a genetic algorithm to multistage detection in CDMA systems," in *Proc. 9th Mediterranean Electrotechnical Conf. (Melecon'98)*, vol. 2, Tel-Aviv, Israel, May 18–20, 1998, pp. 846–850.

[15] L. A. Wolsey, *Integer Programming*. New York: Wiley, 1998.

[16] T. Miyajima and T. Hasegawa, "Multiuser detection using a Hopfield network for asynchronous code-division multiple-access systems," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E79-A, no. 12, pp. 1963–1971, Dec. 1996.

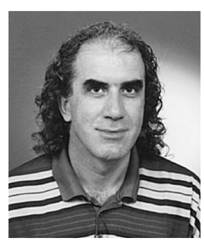
[17] S. Chen and Y. Wu, "Maximum likelihood joint channel and data estimation using genetic algorithms," *IEEE Trans. Signal Processing*, vol. 46, pp. 1469–1472, May 1998.

[18] F. Sweeney and C. F. N. Cowan, "The use of evolutionary techniques in channel equalization," in *Proc. 1st Int. Symp. on Communications Systems and Digital Signal Processing*, vol. 1, Sheffield, U.K., Apr. 6–8, 1998, pp. 36–39.



Cem Ergün was born in Luleburgaz, Turkey, on August 18, 1973. He received the B.S. and M.S. degrees in electrical and electronics engineering from the Eastern Mediterranean University, Famagusta, North Cyprus, in 1996 and 1998, respectively. Currently, he is a Research Assistant, working toward the Ph.D. degree, with the Department of Computer Engineering, Eastern Mediterranean University.

His main research interests include CDMA communications, genetic algorithms, and neural networks.



Kadri Hacioglu was born in Cyprus on September 1, 1958. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronic engineering from the Middle East Technical University, Ankara, Turkey, in 1980, 1984, and 1990, respectively.

After his two-year military service, in 1992, he joined the faculty of Eastern Mediterranean University, Magosa, North Cyprus, where he is now an Associate Professor in the Department of Electrical and Electronic Engineering. Since August 1998, he has been a Visiting Professor in the Computer Science Department, and since January 2000, he has been a Research Associate at the Center for Spoken Language Research, University of Colorado at Boulder, on leave from Eastern Mediterranean University, Gazi Magosa, North Cyprus. His current research interests are applications of neural networks, genetic algorithms and fuzzy logic to communications/signal processing problems, neural network-based recognition, and language modeling.