A Multi-objective Algorithm for DS-CDMA Code Design Based on the Clonal Selection Principle

Daniel Stevens
Dept. of Electrical Engineering
Kansas State University
Manhattan, KS 66506 USA
1 + (785) 532-5600
dans@ksu.edu

Sanjoy Das
Dept. of Electrical Engineering
Kansas State University
Manhattan, KS 66506 USA
1 + (785) 532-5600
sdas@ksu.edu

Bala Natarajan
Dept. of Electrical Engineering
Kansas State University
Manhattan, KS 66506 USA
1 + (785) 532-5600
bala@ksu.edu

ABSTRACT
This paper proposes a new algorithm based on the clonal selection principle for the design of spreading codes for DS-CDMA. The algorithm follows a multi-objective approach, generating complex spreading codes with good autocorrelation as well as good cross-correlation properties. It also enables spreading code design with no restrictions on the number of users or code length. The algorithm maintains a repertoire of codes that are subject to cloning and undergo a process of affinity maturation to obtain better codes. Results indicate that the produces code sets that lie very close to the theoretical Pareto front.

Categories and Subject Descriptors
D.3.3 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *heuristic methods.*

General Terms: Algorithms, Design.

Keywords: immunocomputing, clonal selection, optimization, CDMA.

1. INTRODUCTION
1.1 DS-CDMA Code Design
We are currently experiencing a significant proliferation of wireless communication systems and devices in our daily life. These systems and devices share the wireless channel by employing smart *multiple access* techniques. Direct-sequence code division multiple access (DS-CDMA) [12] is one of the most popular multiple access techniques and is the technology driving the third generation cellular as well as the wireless local area network (WLAN) market.

Considering a DS-CDMA based cellular network as an example, multiple users are allowed to transmit their information (digitized voice or data) signals simultaneously over the same frequency. However, each user is assigned a specific spreading code which serves as the ID for that user. An analogy to DS-CDMA could be that of being at a cocktail party with two humans and two aliens. The humans and aliens talk at the same time and over the same frequency, but the language used by humans is different from that of the aliens. This makes the human communication undetectable to the aliens and vice-versa. Here the language becomes the code.

The choice of the spreading code plays a significant role in the quality of service experienced by the user. One of the most significant limitations in multiple access communication over a finite bandwidth channel is the interference introduced by competing users of the channel. This can be thought of as the effect of increasing the number of humans in the cocktail party. As more people begin talking, the less distinguishable a certain person will become. This multiple access interference (MAI) reduces the practical channel capacity and leads to reduced performance as a result of increased noise levels. In an asynchronous DS-CDMA system, this MAI can be combated using efficient spreading codes with "good" cross-correlation (CC) properties [8]. It is important to note that since many users may be operating in the DS-CDMA system at any time, the CC properties of all sequences in the set should be considered when determining the average performance. The auto-correlation (AC) properties of a set of codes are important for spectrum spreading as well. As with CC, the search for codes with "good" AC properties should consider the average AC properties of a code set.

The design of spreading codes has been a widely studied, and several methods for generating code sets with good AC and CC properties have been proposed [10]. Unfortunately, the prior works provide binary or complex spreading codes that have restrictions on the code length N (i.e., N is limited to 2^n or $2^n \pm 1$, where $n \in I$, or N is prime). Furthermore, there is a restriction on the number of users, K, in the prior work. Walsh-Hadamard codes consider code sets with $K = N$, while in Gold codes, $K = N + 2$.

In this paper, an artificial immune system (AIS) approach for the multi-objective optimization of direct-sequence code division multiple access (DS-CDMA) system performance is proposed. The proposed method allows more flexible code design, placing no restriction on the number of codes N, or users K.

1.2 The Clonal Selection Principle
Clonal selection is a central feature of the vertebrate immune system. The immune system's primary function is to respond and eliminate foreign molecules known as antigens. In order to do so, this system makes use of white blood cells, including the B-cells to recognize antigens. Following the pattern recognition step, the immune system then initiates an adaptive response mechanism to cull out the invading antigens by means of antibodies. The ability of the antibodies to recognize and respond to an antigen is known as its affinity. Cells that are more capable of recognizing and subsequently

GECCO'05, June 25-29, 2005, Washington, DC, USA.

removing a given antigen have a higher affinity than those that do not. These antibodies proliferate the immune system by a mechanism called cloning [4,5]. Cloning is a mitotic process which produces exact copies of the parent cells. The clones are then subject to a process called affinity maturation. Affinity maturation ensures that the affinities of the cells are improved to allow the speedy removal of the foreign antigens. Within affinity maturation, the cells undergo a high rate of mutation called hypermutation. In a bid to increase their affinities, those cells having a lower affinity have a higher probability of mutation. Following the affinity maturation process, cells undergo selection. Those with affinities lower than a certain threshold are removed from the immune system.

Algorithms for optimization modeled after clonal selection theory have been proposed recently. These algorithms maintain a large population/repertoire of solutions and are not unlike genetic algorithms. The individual solutions within the repertoire correspond to antibodies. Better solutions are assigned higher affinities. The individuals undergo affinity maturation, and those with higher affinities are merged with the repertoire. An algorithm by de Castro and von Zuben, CLONALG (CLONal selection ALGorithm) [5] offers a versatile algorithm for learning as well as stochastic optimization. CLONALG provides the necessary theoretical backdrop for the proposed algorithm, although the present research addresses a multi-objective problem.

1.3 Multi-objective Optimization

When dealing with optimization problems with multiple objectives, the conventional concept of optimality does not hold good [2,7]. Hence, the concepts of dominance and Pareto-optimality are applied. Without a loss of generality, if we assume that the optimization problem involves minimizing each objective $e_i(.),\ i = 1...M$, a solution u is said to dominate over another solution v iff $\forall i \in \{1,2,\ldots,M\},\ e_i(u) \le e_i(v)$ with at least one of the inequalities being strict, i.e. for each objective, u is better than or equal to v and better in at least one objective. This relationship is represented as $u \prec v$. In a set of solution vectors, the set of all non-dominating solutions is called the Pareto front. In other words, if S is the population, the Pareto Front is the set,

$$\Gamma = \{u \in S \mid \forall v \in S, \neg(v \prec u)\}. \quad (1)$$

Given any set S of solutions, we will use the notation $u \prec S$ to mean $v \in S \Rightarrow u \prec v$.

The simplistic approach of aggregating multiple objectives into a single one often fails to produce good results as it produces only a single optimal solution. Multi-objective optimization on the other hand involves extracting the entire Pareto front from the solution space. In recent years, many biologically inspired algorithms for multi-objective optimization have been proposed. Recent multi-objective algorithms make use of an archive to record the non-dominant solutions that they compute at intermediate stages [6,13,14]. Multi-objective optimization approaches that are based on artificial immune systems have recently been proposed [3].

2. PROBLEM DESCRIPTION

A user x's code u_x is a vector of the form $[e^{j\phi_{xi}}]$, i = 1, 2, ..., N, where each $\phi_{xi} \in [0,2\pi]$ is a phase angle and N the code length.

The aperiodic correlation function between any two codes u_x and u_y of users x and y, for any displacement l is given by,

$$C_{xy}(l) = \begin{cases} \sum_{k=0}^{N-1-l} u_x(k) u_y^*(k+l), & 0 \le l \le N-1 \\ \sum_{k=0}^{N-1+l} u_x(k-l) u_y^*(k), & 1-N \le l \le 0 \\ 0, & \text{elsewhere} \end{cases} \quad (2)$$

When there are K such users, the AC function is given by,

$$R_{ac} = \frac{1}{KN^2} \sum_{x=1}^{K} \sum_{\substack{l=1-N \\ l \neq 0}}^{N-1} \left| C_{xx}(l) \right|^2 \quad (3)$$

where $C_{xx}(l)$ is obtained from Equation (2) by letting $y = x$. In a similar manner, the CC function is,

$$R_{cc} = \frac{1}{K(K-1)N^2} \sum_{x=1}^{K} \sum_{\substack{y=1 \\ y \neq x}}^{K} \sum_{l=1-N}^{N-1} \left| C_{xy}(l) \right|^2 \quad (4)$$

In Equations (3) and (4), the R_{cc} and R_{ac} have been normalized with respect to the code length in order to facilitate comparisons between codes of different lengths.

This paper describes a method based on the clonal selection principle to determine a set of K codes of length N with good R_{cc} and R_{ac} properties. Ideally, both correlations should be minimized in an optimal set of codes. Unfortunately, codes with very good ACs will have a strong CC between them, while codes with minimal CC properties will have a strong AC. In fact it has been shown that for any set of K codes of length N,

$$R_{cc}(K-1) + R_{ac} > K-1 \quad (5)$$

Replacing the ">" sign with an equality ("=") in Equation (5) defines a straight line in the objective function space. The more optimal a code set is in terms of its AC and CC properties, the closer it will remain to the straight line. In other words, this line defines the theoretical Pareto front in the space of all possible code sets.

In an earlier work, a multi-objective genetic algorithm was proposed for [1,9] optimizing code-sets. The algorithm was able to extract excellent codes for any values of N and K, but with markedly slow convergence. Mutation considered in that paper that work was purely random and did not consider the individual contribution of codes within code-sets towards the AC and CC, resulting in slower convergence. In this paper a multi-objective approach based on the clonal selection principle to extract code sets is proposed. Our algorithm considers problem-specific hypermutations which preferentially attempt to remove codes with worse correlation properties. Furthermore, the genetic algorithm was encumbered with the task of having to maintain a large population which contained dominated individuals. In this study, only a repertoire of non-dominated code-sets is allowed in each iteration, while the dominated ones are removed before the next iteration. As will be seen later, these steps result in considerably faster convergence. With the improved speedup, the algorithm comes one step closer to practical implementation in a cellular system. The overall goal is to

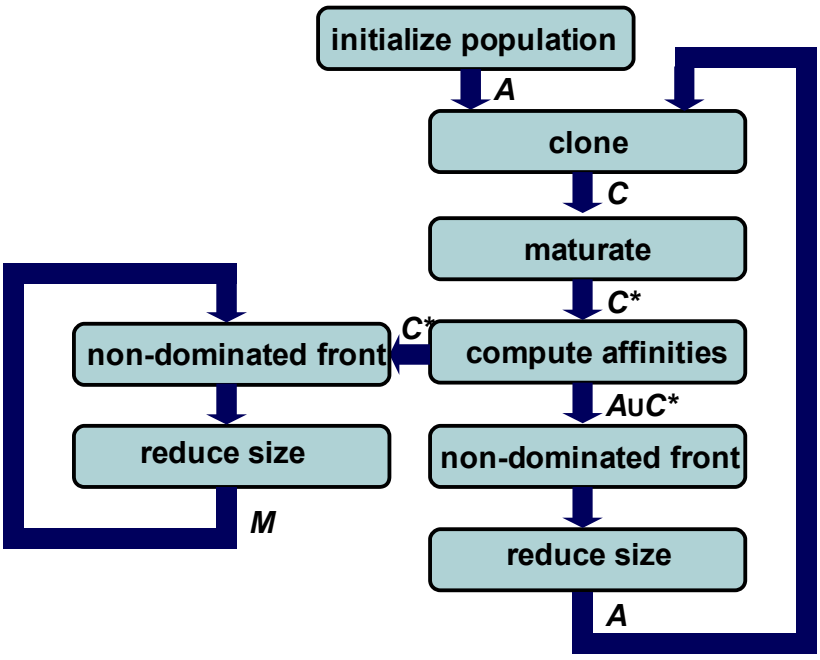

Figure 0. A Schematic of the Proposed Algorithm

employ the code generation algorithm at the cellular base station (tower) and periodically optimize the code sets in use in a particular cell. Therefore, speed of optimization is critical in determining how well we can adapt to changes in the system load in terms of the number of users.

3. PROPOSED APPROACH

3.1 Algorithm Description

The proposed algorithm maintains a repertoire of individual code-sets, each code set corresponding to an antibody. The phase angles of the codes in each individual of that repertoire are arranged as a $K \times N$ matrix of the form $[\phi_{xi}]$, $i = 1, 2, ..., N$, $x = 1, 2, ..., K$. The algorithm begins generating codes whose phase angles are randomly assigned values between 0 and 2π. The AC and CC of each individual code-set is computed and only the non-dominated individuals are selected to form the initial repertoire $\boldsymbol{A}$, while the rest are discarded. This non-dominated front also forms the initial archive $\boldsymbol{M}$.

Within each iteration of the algorithm, the individual code sets in the repertoire are cloned. The set of clones $\boldsymbol{C}$ per iteration, whose size is fixed throughout the algorithm, then undergoes affinity maturation. During the affinity maturation process, the codes are hypermutated by selecting, in a probabilistic manner, a row of each clone which contributes poorly to the CC or AC, and then mutates that row. The details of this technique are postponed until subsection 3.2.

Following affinity maturation, a set of hypermutated clones, $\boldsymbol{C}^*$ is obtained. The affinities of the hypermutated clones in $\boldsymbol{C}^*$ are computed, following which they are combined with the main antibody repertoire $\boldsymbol{A}$ to form $\boldsymbol{A}^*$. The dominated individuals in $\boldsymbol{A}^*$ are removed. In order to prevent $\boldsymbol{A}$ from getting too large, only a fixed number of individuals are picked from the remaining solutions for the next generation of the algorithm. Initial experiments with clustering on immune system based algorithms for code set design by the authors, to pick uniformly spaced individuals, have shown that clustering have shown no perceptible effect on the convergence of the algorithm, and hence the computationally less expensive method of randomly selecting individuals from the non-dominated front in $\boldsymbol{C}^* \cup \boldsymbol{A}$ was adopted here.

The archive $\boldsymbol{M}$ is also updated by introducing the new individuals $\boldsymbol{A}^*$ and removing the individuals from $\boldsymbol{A}^* \cup \boldsymbol{M}$ that are dominated or repeated. Unlike in the elitist strategy pursued by current multi-objective evolutionary algorithms [6,14], where archived individuals can reenter the main population, initial studies for the problem of minimizing AC and CC revealed that such a strategy was not effective in DS-CDMA code design. Although the CC and AC measures are competing objectives, the CC which has a higher degree of epistasis involving all the codes in the individual, is more difficult to optimize. Consequently, the average AC in the repertoire is optimized at a much faster rate than the CC. Archived elite individuals from previous generations would have better AC properties than CC, and hence reintroducing them into A is counter-productive.

The overall algorithm is outlined in Figure 1.

3.2 Affinity Maturation

The exploratory nature of stochastic algorithms, such as immune system based algorithms, as well as traditional evolutionary algorithms, enables them to sample a variety of regions in the search space, locating regions that contain better individuals. Unfortunately a purely exploratory algorithm ignores local features in the solution space, resulting in slow convergence. Empowering the algorithms with the ability to exploit local features results in faster convergence. The present algorithm combines exploratory and exploitative behavior into a single step, the hypermutation operation, which is carried out to maturate the freshly generated clones. Hypermutation attempts to strike a balance between exploration and exploitation by

considering different ways to produce new code-sets from old ones. While this operation often carries out purely random perturbations, it also allows the removal of inferior codes from existing code-sets.

Hypermutation is carried out by selecting a code, a single row in each solution matrix, and replacing it with a randomly generated code whose N phase angles each lie in the interval $[0, 2\pi]$. Only a single code is selected for replacement per individual. In order to identify better codes from worse ones, their individual contributions to the affinity measure of Equations (3) and (4) need to be computed. The contribution of any code x to the AC of Equation (2) is given by,

$$R_{ac,x} = \frac{1}{N^2} \sum_{\substack{l=1-N \\ l\neq 0}}^{N-1} \left| C_{xx}(l) \right|^2 , \qquad (6)$$

such that,

$$R_{cc} = \frac{1}{K(K-1)} \sum_{x=1}^{K} R_{cc,x} \qquad (7)$$

Likewise, any code x's contribution to the CC of Equation (4) can be written as,

$$R_{cc,x} = \frac{1}{N^2} \sum_{\substack{y=1 \\ y\neq x}}^{K} \sum_{l=1-N}^{N-1} \left| C_{xy}(l) \right|^2 . \qquad (8)$$

Therefore, Equation (4) can be rewritten as,

$$R_{cc} = \frac{1}{K(K-1)} \sum_{x=1}^{K} R_{cc,x} \qquad (9)$$

The code, x to be replaced by hypermutation is identified in a probabilistic manner, using one of three possible strategies (*i*) *random*, i.e. x is chosen randomly from $\{1,2,...K\}$ with equal probability; (*ii*) *worst AC*, i.e. let $x = \arg\max_y \{R_{ac,y}\}$ (*iii*) worst CC, i.e. $x = \arg\max_y \{R_{cc,y}\}$ The probabilities to do so are fixed at rates μ_{rand}, μ_{ac} and μ_{cc} respectively, such that $\mu_{rand} + \mu_{ac} + \mu_{cc} = 1$.

4. RESULTS AND DISCUSSION

4.1 Experimental Setup

The entire algorithm was implemented in MATLAB. The size of the antibody repertoire was fixed at 10. code sets of length $N = 8$, with $K = 8$, 12, and 16; $N = 16$, with $K = 16$, 24, and 32; $N = K = 32$; $N = K = 64$, were tested. All the simulations were run for 10,000 iterations, although simulations for N and $K \leq 16$ typical converged after only 3,000 to 5,000 iterations. Each antibody was cloned four times, and the hypermutation rates were fixed at $\mu_{rand} = 0.7$, $\mu_{ac} = 0.15$, and $\mu_{cc} = 0.15$. A discrete version of the algorithm was also run, where the phase angles, $\phi_{xi} \in [0, 2\pi]$ were allowed to acquire only discrete values in $\{2\pi/N, 4\pi/N, 6\pi/N, ..., 2\pi\}$.

4.2 Performance Analysis

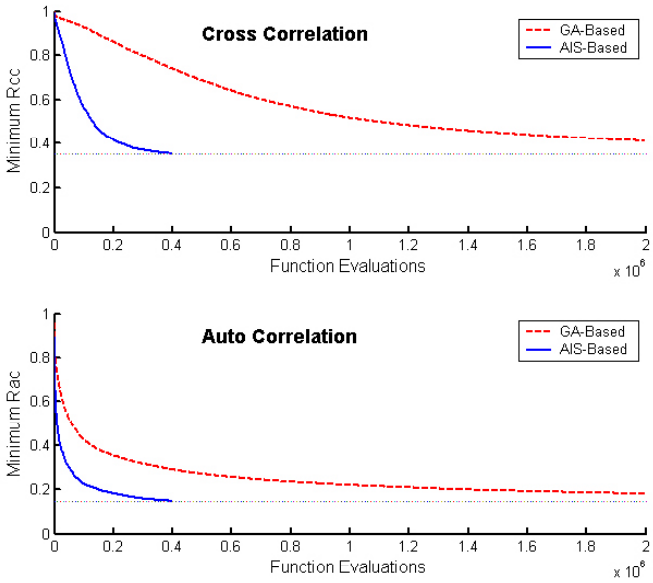

Figure 1. Comparison with Genetic Algorithm

In comparison to the genetic algorithm, the proposed method converged significantly faster. The comparison of the two is shown in Figure 2 for $N = 16$ and $K = 16$.

The simulations indicate that the solutions obtained using the proposed AIS approach was very close to the theoretical Pareto front of Equation (5). Figure 3 shows the fronts obtained with $N = 8$, 16, 32, 64. The number of users, K was set equal to N in each case.

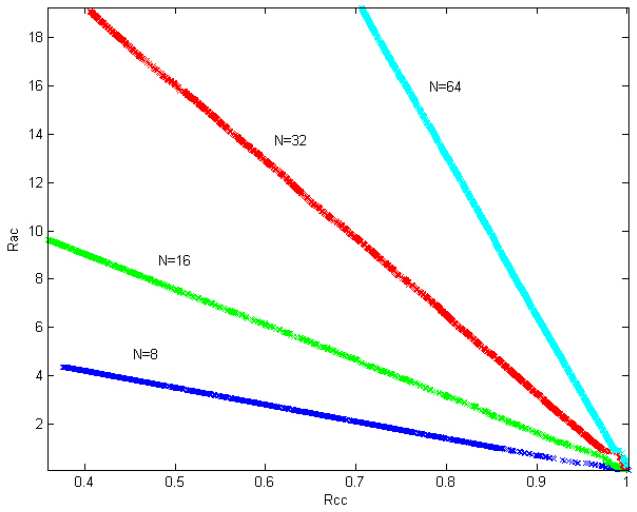

Figure 2. The Pareto front obtained by the proposed AIS approach for different values of N.

To show that the algorithm could efficiently extract the Pareto front even when K was different from N, it was run for several values of K, while N was held constant at 16. The result is shown in Figure 4, for K = 16, 24 and 32. The results clearly show that the proposed method can obtain the Pareto-optimal code sets even when $K > N$.

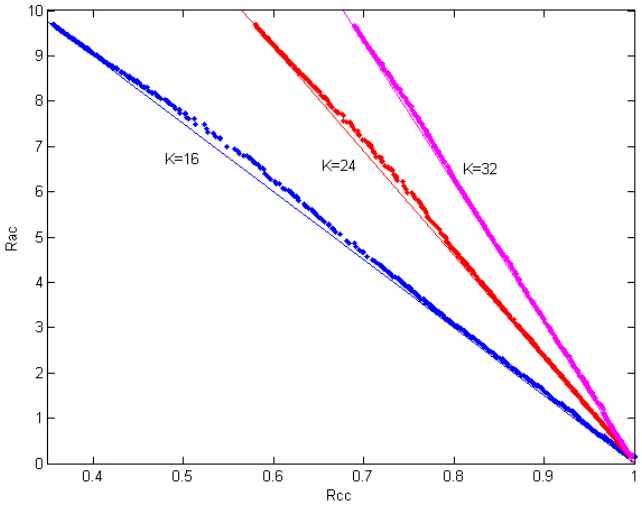

Figure 3. The Pareto front obtained by the proposed AIS algorithm for different values of K.

Finally, Table 1, compares the CC and AC properties of codes produced by the AIS algorithm with other well known code sets [11]. It is evident that the AIS-based approach not only provides codes that have comparable correlation properties but also yields a wide range of codes with intermediate R_{cc} and R_{ac} values. These intermediate codes tradeoff R_{cc} properties for R_{ac} and are vital for DS-CDMA system design. It can be observed that the intermediate Opperman codes are comparable to the intermediate codes of the multi-objective AIS, but the Opperman codes cannot be designed for any length N (while maintaining the K=N-1 capacity limit).

Table 1. A comparison of AIS based codes with those obtained by other methods

Code Set N=31 K=30	R_{cc}	R_{ac}
AIS Codes	0.354	19.071
	0.507	14.875
	0.700	9.1934
	1.000	0.1473
HW (N=32, K = 31)	1.000	0.219
CI Code	0.355	19.677
Gold Codes	0.970	0.900
EOE-Gold Codes	0.950	0.952
FZC Codes	1.000	0.344
Opperman Codes	0.400	18.200
	0.500	19.670
	1.000	0.620

The proposed approach is extended to consider only discrete phase angles. A discrete version of the algorithm not only significantly lowers the implementation complexity, but also makes a it less expensive to store the codes. Instead of storing the entire phase angles in double precision format, only the index of a code can be stored. Storing each phase angle would be accomplished in a byte, greatly diminishing the storage space requirements.

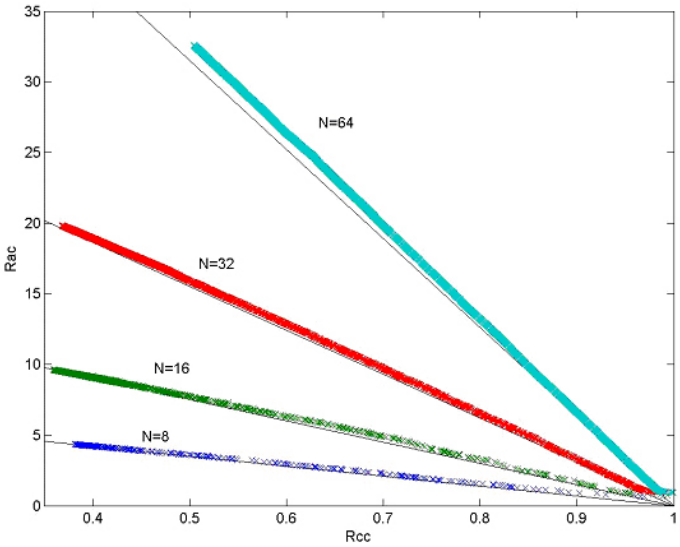

Figure 5. The Pareto front obtained by the proposed AIS approach for different values of N (discrete version).

Figure 5 shows the fronts obtained from the discrete algorithm with N = 8, 16, 32, 64. As before, the number of users, K was set equal to N in each case.

5. CONCLUSION

This research outlines an AIS based technique to determine optimal DS-CDMA spreading codes. Stochastic methods such as the proposed algorithm offer a far greater degree of flexibility than conventional methods in terms of the code length and number of codes. Additionally, the output code sets produced exhibit a wide range of correlation properties that is useful in actual code design.

6. REFERENCES

[1] Buehler, E., Natarajan, B., and Das, S. Multi-objective Genetic Algorithm Based Complex Spreading Code Sets with a Wide Range of Correlation Properties. Proceedings. *IEEE 15th International Conference on Wireless Communications, Calgary, Alberta, Canada*, 2 (2003) 548-552.

[2] Coello Coello, C.A. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1, 3 (Aug 1999) 269-308.

[3] Coello Coello, C.A., and Cortez, N.C. An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System. *First International Conference on Artificial Immune Systems (ICARIS'2002)*, Canterbury, U.K., (Sept. 2002), 212-221.

[4] Cutello, V., and Nicosia, G. The Clonal Selection Principle for In Silico and In Vitro Computing. *Recent Developments in Biologically Inspired Computing,* Eds. de Castro, L.N., Von Zuben, F.J., Idea Group, Hershey, PA, 2004.

[5] de Castro, L.N., and Von Zuben, F.J. Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation,* 6, 3 (June 2002) 239-251.

[6] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation.* 6, 2 (2002) 182-197.

[7] Fonseca, C.M., and Fleming, P.J. An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, 3, 1 (Spring 1995) 1-16.

[8] Karkkainen, K.H. Mean-square cross-correlation as a performance measure for spreading code families. *Proc. IEEE 2nd Intl. Symp. on Spread Spectrum Techniques and Applications*, 147-150

[9] Natarajan B., Das S., and Stevens, D. A Multi-objective Evolutionary Approach to DS-CDMA Complex Spreading Code Design. *IEEE Transactions on Wireless Communications*, accepted for publication.

[10] Opperman, I., and Vucetic, B.S. Complex spreading sequences with a wide range of correlation properties. *IEEE Transactions on Communications,* 45 (1997), 365-375

[11] Rappaport, T.S., *Wireless Communications – Principles and Practice,* Prentice Hall, 2002.

[12] Viterbi, A.J. *CDMA: Principles of spread spectrum communication*. Addison Wesley Publishing Company, 1995

[13] Zitzler, E., and Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999) 257-271.

[14] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., and da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, 7, 2 (Apr. 2003) 117-132.