A Hybrid Evolutionary Programming Algorithm for Spread Spectrum Radar Polyphase Codes Design

Angel M. Perez-Bellido
Department of Signal Theory and Communications
Universidad de Alcala
28871 Alcala de Henares, Madrid, Spain.
apb16758@alu.uah.es

Sancho Salcedo-Sanz
Department of Signal Theory and Communications
Universidad de Alcala
28871 Alcala de Henares, Madrid, Spain.
sancho.salcedo@uah.es

Emilio G. Ortiz-Garcia
Department of Signal Theory and Communications
Universidad de Alcala
28871 Alcala de Henares, Madrid, Spain.
eog14960@alu.uah.es

Antonio Portilla-Figueras
Department of Signal Theory and Communications
Universidad de Alcala
28871 Alcala de Henares, Madrid, Spain.
antonio.portilla@uah.es

ABSTRACT

This paper presents a hybrid evolutionary programming algorithm to solve the spread spectrum radar polyphase code design problem. The proposed algorithm uses an Evolutionary Programming (EP) approach as global search heuristic. This EP is hybridized with a gradient-based local search procedure which includes a *dynamic step adaptation procedure* to perform accurate and efficient local search for better solutions. Numerical examples demonstrate that the algorithm outperforms existing approaches for this problem.

Categories and Subject Descriptors

J.6 [**Computer Applications**]: Computer-aided Engineering.

General Terms

Algorithms, Design.

Keywords

Polyphase codes, code design, evolutionary programming, hybrid algorithms

1. INTRODUCTION

Range resolution of radar systems can be significantly improved by using short pulses. Unfortunately, utilizing short pulses decreases the average transmitted power, which can affect the radar's normal modes of operation. Since the average transmitted power is directly linked to the receiver SNR, it is often desirable to increase the pulse width (i.e., increase the average transmitted power) while simultaneously maintaining adequate range resolution. This can be made possible by using *pulse compression* techniques. Pulse compression allows us to achieve the average transmitted power of a relatively long pulse, while obtaining the range resolution corresponding to a short pulse. Thus, the majority of modern radars systems generally incorporate pulse compression waveforms.

In radar systems with pulse compression, the choice of the appropriate waveform is a key point. There are several pulse compression methods, like Barker codes, chirp-type modulation or polyphase codes [1]. Among these methods for radar pulse modulation, the polyphase codes offer some convenience in comparison to analog techniques, such as chirp-type modulations [2]: polyphase codes produce lower sidelobes in the compressed signal, and easier digital processing techniques implementation. Figure 1 shows a possible implementation of polyphase codes in a radar system.

In [2], Dukic and Dobrosavljevic introduced a new method for polyphase pulse compression code design, based on the properties of the aperiodic autocorrelation function, and considering coherent radar pulse processing in the receiver. It can be modelled as a min-max nonlinear optimization problem, in the following way:

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) = \max\{\varphi_1(\mathbf{x}), \cdots, \varphi_{2m}(\mathbf{x})\}, \quad (1)$$

$$X = \{(x_1, \cdots, x_n) \in \mathbb{R}^n | 0 \leq x_j < 2\pi, j = 1, \cdots, n\}, \quad (2)$$

where $m = 2n - 1$, and

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

$$\varphi_{2i-1}(\mathbf{x}) = \sum_{j=1}^{n} cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right),\ i = 1, \cdots, n, \quad (3)$$

$$\varphi_{2i}(\mathbf{x}) = 0.5 + \sum_{j=i+1}^{n} cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right),\ i = 1, \cdots, n-1, \quad (4)$$

$$\varphi_{m+i}(\mathbf{x}) = -\varphi_i(\mathbf{x}),\ i = 1, \ldots, m.$$

Note that the variables x_k represent symmetrized phase differences, and the objective of the problem is to minimize the module of the biggest among the samples of the autocorrelation function φ.

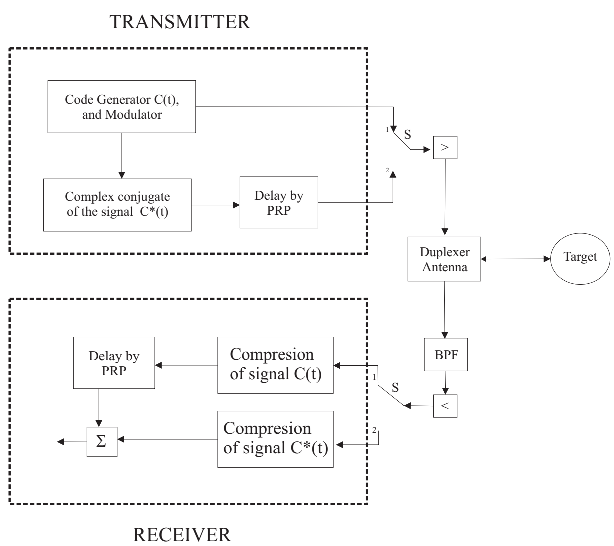


Figure 1: Configuration of a radar system using the proposed polyphase codes. Abbreviations PRP and BPF stand for Pulse Repetition Period and Band-Pass Filter, respectively. S stands for a switch which is turned from position 1 to 2 with PRP.

The problem associated with the design of polyphase codes in radar systems using this model is usually called Spread Spectrum Radar Polyphase code design problem (SSRP), and has been tackled using different approaches in the literature. In [6] a Tabu search approach has been proposed to solve the problem. In [3] several heuristic algorithms to the SSRP are proposed and compared, including a multi-level tabu search heuristic and a variable neighborhood algorithm among them. In [8] a genetic algorithm is able to improve the results obtained by the Tabu search in several SSRP instances.

In this paper we present a novel approach to the SSRP, based on a memetic algorithm [10]. Our memetic approach is based on a Fast Evolutionary Programming approach [7], hybridized with a *Gradient-guided* local search procedure to improve the quality of the individuals in the population. We implement a gradient-guided local algorithm with controlled step size. This technique is easy to implement and does not increase much the computational cost of the algorithm. Our approach is compared with the results obtained by previous approaches to the SSRP: a tabu search, a variable neighborhood search and a genetic algorithm are compared with our hybrid EP approach. We show that our proposal obtains the best solutions for the SSRP instances compared.

The rest of the paper is structured as follows: next section describes the hybrid evolutionary programming approach proposed in this paper, including gradient-based local search procedure. Section 3 presents several simulations in different instances to show the performance of our algorithm. Comparisons with the results obtained by a tabu search approach, a variable neighborhood search and an existing genetic algorithm are provided. Section 4 gives some final conclusions for this paper.

2. A HYBRID EVOLUTIONARY PROGRAMMING APPROACH

Evolutionary Programming (EP) [7], is a population-based meta-heuristic algorithm which has been successfully applied to many numerical and combinatorial optimization problems. Optimization using EP can be summarized to have two major steps: first, mutation of solutions in the current population, and second, selection of the next generation population from the mutated and the current solutions set. The algorithm starts with a randomly generated initial population of strings, where each string is formed by a pair of real encoded vectors. The first part of the string is a possible value of the problem solution vector of phase differences x, and the second part of the encoding is formed by the standard deviations for the mutations which will be applied following in the algorithm. These standard deviations are self-adaptive parameters in the EP algorithm [7]. The population is then evaluated, assigning a fitness score to each individual. This fitness score is based on the objective function of the SSRP, given by Equation (1). Each parent creates then a single offspring by means of applying a mutation procedure. Several mutations have been used, with different properties, depending on the problem tackled, among them, Gaussian mutations and Cauchy-type mutations have been applied with success in continuous optimization problems [7]. The Classical Evolutionary Programming (CEP) can be summarized with the following steps:

1. Generate an initial population of μ individuals (solutions). Set $k = 1$. Each individual is taken as a pair of real-valued vectors $(\mathbf{x}_i, \sigma_i)$, $\forall i \in \{1, \cdots, \mu\}$, where $\mathbf{x}_i$'s are objective variables (e.g., magnitudes of the phase differences for the SSRP), and σ_i's are standard deviations for Gaussian mutations.

2. Evaluate the fitness value for each individual ($\mathbf{x}_i$, σ_i) (using the defined objective function in Equation (1)).

3. Each parent $(\mathbf{x}_i, \sigma_i)$, $\forall i \in \{1, \cdots, \mu\}$ then creates a single offspring ($\mathbf{x}'_i$, σ'_i) as follows:

$$\mathbf{x'}_i = \mathbf{x}_i + \sigma_\mathbf{i} \cdot \mathbf{N(0,1)} \quad (5)$$

$$\sigma'_\mathbf{i} = \sigma_\mathbf{i} \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot \mathbf{N(0,1)}) \quad (6)$$

where $N(0,1)$ denotes a normally distributed one dimensional random number with mean zero and standard deviation one, and $\mathbf{N(0,1)}$ is a vector containing random numbers of mean zero and standard deviation one, generated anew for each value of i. The parameters τ and τ' are commonly set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$, respectively [7].

4. If $x_i(j) \geq 2\pi$ or $x_i(j) < 0$ then $x_i(j) = x_i \mod 2\pi$.

5. Calculate the fitness values associated with each offspring $(\mathbf{x}'_i, \sigma'_i)$.

6. Conduct pairwise comparison over the union of parents and offspring: For each individual, p opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is better than the opponent's, it receives a "win".

7. Select the μ individuals out of the union of parents and offspring that have the most wins to be parents of the next generation.

8. Stop if the halting criterion is satisfied, and if not, set $k = k + 1$ and go to Step 3.

The Fast Evolutionary Programming (FEP) is described and compared with the CEP in [7]. The FEP is similar to the CEP algorithm, but it performs a mutation following a Cauchy probability density function, instead of a Gaussian based mutation. The one dimensional Cauchy density function centered at the origin is defined by

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2} \tag{7}$$

where $t > 0$ is a scale parameter. See [7] for further information about this topic. Using this probability density function, the FEP algorithm substitutes step 5 of the CEP by the following:

$$\mathbf{x_i'} = \mathbf{x_i} + \sigma_\mathbf{i}\delta \tag{8}$$

where δ is a Cauchy random variable vector with the scale parameter set to $t = 1$.

In order to perform a more accurate search for the current vector of phase differences $\mathbf{x}$, we incorporate a local search procedure to the EP algorithm, to form a hybrid evolutionary programming approach [5].

2.1 Gradient-guided local search procedure

For each phase difference vector $\mathbf{x}$ in the population, the gradient-guided local search works in the following way:

1. Set a maximum number of iterations *max_ite*, and set a counter $k = 1$, and $\hat{\mathbf{x}} = \mathbf{x}$.

2. Calculate the value of the objective function $f(\hat{\mathbf{x}}) = \max\{\varphi_1(\hat{\mathbf{x}}), \cdots, \varphi_{2m}(\hat{\mathbf{x}})\}$.

3. Calculate the gradient of $f(\hat{\mathbf{x}})$, defined as

$$\phi(\hat{\mathbf{x}}) = \frac{\nabla f(\hat{\mathbf{x}})}{\|\nabla f(\hat{\mathbf{x}})\|} \tag{9}$$

4. Set an initial step size ϵ.

5. Modify vector $\hat{\mathbf{x}}$, ϵ units backwards the direction of $\phi(\hat{\mathbf{x}})$:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}} - \epsilon \cdot \phi(\hat{\mathbf{x}}), \tag{10}$$

and calculate then $f(\hat{\mathbf{x}}) = \max\{\varphi_1(\hat{\mathbf{x}}), \cdots, \varphi_{2m}(\hat{\mathbf{x}})\}$.

6. The value of the parameter ϵ is dynamically adapted, in such a way that if the value of function $f(\hat{\mathbf{x}})$ is improved five times in a row, then the parameter ϵ is increased to 1.5ϵ. On the other hand, if the value of function $f(\hat{\mathbf{x}})$ is not improved in any of three times in a row, the parameter is decreased to $\frac{\epsilon}{2}$.

7. If $k < max\_ite$: $k = k + 1$, go to 2. Otherwise: $f(\mathbf{x}) = f(\hat{\mathbf{x}})$, stop.

The gradient-guided local search with dynamic step adaptation performs an accurate minimization in the neighborhood of the initial point $\mathbf{x}$. The fitness associated with the final point reached $(\hat{\mathbf{x}})$ is assigned to the original individual $\mathbf{x}$ in the EP algorithm substituting its initial value.

2.2 Mixing the EP and the gradient-guided local procedure

The gradient-guided local search procedure is applied in the EP after the mutation operator, as a previous step to the selection mechanism. The complete hybrid EP algorithm takes the form given in Section 2, including the following points:

4. $\cdots$

5. Calculate the fitness values associated with each offspring $(\mathbf{x}'_i, \sigma'_i)$.

6. Launch the gradient-guided local search procedure, with starting point the individuals of the Evolutionary Programming $\mathbf{x} = \mathbf{x_i'}$.

7. Replace the fitness value of the population with the fitness function value of the individuals after the application of the gradient-guided local search procedure.

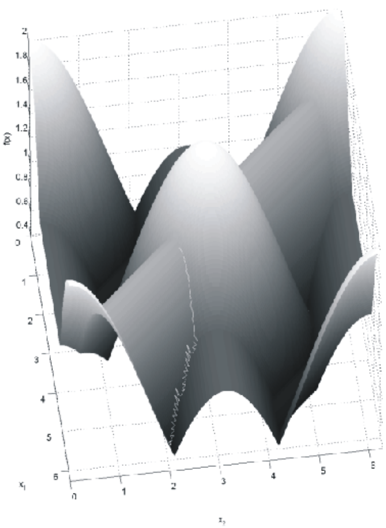
8. Conduct pairwise comparison over the union of parents and offspring: For each individual, p opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is better than the opponent's, it receives a "win".

9. $\cdots$

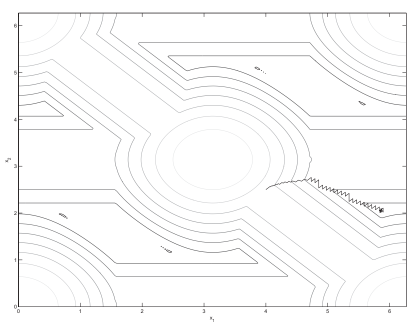
Note that we have described our hybrid EP algorithm using the CEP algorithm, however, we have implemented an *improved FEP* (IFEP) [7], where the Cauchy pdf given by Equation (7) is also used to perform the mutation of the EP in our algorithm, and then the best result obtained between the Gaussian mutation and the Cauchy mutation is selected to complete the process.

3. EXPERIMENTAL PART

The performance of our algorithm has been evaluated in several SSRP instances, with different values of the parameter n. We compare the results obtained by our hybrid

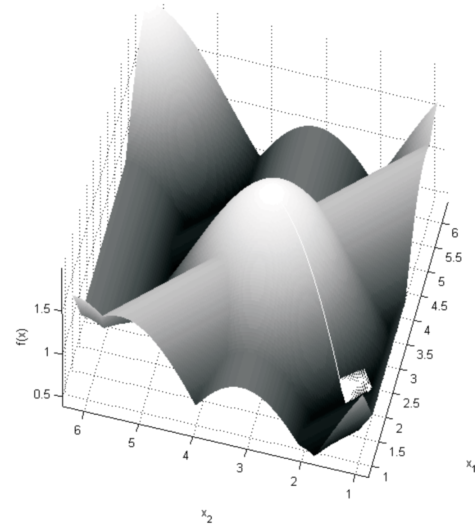


(a)

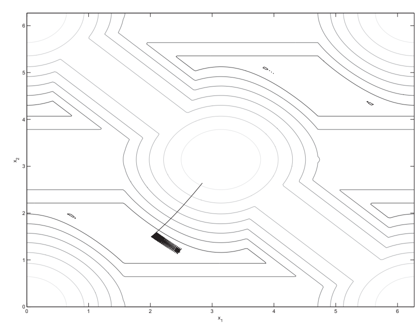


(b)

Figure 2: Example of the gradient-guided procedure's performance; (a) 3D view; (b) View in a map of equal height lines.



(a)



(b)

Figure 3: Example of the gradient-guided procedure's performance starting from a different initial point; (a) 3D view; (b) View in a map of equal height lines.

EP algorithm with the performance of several algorithms: a Tabu Search approach [3], a variable neighborhood search approach [3], and a genetic algorithm described in [8]. All the algorithms have been implemented for a precision of $N = 32$ bits. First we describe the main characteristics of these algorithms and then we show the results obtained with the hybrid EP and the comparison approaches.

3.1 Implemented algorithms for comparison

3.1.1 Tabu search

Tabu search is a metaheuristic that guides the local exploration of a given search space by means of a "tabu list", or forbidden moves to previous low-quality solutions, to avoid local optimum solutions [11], [12], [13]. A search begins with a feasible solution usually produced with another heuristic. For each feasible solution, a class of neighbor solutions is generated by a move or a perturbation. A move is generally obtained by changing some of the attributes of the current solution. The aim of moves is to search for better solutions. In order to prevent the algorithm from deterministically cycle among solutions already visited, one or more of the attributes of the current solution are kept in the tabu list as *forbidden moves*. The number of moves in the list is referred to as the *tabu length*. The exclusion period that a move remains in tabu is called the *tabu tenure* of it. The key to TS is to force a problem-specific local search heuristic to accept "uphill moves" to escape local optimality and sometimes to jump to a totally new region for carrying on searching.

When a better, not forbidden, solution is found, the current solution is replaced and the search continues from it. If all moves are worse than the current solution or the better ones are forbidden, the best among those that are not forbidden is selected. In this way, the algorithm is able to accept a worse solution to avoid being trapped in local optima. The best solution that has been found so far in the search is labelled as "best" to avoid lost. If a move is better than the "best", it is selected as a new "best". However, if it is forbidden, the *aspiration criterion* enables the tabu status to be overridden, and thus a new "best" is established too. This process is repeated for a pre-specified number of iterations then the "best" visited so far is designated as the optimal solution to the problem.

A Tabu search approach mixed with a local search algorithm has been successfully proposed for the SSRP in [6] and [3].

3.1.2 Variable neighborhood search

Variable Neighborhood Search (VSN) is a recently proposed metaheuristic [14], [15] for solving combinatorial optimization problems. It is based on the use of more than a neighborhood structure (tabu search uses only one, for example), and to proceed to a systematic change of them within a local search. The algorithm retains the best solution found so far, until another better solution is found. Neighborhoods are usually ranked, in such a way that intensification of the search around the current solution is followed naturally by diversification. Results on the application of Variable neighborhood search to the SSRP can be found in [3].

3.1.3 Genetic algorithm

Genetic Algorithms (GAs) have been widely used as global search heuristics in hybrid algorithms [9], [10]. In our case, the standard GA is used, which applies the following basic operators:

- Selection, where the individuals of a new population are selected from the old one. In the standard implementation of the Selection operator, each individual has a probability of surviving for the next generation proportional to its associated objective function value (roulette wheel).
- Crossover, where new individuals are searched starting from couples of individuals in the population. Once the couples are randomly selected, the individuals have the possibility of swapping parts of themselves with its couple, the probability of this happens is usually called *crossover probability*, P_c.
- Mutation, where new individuals are searched by randomly changing bits of current individuals with a low probability P_m (*probability of mutation*).

A genetic algorithm has been applied to the SSRP in [8].

3.2 Results

Before presenting the results obtained by the compared algorithms, we show two examples of how the guided-local search procedure used in our hybrid EP works, for the case of $n = 2$. Figures 2 and 3 show the two examples of the procedure performance, for two different values of the starting point $\mathbf{x}$ (note that the final local solution found by the gradient-guided procedure completely depends on the initial starting point $\mathbf{x}$). In both cases (Figures 2 (b) and 3 (b) it can be seen that the moves in $\mathbf{x}$ follow the direction of a line orthogonal to the equal height lines of the function $f(\mathbf{x})$. Figures 2 (a) and 3 (a) display the same cases in a 3D view. Figure 4 shows how is the evolution of the objective function value ($n = 2$) within the 200 iterations of the local search. Figure 4 (a) shows this evolution for the case of Figure 2, whereas Figure 4 (b) shows it for the case of Figure 3. It is easy to appreciate that when a flat region is reached, the objective function values quickly vary in a narrow band, and finally, the best local minimum is reached in both cases.

Table 1 presents the values of the objective function, calculated by means of Equation (1), obtained by the four compared algorithms. The hybrid EP (HEP in the table) algorithm produces better results than the compared approaches in all considered cases (the best values obtained by comparing all the considered algorithms have been marked with a † symbol). Optimal known values of the objective function are obtained with the hybrid EP for problems $n = 2$, 3, 4 and 5. Table 2 shows a summarize of the results obtained with our hybrid approach, including the best and worst objective function values found, and the mean and standard deviation of the 20 experiments run for each value of n.

Table 1: Results obtained with the different methods compared in the SSRP instances tackled, for different values of vector of phase differences' dimension (n). Symbol * stands for the optimum known values of the objective function, obtained by means of an Implicit Enumeration Technique [7], in problems $n = 1$ to $n = 5$. Symbol † stands for the best over all values found by the algorithms.

n	TS	VNS	GA	HEP
2	$0.3852^{*\dagger}$	$0.3852^{*\dagger}$	$0.3852^{\dagger *}$	$0.3852^{*\dagger}$
3	0.2611	$0.2610^{*\dagger}$	$0.2610^{*\dagger}$	$0.2610^{*\dagger}$
4	0.0573	0.0562	$0.0560^{*\dagger}$	$0.0560^{*\dagger}$
5	0.3404	0.3375	0.3374	$0.3371^{*\dagger}$
6	0.4574	0.4562	0.4645	$0.4556^{\dagger}$
7	0.5114	0.4972	0.5232	$0.4963^{\dagger}$
8	0.4130	0.3871	0.4328	$0.3854^{\dagger}$
9	0.3548	0.3290	0.3386	$0.3208^{\dagger}$
10	0.4568	0.4105	0.4709	$0.4067^{\dagger}$

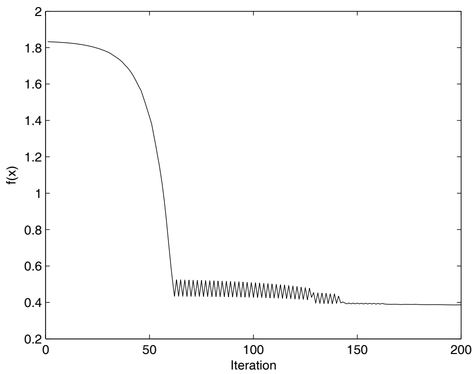
Table 2: Results obtained with the hybrid evolutionary programming proposed.

n	best	worst	mean	std. dev.
2	0.3852	0.3852	0.3852	0
3	0.2610	0.2610	0.2610	0
4	0.0560	0.0560	0.0560	0
5	0.3371	0.3371	0.3371	0
6	0.4556	0.4556	0.4556	0
7	0.4963	0.4963	0.4963	10^{-6}
8	0.3854	0.3854	0.3854	0
9	0.3208	0.5632	0.3387	0.0578
10	0.4067	0.5405	0.4346	0.0291

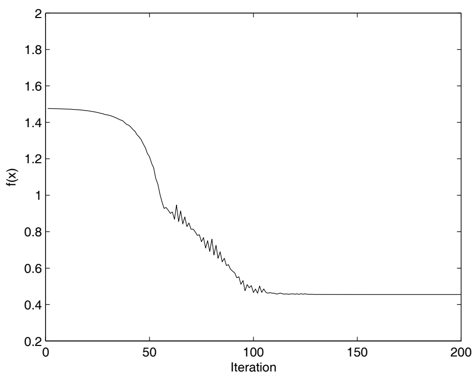
The effect of including the local search procedure into the EP can be analyzed by comparing the results of Table 2 with the results obtained using the EP algorithm without local search. Table 3 shows the results obtained by the EP algorithm without gradient local search applied. The results reported in Table 2 are more accurate than the results in Table 3, as expected. Finally, the best phase differences obtained for each instance are provided in Table 4.

4. CONCLUSIONS

This paper proposes a hybrid evolutionary programming approach to the spread spectrum radar polyphase codes design problem (SSRP). The algorithm is formed by a fast evolutionary programming hybridized with a gradient-guided



(a)



(b)

Figure 4: Evolution of the objective function value when the local search is applied; (a) Local search starting from point 1 (case of Figure 2); (b) Local search starting from point 2 (case of Figure 3).

Table 3: Results obtained with the evolutionary programming without local search.

n	best	worst	mean	std. dev.
2	0.3852	0.3852	0.3852	0
3	0.2610	0.2610	0.2610	0
4	0.0560	0.0562	0.0560	$6 \cdot 10^{-5}$
5	0.3371	0.4798	0.3782	0.0493
6	0.4557	0.6476	0.5078	0.0668
7	0.4969	0.8165	0.6079	0.1053
8	0.4284	0.9097	0.6428	0.1270
9	0.3682	1.0592	0.7878	0.1521
10	0.6276	1.0725	0.8202	0.1269

local search procedure to improve the search space exploration. Simulations in several SSRP instances have shown that our approach improves the performance of previous approaches to this problem.

5. ACKNOWLEDGMENTS

This work has been partially supported by Comunidad de Madrid, Universidad de Alcalá and Ministerio de Ciencia y Tecnología, with projects numbers CAM-UAH 2005/019, UAH-PI2005/078 and TEC2006-07010.

Table 4: Phase differences of the best solution for different values of n.

n	Phase differences $(x_1, \cdots, x_n)$
2	2.4501, 1.1754
3	1.0726, 3.1642, 1.3068
4	5.6180, 1.4198, 2.5587, 1.6268
5	5.9674, 2.0926, 1.7137, 3.3218, 1.2270
6	5.0217, 1.8553, 4.4867, 4.1258, 4.8854, 4.8386
7	3.9114, 4.2917, 3.4892, 4.1971, 5.6431, 5.7602, 5.2317
8	2.4993, 4.5954, 4.2632, 2.8861, 4.8819, 5.9198, 5.2289, 5.0556
9	3.4887, 0.3730, 3.6832, 3.3446, 4.7457, 6.2203, 4.8504, 6.3198, 4.4955
10	0.9777, 4.8338, 5.1645, 1.0583, 4.9819, 2.6894, 1.7569, 0.6754, 2.4862, 1.9496

6. REFERENCES

[1] B. Lewis, F. Kretschmer Jr. and W. Shelton *Aspects of radar signal processing*, Artech House, 1986.

[2] M. Dukic and Z. Dobrosavljevic, "A method of spread-spectrum radar polyphase code design," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 743-749, 1990.

[3] N. Mladenovic, J. Petrovic, V. Kovacevic-Vujcic and M. Cangalovic, "Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search," *European Journal of Operational Research*, vol. 151, pp. 389-399, 2003.

[4] E. Erozden and O. Akay, "Use of fractional autocorrelation for efficient detection and parameter estimation of polyphase-coded radar signals," In *Proc of the IEEE Signal Processing and Communications Applications Conference*, pp. 41-44, 2004.

[5] B. Popovic, "Complementary sets of chirp-like polyphase sequences Popovic," *Electronics Letters*, vol. 27, no. 3, pp. 254-255, 1991.

[6] M. Asic, M. Cangalovic, V. Kovacevic-Vujcic and M. Ivanovic and M. Drazic, "An application of Tabu search to spread spectrum radar polyphase code design," In *Proc. of the 23rd Yuguslav Symposium on Operations Research*, pp. 401-404, 1996.

[7] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput*, vol. 3, no. 2, pp. 82-102, 1999.

[8] J. Kratica, D. Tosic, V. Filipovic and I. Ljubic, "Genetic Algorithm for Designing a Spread-Spectrum Radar Polyphase Code," In *Proc. of the 5th Online World Conference on Soft Computing Methods in Industrial Applications*, pp. 191-197, 2000.

[9] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading:MA, 1989.

[10] N. Krasnogor and J. Smith, “A tutorial for competent memetic algorithms: model, taxonomy, and design issues,” *IEEE Trans. Evol. Comput*, vol. 9, no. 5, pp. 474-488, 2005.

[11] F. Glover, “Tabu search – part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190-206, 1989.

[12] F. Glover, “Tabu search – part II,” *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4-32, 1990.

[13] F. Glover and M. Laguna, *Tabu search*, Kluwer Academic Press, 1997.

[14] P. Hansen and N. Mladenovic, “An introduction to VNS,” In *Metaheuristics Advances and Trends in Local Search Paradigm for Optimization*, Kluwer, pp. 433-458, 1998.

[15] P. Hansen and N. Mladenovic, “Variable neighborhood search: principles and applications,” *European Journal of Operational Research*, vol. 130, pp. 449-467, 2001.