Expert Systems with Applications 39 (2012) 11089–11093

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid harmony search algorithm for the spread spectrum radar polyphase codes design problem

Sergio Gil-López [a], Javier Del Ser [a], Sancho Salcedo-Sanz [b,*], Ángel M. Pérez-Bellido [b], José María Cabero [a], José A. Portilla-Figueras [b]

[a] Tecnalia, Telecom Unit, P. Tecnológico, Ed. 202, E-48170, Zamudio, Bizkaia, Spain
[b] Department of Signal Theory and Communications, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain

ARTICLE INFO

Keywords:
Polyphase code design
Harmony search
Soft-computing techniques
Gradient-guided local search

ABSTRACT

In this paper we present the application of a hybrid harmony search (HS) algorithm to the Spread-Spectrum Radar Polyphase (SSRP) codes design. Such a design can be formulated as a non-linear max–min optimization problem, hard to be solved using classical numerical techniques. Soft-computing approaches have then been successfully applied to solve the SSRP in the past, such as evolutionary computation techniques, variable neighborhood approaches or tabu search algorithms. In this paper we elaborate on the proposed hybrid HS approach, which consists of a naive implementation of the HS algorithm along with an adaptive-step gradient-guided local search procedure. Intensive computer simulations show that the proposed hybrid HS algorithm is able to outperform existing algorithms for the SSRP design problem (including the best reported so far), with significant differences in large-size SSRP instances.

1. Introduction

In radar systems with pulse compression, the choice of the appropriate waveform is a major problem which has been profusely treated in the literature (Dukic & Dobrosavljevic, 1990; Mladenovic, Petrovic, Kovacevic-Vujcic, & Cangalovic, 2003). Several pulse compression methods have been proposed such as Barker codes, chirp-type modulation or polyphase codes (Lewis, Kretschmer, & Shelton, 1986). Among these methods for radar pulse modulation, the polyphase codes offer several advantages in comparison to other techniques, such as chirp-type modulations (Dukic & Dobrosavljevic, 1990): polyphase codes produce lower side-lobes in the compressed signal, and easier digital processing techniques implementation.

It is important to observe that the resolution of radar systems can be significantly improved by using short pulses (Mow, 1995). The drawback is that the use of short pulses decreases the average transmitted power in the system, which can ultimately affect the radar's normal modes of operation. In order to overcome this issue, the majority of modern radar systems generally incorporate pulse compression waveforms, since they allow achieving the average transmitted power of a relatively long-pulse radar scheme, while simultaneously obtaining the range resolution of a short-pulse radar. In Dukic and Dobrosavljevic (1990), Dukic and Dobrosavljevic introduced a new method for polyphase pulse-compression code design, based on the properties of their aperiodic autocorrelation function, and also on considering coherent radar pulse processing at the receiver. This method is specially interesting since it can be modeled as a min–max nonlinear optimization problem, defined as follows:

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

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

where $m = 2n - 1$, and

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

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

$$\varphi_{m+i}(\mathbf{x}) = -\varphi_i(\mathbf{x}), \quad i = 1, \ldots, m. \tag{5}$$

In the definition of this problem the variables x_k represent symmetrized phase differences, and the objective of the problem is to minimize the module of the largest among the samples of the autocorrelation function φ.

This optimization problem associated with the design of polyphase codes in radar systems using this model is usually called Spread-Spectrum Radar Polyphase (SSRP) code design problem (SSRP). The SSRP has been tackled by using different – mainly

* Corresponding author. Address: Department of Signal Theory and Communications, Universidad de Alcalá. 28871 Alcalá de Henares, Madrid, Spain. Tel.: +34 91 885 6731; fax: +34 91 885 6699.
E-mail address: sancho.salcedo@uah.es (S. Salcedo-Sanz).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.063

meta-heuristic – approaches in the literature. As such, in Asic, Cangalovic, Kovacevic-Vujcic, and Ivanovic (1996) a Tabu search approach has been proposed to solve this problem. In Mladenovic et al. (2003) several heuristic algorithms to the SSRP are proposed and compared with each other, including a multi-level heuristic tabu search and a variable neighborhood algorithm. In Kratica, Tosic, Filipovic, and Ljubic (2000) a genetic algorithm is able to improve the results obtained by the Tabu search in several SSRP instances, and finally in Pérez-Bellido, Salcedo-Sanz, Ortiz-García, Portilla-Figueras, and López-Ferreras (2008), Pérez-Bellido, Salcedo-Sanz, Ortiz-García, and Portilla-Figueras (2007) an Evolutionary Programming (EP) algorithm was successfully applied to this problem.

This manuscript advances over previous avantgarde approaches by presenting a novel hybrid algorithm for the SSRP inspired by the harmony search (HS) optimization algorithm. This is a new meta-heuristic approach recently proposed in Geem, Kim, and Loganathan (2001), which has heretofore rendered better results than other meta-heuristic approaches indifferent optimization problems (Wang & Huang, 2010; Ahmad, Mohammad, Salman, & Hamdan, 2012). In the SSRP problem considered herein, a naive implementation of the HS algorithm is hybridized with a local search method aimed at refining the search. We report intensive simulation results for several SSRP instances of different size, jointly with a comparison with existing approaches, which verify that the proposed hybrid HS algorithm dominates not only in terms of the statistical behavior of successive Monte Carlo realizations of the algorithm, but also in what relates to the complexity of the considered techniques.

The rest of this paper is structured as follows: Section 2 thoroughly describes the hybrid HS algorithm proposed in this paper, whereas Section 3 presents several computer simulations for different SSRP instances to show the performance of our approach. Comparisons with the results obtained by a tabu search approach, a variable neighborhood search and an existing genetic algorithm are also provided in this chapter. Finally, Section 4 draws some concluding remarks in light of the obtained results.

2. Proposed hybrid harmony search SSRP code design algorithm

Harmony search (HS) is a meta-heuristic algorithm first proposed by Geem et al. in Geem et al. (2001), and thenceforth successfully applied to a wide range of optimization problems in very diverse fields such as design of water pipeline networks (Geem, 2006; Geem, 2008), multicast routing (Forsati, Haghighat, & Mahdavi, 2008), vehicle routing (Geem, Lee, & Park, 2005), scheduling of multiple dam systems (Geem, 2007), the so-called Switch Location Problem (SLP, (Gil-Lopez, Del Ser, Landa, & Salcedo-Sanz, 2010; Landa-Torres, Gil-Lopez, Salcedo-Sanz, Del Ser, & Portilla-Figueras, 2012)), spectrum channel allocation in cognitive radio networks (Del Ser, Matinmikko, Gil-Lopez, & Mustonen, 2010), flow shop scheduling problems (Pan, Suganthan, Liang, & Tasgetiren, 2011) or even as an efficient means to solve the *Sudoku* puzzle (Geem, 2008). HS is inspired by the improvisation process of an orchestra in their attempt to compose the most harmonious melody under an esthetic point of view.

The inherent good balance between explorative and exploitative behavior of the HS algorithm is due to the involvement of three operational processes in the refinement of a set of potential candidate solutions: the Harmony Memory Considering Rate (HMCR), the Pitch Adjusting Rate (PAR) and the Random Selection Rate (RSR). By imposing a strong explorative behavior on the algorithm, wider regions of the solution space are explored by evaluating distant points with no further search made on any of them, which in general yields slow convergence speed. On the other hand, a exploitative behavior will imply that the algorithm utilizes the knowledge acquired at previous stages for searching on regions where potentially *good* solutions are located. Typically, a well-designed meta-heuristic search strategy assumes a strong explorative behavior of the algorithm at the beginning of the iterative process (trying to collect information of the solution space). However, as the algorithm evolves attention is placed on the regions where the best solutions are located (i.e. the exploitative behavior prevails in the algorithmic search method).

Following the notation proposed by Geem et al. (2001), we denote a solution vector of phase differences $\mathbf{x}(i)$ $(i \in \{1,\ldots,\xi\})$ as *harmony*, whereas *note* will stand for any of its compounding elements $x_j(i)$, $\forall j \in \{1,\ldots,n\}$. Harmony search works with a set of ξ possible solution vectors or *harmonies* commonly denoted as Harmony Memory (HM), which are refined through the three aforementioned refining processes, and then successively evaluated at each iteration under a predefined fitness function. The HM is updated whenever any of the new ξ improvised harmonies at a given iteration sounds *better* (under the fitness criterion) than any of the remaining ξ harmonies from the previous iteration. This procedure is iteratively repeated until the best harmony is reached or alternatively, until a fixed number of attempts or iterations are completed.

Schematically, the meta-heuristic HS search process consists of four steps:

A. The initialization of the notes' values of all harmonies included in the HM is only executed at the first iteration. Since no a priori knowledge of the solution is assumed at this point, the notes values $x_j(i)$ $\forall i \in \{1,\ldots,\xi\}$ and $j \in \{1,\ldots,n\}$ (i.e. the magnitudes of the phase differences for the SSRP) are filled with realizations of a random variable uniformly distributed in the range $[0, 2\pi)$.

B. The improvisation process is sequentially applied to each note of the complete set of ξ harmonies. As opposed to the nominal HS scheme, where only two parameters (HMCR, PAR) are considered for improvising the new refined set of harmonies, three arbitrary parameters are used in this contribution to get more control on the trade-off between the explorative and exploitative behavior of the algorithm:

   - The Harmony Memory Considering Rate, HMCR $\in \mathbb{R}[0,1]$, which establishes the probability that the new value for a note is drawn from the values of the same note taken in all the other $\xi - 1$ harmonies existing in the Harmony Memory. Notice that the smaller HMCR is, the less the use of partial knowledge acquired during the iterative process will be, and hence the more explorative the algorithm will behave.
   - The Random Selection Rate, RSR $\in \mathbb{R}[0,1]$, which measures the probability that the proposed new value for a note is drawn from an uniform distribution in the range $[0, 2\pi)$. Observe that in general RSR $\neq 1 -$ HMCR, i.e. it will be set different than the complementary probability $1 -$ HMCR used by the nominal HS algorithm. This parameter contributes to the diversity of the proposed solutions by exploring wider regions of the solution space through the insertion of randomly drawn note values (which may in turn favor not to stall in local minima).
   - The Pitch Adjusting Rate, PAR $\in \mathbb{R}[0,1]$ acts as a fine adjusting rate of the note vocabulary by defining the probability that the new value for a given note is picked from its neighborhood in the range $[0, 2\pi)$. In other words, this probabilistic parameter defines the new value $x_j^*(i)$ for a certain note $x_j(i)$ (after HMCR and RSR processing) as

$$x_j^*(i) = \begin{cases} x_j(i) + \beta \cdot z & \text{with probability PAR}, \\ x_j(i) & \text{with probability } 1 - \text{PAR}, \end{cases} \quad (6)$$

where z is the realization of an uniform random variable Z with continuous support in the range $[-1,1]$, and $\beta \in \mathbb{R}^+$ imposes the bandwidth for the pitch adjustment. Analogously to the HMCR parameter, a high value of PAR jointly with a increased value of β sets a highly explorative behavior of the algorithm around the iteratively-identified potential candidates or harmonies, while narrower bandwidths (i.e. lower values of β) the PAR refinement reduces to a restricted local search procedure.

C. If any improvised note $x_j^*(i)$ is out of the range $[0,2\pi)$ then the algorithm finally improvises a new note as $x_j^\dagger(i) = \text{mod}\left(x_j^*(i), 2\pi\right)$. Otherwise, $x_j^\dagger(i) = x_j^*(i)$, i.e. no modulus operation is performed.[1]

D. At each iteration the quality of the improvised harmonies is evaluated by means of the fitness function in Expression (1). Then, based on these metric evaluations and their comparison with the fitness of harmonies remaining from the previousiteration, the ξ best harmonies are kept and the Harmony Memory is hence updated by excluding the worst harmonies.

E. The stopping criterion is selected based on a fixed number of iterations $\mathcal{T}$.

2.1. Adaptive-step gradient-guided local search procedure

For each phase difference vector $\mathbf{x}(i)$ in the harmony memory HM ($i \in \{1,\ldots,\xi\}$), a gradient-guided local search with dynamic step adaptation is used to perform an accurate minimization in the neighborhood of every initial point $\mathbf{x}(i)$. The application of the local search is carried out in the proposed HS global search algorithm before a newly improvised harmony is evaluated. Such algorithm operates as follows:

(1) Set a maximum number of iterations $\mathcal{K}_{max}$, a counter $k = 1$, an initial step size ϵ (set to 0.5 for each candidate vector $\mathbf{x}(i)$), and $\hat{\mathbf{x}}(i) = \mathbf{x}(i)$.

(2) Calculate the value of the objective function

$$f(\hat{\mathbf{x}}(i)) = \max\{\varphi_1(\hat{\mathbf{x}}(i)), \ldots, \varphi_{2m}(\hat{\mathbf{x}}(i))\}.$$

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

$$\phi(\hat{\mathbf{x}}(i)) \doteq \frac{\nabla f(\hat{\mathbf{x}}(i))}{\|\nabla f(\hat{\mathbf{x}}(i))\|} \quad (7)$$

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

$$\hat{\mathbf{x}}(i) = \hat{\mathbf{x}}(i) - \epsilon \cdot \phi(\hat{\mathbf{x}}(i)), \quad (8)$$

and compute again $f(\hat{\mathbf{x}}(i)) = \max\{\varphi_1(\hat{\mathbf{x}}(i)), \ldots, \varphi_{2m}(\hat{\mathbf{x}}(i))\}$.

(5) The value of the parameter ϵ is dynamically adapted in such a way that if the value of function $f(\hat{\mathbf{x}}(i))$ is improved five times in a row, then the parameter ϵ is increased to 1.5ϵ. Otherwise, if the value of function $f(\hat{\mathbf{x}}(i))$ is not improved in any of three times in a row, the parameter is decreased to $\frac{\epsilon}{2}$. After $\mathcal{K}$ iterations $(1 < \mathcal{K} < \mathcal{K}_{max})$ the ϵ value is kept fixed to 0.05.

(6) If $k < \mathcal{K}_{max}$: $k = k + 1$, go to 2. Otherwise, the algorithm is stopped and the harmony (in the case of a Lamarckian[2] approach) and its fitness are updated to $\hat{\mathbf{x}}(i)$ and $f(\hat{\mathbf{x}}(i))$, respectively.

[1] This produces a slightly computational time reduction by only executing the mod $(\cdot)$ operation when needed.

[2] The authors refer to Pérez-Bellido et al. (2008) for a detailed insight on the Lamarckian and Baldwinian concepts.

Table 1
Comparative results of the proposed hybrid HS algorithm against a hybrid Evolutionary Programming approach in different SSRP instances. Both algorithms include the same local search heuristic and are running in Lamarckian mode.

n	Best value		Mean value		Std. deviation	
	IFEP	HS	IFEP	HS	IFEP	HS
2	0.3854	0.3852	0.3855	0.3852	0.0001	0.0000
3	0.2611	0.2610	0.2614	0.2610	0.0002	0.0000
4	0.0565	0.0560	0.0567	0.0560	0.0002	0.0000
5	0.3427	0.3371	0.3458	0.3371	0.0028	0.0000
6	0.4621	0.4556	0.4630	0.4527	0.0006	0.0005
7	0.5031	0.4964	0.5068	0.4966	0.0026	0.0001
8	0.4218	0.3855	0.4310	0.3856	0.0120	0.0002
9	0.3832	0.3211	0.3860	0.3307	0.0037	0.0398
10	0.4773	0.4075	0.4856	0.4280	0.0086	0.0055
11	0.5752	0.3661	0.6028	0.3791	0.0343	0.0331
12	0.5881	0.4624	0.6168	0.5016	0.0226	0.0206
13	0.5762	0.4525	0.6587	0.4908	0.0627	0.0611
14	0.8013	0.4365	0.8434	0.5389	0.0379	0.0736
15	0.8462	0.4424	0.8988	0.5797	0.0451	0.0738

3. Experimental results

The performance of the proposed hybrid HS scheme has been evaluated in several SSRP instances with different values of the parameter n, i.e. the parameter that sets the dimensions of the SSRP code. We compare the statistical performance results obtained by our approach with those of several existing approaches in the related literature: a Tabu Search scheme (Mladenovic et al., 2003), a variable neighborhood search technique (Mladenovic et al., 2003), a Genetic Algorithm as described in Kratica et al. (2000), and a hybrid Evolutionary Programming algorithm (Pérez-Bellido et al., 2008). All the algorithms have been implemented for a precision of 32 bits.

Regarding the set of parameters driving the HS algorithm, in all the considered experiments we have utilized the same set of values, namely: $\{\text{HMCR}, \text{PAR}, \text{RSR}\} = \{0.6, 0.6, 0.1\}$. Furthermore, the bandwidth value is adjusted dynamically and linearly through the iterations $t \in \{1, \ldots, \mathcal{T}\}$ from 0.3 ($t = 1$) to 0.1 ($t = \mathcal{T}$). The size of the harmony memory ξ has been set to 50, which ensures a fair comparison in terms of number of objective function evaluations against the rest of the considered techniques. The maximum number of iterations of the algorithms is kept fixed to $\mathcal{T} = 300$. As for the local search procedure, $\mathcal{K}$ and $\mathcal{K}_{max}$ are set equal to 100 and 200, respectively.

3.1. Results

First of all we carry out a comparative analysis of the proposed hybrid HS algorithm versus an EP algorithm (Pérez-Bellido et al., 2008), both subject to the same local search procedure explained in Subsection 2.1. This analysis will give us an idea of the real performance of the naive HS algorithm when applied to the SSRP design problem at hand, since the local search is the same for both algorithms. Tables 1 and 2 show the mean, minimum and standard deviation of the objective function computed over 30 different realizations of the algorithm for both the HS and the EP approaches working in both Lamarckian and Baldwinian modes. It should be emphasized that we have implemented a version of the Improved Fast Evolutionary Programming (IFEP) algorithm as described in Yao, Liu, and Lin (1999). Please recall from (Pérez-Bellido et al., 2008) that in the Lamarckian mode, the local search modifies the successively proposed candidate solutions in the global search algorithms, whereas in the Baldwinian mode the local search does not modify such solutions in the global search algorithm, but only the value of their objective function. Tables 1 and 2 assess that the proposed HS algorithm performs equal to or better than the IFEP

Table 2
Comparative results of the proposed hybrid HS algorithm against a hybrid Evolutionary Programming approach in different SSRP instances. Both algorithms include the same local search heuristic and are running in Baldwinian mode.

n	Best value		Mean value		Std. deviation	
	IFEP	HS	IFEP	HS	IFEP	HS
2	0.3853	0.3852	0.3853	0.3852	0.0000	0.0000
3	0.2612	0.2610	0.2613	0.2610	0.0002	0.0000
4	0.0570	0.0560	0.0571	0.0560	0.0001	0.0000
5	0.3422	0.3371	0.3433	0.3371	0.0010	0.0000
6	0.4633	0.4556	0.4641	0.4556	0.0006	0.0000
7	0.5064	0.4964	0.5077	0.4966	0.0009	0.0001
8	0.4187	0.3855	0.4330	0.3856	0.0133	0.0001
9	0.3601	0.3211	0.3959	0.3221	0.0291	0.0005
10	0.4936	0.4078	0.4984	0.4270	0.0047	0.0100
11	0.5480	0.3663	0.5650	0.4005	0.0178	0.0447
12	0.6198	0.4748	0.6383	0.5013	0.0164	0.0240
13	0.7020	0.4560	0.7205	0.5242	0.0140	0.0611
14	0.7165	0.4549	0.7728	0.5623	0.0407	0.0719
15	0.8749	0.5089	0.9204	0.6773	0.0339	0.1118

Table 3
Comparison of the best results obtained in different SSRP instances by the proposed hybrid HS algorithm and several existing approaches in the literature.

n	TS Mladenovic et al. (2003)	VNS Mladenovic et al. (2003)	GA Kratica et al. (2000)	IFEP Pérez-Bellido et al. (2008)	HS
2	0.3852	0.3852	0.3852	0.3853	0.3852
3	0.2611	0.2610	0.2610	0.2611	0.2610
4	0.0573	0.0562	0.0560	0.0565	0.0560
5	0.3404	0.3375	0.3374	0.3422	0.3371
6	0.4574	0.4562	0.4645	0.4621	0.4556
7	0.5114	0.4972	0.5232	0.5031	0.4964
8	0.4130	0.3871	0.4328	0.4187	0.3855
9	0.3548	0.3290	0.3386	0.3601	0.3211
10	0.4568	0.4105	0.4709	0.4773	0.4075
11	0.4024	0.4318	0.6387	0.5480	0.3661
12	0.5505	0.4907	0.6786	0.5881	0.4624
13	0.6666	0.4899	0.8307	0.5762	0.4525
14	0.5565	0.4746	1.0042	0.7165	0.4365
15	0.7488	0.4857	0.9674	0.8462	0.4424

approach in all the instances tested. In the Lamarckian running mode, the differences are small in the SSRP instances with small n, but they become more significant as n increases, specially for $n > 10$. The same conclusion holds for the algorithms in the Baldwinian running mode: significant differences can be found between the HS and the IFEP approaches, differences that become sharper as the size of the SSRP instance increases.

Table 3 shows a comparison among the best (minimum) values of the objective function obtained, over 30 realizations, by the different algorithms compared in this paper, i.e. a Tabu Search (TS), a Variable Neighborhood Search (VNS), a Genetic Algorithm (GA),

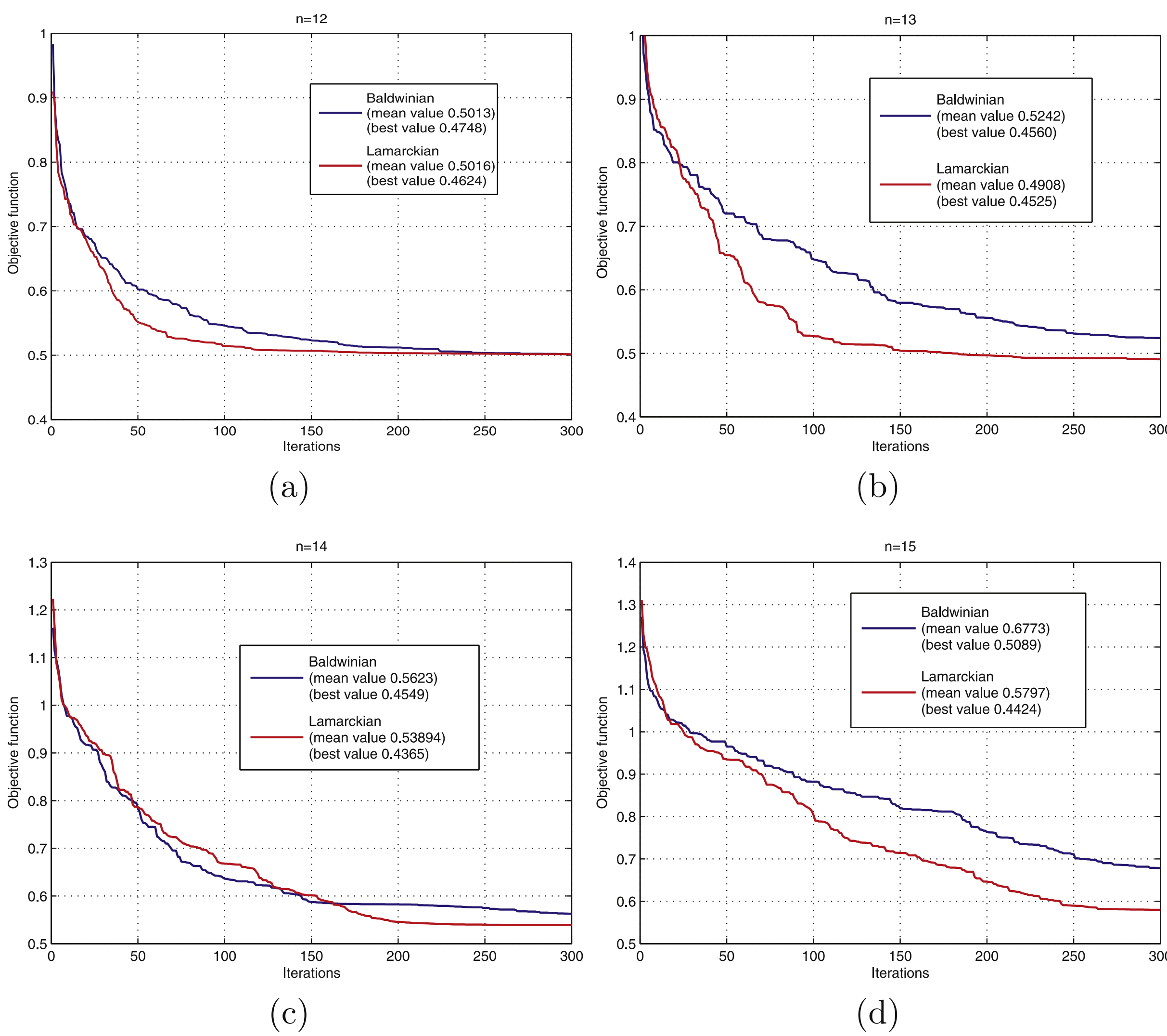

Fig. 1. Mean evolution (over 30 runs) of the proposed HS algorithm in SSRP instances $n = 12$ (a), $n = 13$ (b), $n = 14$ (c) and $n = 15$ (d).

and finally the aforementioned IFEP and HS approaches. Notice that the proposed HS algorithm outperforms the other existing approaches in all the simulated SSRP instances. The differences are again remarkable for relatively large n, specially for $n > 10$. The VNS approach described in Mladenovic et al. (2003) also provides good solutions to the SSRP problem; however, the proposed HS approach render equal or better solutions in all the SSRP instances.

Finally, Fig. 1 depicts the evolution of the mean value of the objective function (averaged over 30 realizations) achieved by the proposed hybrid HS algorithm for $n \in \{12, 13, 14, 15\}$ in both Baldwinian and Lamarckian modes. In these figures observe that in average, the convergence rate of the proposed HS algorithm is higher in the Lamarckian mode.

4. Conclusions

This paper presents a hybrid harmony search (HS) algorithm for a NP-hard optimization problem arising in radar signal processing: the SpreadSpectrum Radar Polyphase (SSRP) code design problem. Although this problem has been traditionally tackled by using meta-heuristics approaches, to the author's knowledge no application of the HS algorithm to this problem has been so far reported in the literature. We have shown that our novel HS approach is a robust algorithm capable to outperform previous heuristic approaches tailored for this specific optimization problem. Experiments carried out for several SSRP instances have assessed that the proposed HS algorithm renders better statistical results that the aforementioned existing approaches, specially for high values of the code size n. These results buttress the outstanding scalability properties of the novel hybrid HS algorithm proposed herein.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation, under project TEC2011–28250-C02–02, the CONSOLIDER-INGENIO 2010 (CSD200800010) and the Torres-Quevedo (PTQ-09–01-00740, PTQ-06–01-0159) funding programmes, and by the Basque Government through the ETORTEK programme (*Future Internet* EI08–227 project).

References

Ahmad, I., Mohammad, M. Gh., Salman, A. A., & Hamdan, S. A. (2012). Broadcast scheduling in packet radio networks using harmony search algorithm. *Expert Systems with Applications, 39*(1), 1526–1535.

Asic, M., Cangalovic, M., Kovacevic-Vujcic, V., Ivanovic, M. & Drazic M. (1996). An application of tabu search to spread spectrum radar polyphase code design. In *Proceedings of the 23rd Yuguslav Symposium on Operations Research*, (pp. 401–404).

Del Ser, J., Matinmikko, M., Gil-Lopez, S., & Mustonen, M. (2010). A novel harmony search based spectrum allocation technique for cognitive radio networks. In *IEEE International Symposium on Wireless Communication Systems, September*

Dukic, M., & Dobrosavljevic, Z. (1990). A method of spread-spectrum radar polyphase code design. *IEEE Journal on Selected Areas in Communications, 8*, 743–749.

Forsati, R., Haghighat, A. T., & Mahdavi, M. (2008). Harmony search based algorithm for bandwidth-delay-constrained least-cost multicast routing. *Computer Communications, 31*, 2505–2519.

Geem, Z. W. (2006). Optimal cost design of water distribution networks using harmony search. *Engineering Optimization, 38*, 259–277.

Geem, Z. W. (2007). Optimal scheduling of multiple dam system using harmony search algorithm. *Lecture Notes on Computer Science, 4507*, 316–323.

Geem, Z. W. (2008). Novel derivative of harmony search algorithm for discrete design variables. *Applied Mathematics and Computation, 199*(1), 223–230.

Geem, Z. W. (2008). Harmony search algorithm for solving sudoku. *Lecture Notes on Computer Science, 4692*, 371–378.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68.

Geem, Z. W., Lee, K. S., & Park, Y. (2005). Aplication of harmony search to vehicle routing. *American Journal of Applied Sciences, 2*(12), 1552–1557.

Gil-Lopez, S., Del Ser, J., Landa, I., Garcia-Padrones L., Salcedo-Sanz, S. & Portilla-Figueras J. A. (2010). On the application of a novel grouping harmony search algorithm to the switch location problem. In *Second International Conference on Mobile Lighweight Wireless Systems, May* Barcelona (Spain).

Kratica, J., Tosic, D., Filipovic, V., & Ljubic, I. (2000). Genetic algorithm for designing a spread-spectrum radar polyphase code. In *Proceedings of the fifth Online World Conference on Soft Computing Methods in Industrial Applications*, (pp. 191–197).

Landa-Torres, I., Gil-Lopez, S., Salcedo-Sanz, S., Del Ser, J., & Portilla-Figueras, J. A. (2012). A novel grouping harmony search algorithm for the multiple-type access node location problem. *Expert Systems with Applications, 39*(5), 5262–5270.

Lewis, B., Kretschmer, F., Jr., & Shelton, W. (1986). *Aspects of radar signal processing*. Artech House.

Mladenovic, N., Petrovic, J., Kovacevic-Vujcic, V., & Cangalovic, M. (2003). Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search. *European Journal of Operational Research, 151*, 389–399.

Mow, W. H. (1995). *Sequence design for spread spectrum*. Chinese University Press.

Pan, Q. K., Suganthan, P. N., Liang, J. J., & Tasgetiren, M. F. (2011). A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem. *Expert Systems with Applications, 38*(4), 3252–3259.

Pérez-Bellido, A., Salcedo-Sanz, S., Ortiz-García, E.G. & Portilla-Figueras, A. (2007). A Hybrid Evolutionary Programming Algorithm for Spread Spectrum Radar Polyphase Codes Design. In *Proceedings of the nineth Genetic and Evolutionary Computation Conference*, (pp. 682–688).

Pérez-Bellido, A. M., Salcedo-Sanz, S., Ortiz-García, E. G., Portilla-Figueras, J. A., & López-Ferreras, F. (2008). A comparison of memetic algorithms for the spread spectrum radar polyphase codes design problem. *Engineering Applications of Artificial Intelligence, 21*, 1233–1238.

Wang, Chia-Ming, & Huang, Yin-Fu (2010). Self-adaptive harmony search algorithm for optimization. *Expert Systems with Applications, 37*(4), 2826–2837.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82–102.