Teaching Advanced Features of Evolutionary Algorithms Using Japanese Puzzles

Sancho Salcedo-Sanz, *Member, IEEE*, Jose A. Portilla-Figueras, *Member, IEEE*, Emilio G. Ortíz-García, Ángel M. Pérez-Bellido, and Xin Yao, *Fellow, IEEE*

***Abstract*—In this paper, a method to teach advanced features of evolutionary algorithms (EAs), using a famous game known as Japanese puzzles is presented. The authors show that Japanese puzzles are constrained combinatorial optimization problems, that can be solved using EAs with different encodings, and are challenging problems for EAs. Other features, such as special operators and local search heuristics and its hybridization with genetic algorithms, can also be taught using these puzzles. The authors report an experience using this method in a course taught at the Universidad de Alcalá, Madrid, Spain.**

***Index Terms*—Advanced encodings, evolutionary algorithms (EAs), Japanese puzzles, local search, memetic algorithms, teaching method.**

I. Introduction

EVOLUTIONARY ALGORITHMS (EAs) are a class of population-based search and optimization techniques that work on a principle inspired by nature: evolution of species [1]. EAs have proven to be very useful tools in a large number of applications in optimization, control, signal processing, or machine learning [1], [2]. Following this interest in EAs, the majority of computer science schools in the world offer undergraduate and graduate courses about EAs and their applications [3], [4].

Several types of EAs have different encodings and operators. Maybe, the most popular EA is the *genetic algorithm* (GA) [1], [11], [12]. In [1] the description of a *canonical* GA is given. Canonical GAs have been successfully applied to different problems [12]; however, GAs do not perform well in complex combinatorial spaces, for example in highly constrained ones. In these situations, hybridization of GAs with local search techniques may be used to improve their performance [13].

Experience in graduate and undergraduate courses indicate that these advanced features of EAs are better explained in concrete applications [4], where the students have the opportunity of comparing the performance of EAs with (or without) a given feature. The main problem with this methodology is that very few simple problems are available, and challenging for an EA, where advanced features of EAs can be tested. One of these traditional benchmark problems is the *Travelling Salesman Problem* (TSP) which has been used earlier in education [5]. The TSP is a very good benchmark for teaching EAs with permutations encoding, but EAs with other types of encoding (i.e., GAs) always require special operators to ensure the feasibility of the TSP solutions. Sometimes this point makes the use of the TSP to learn EAs difficult. Another possibility to study advanced features of EAs is to use realistic problems. However, this approach requires, in the majority of cases, that one is familiar with the problem, a situation which often complicates the final task of learning about EAs.

Several works describe methods to teach basic EAs in the literature. In [6], the authors offer a very interesting introductory course about GAs. Other works introduce different strategies to teach GAs [7]–[10]. Apart from these works primarily aimed to undergraduate students, on how to teach GAs, a reference does not exist describing methods to teach advanced features of EAs in graduate courses.

In this paper a method to teach some advanced features of EAs is proposed. A very simple and famous game is used: the *Japanese puzzle* or *Nonogram* [14]–[17]. A case of application of the proposed teaching method in a Spanish university is presented and discussed in the paper.

The rest of the article is structured as follows: the next section provides and briefly introduces Japanese puzzles, giving a short note on their history. Section III introduces the different advanced features of EAs to be taught using Japanese puzzles. Section IV explains the proposed teaching method and an experience of its application in a graduate course in Universidad de Alcalá, Madrid, Spain. Finally, Section V closes the paper.

II. Japanese Puzzles

A Japanese puzzle is an interesting and addictive game, which takes the form of a $N \times M$ grid, with numbers situated on the left, top rows, and columns [Fig. 1(a)]. In a Japanese puzzle, the numbers in rows and columns represent how many blocks of cells must be filled in the grid, in the order in which they appear. With two or more numbers, the blocks of cells in the grid must be separated by at least one blank square.

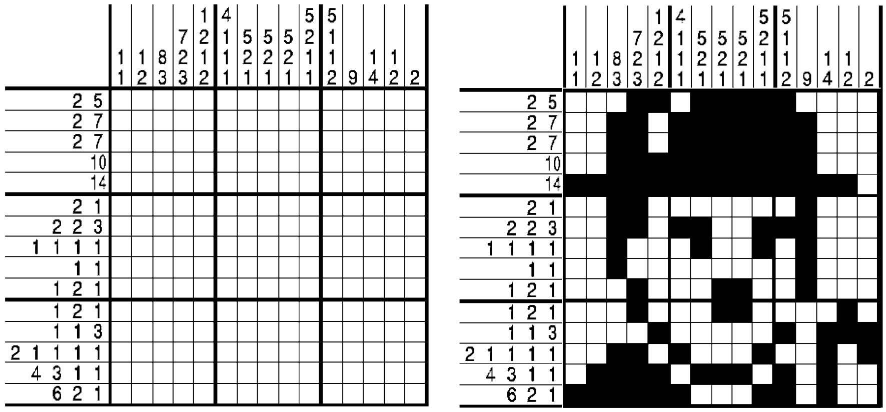
A Japanese puzzle is a constrained combinatorial optimization problem: consider a Japanese puzzle defined in an $N \times M$ grid (N rows, M columns). Since the puzzles are formed by black (filled) and white (blank) squares to be fixed in the grid, the puzzle can be reformulated in terms of finding a $N \times M$ binary matrix, with the number of 1s given by the conditions in rows and columns, separated by at least one 0 when more than one condition is present.

Manuscript received May 5, 2006; revised January 4, 2007.

S. Salcedo-Sanz, J. A. Portilla-Figueras, E. G. Ortíz-García, and Á. M. Pérez-Bellido are with the Department of Signal Theory and Communications, Universidad de Alcalá, Escuela Politécnica Superior, 28871 Alcalá de Henares, Madrid, Spain (e-mail: sancho.salcedo@uah.es).

X. Yao is with the Center for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, The University of Birmingham, B152TT Birmingham, U.K.

Digital Object Identifier 10.1109/TE.2007.893170



(a) (b)

Fig. 1. (a) Japanese puzzle grid corresponding to the puzzle "Chaplin," and (b) its corresponding solution.

In addition, an objective function, which compares the number of 1s and 0s in each line (row or column) of a puzzle solution X with the desired number of 1s and 0s, can be considered. This desired number of 0s and 1s in a line are given by the conditions of the puzzle r_{ij} and c_{ij} (where $r_{\mathrm{ij}}, i \in N, \forall j$, and $c_{\mathrm{kj}}, k \in M, \forall j$, stand for the puzzle conditions in rows and columns, respectively). For example, in Fig. 1(a), $r_{11} = 2, r_{12} = 5, r_{13} = 0$, etc., and $c_{11} = 1, c_{12} = 1, c_{42} = 2$, etc. With these simple definitions, the following objective function can be obtained:

$$f(X) = f_1(X) + f_2(X) + f_3(X) + f_4(X) \tag{1}$$

$$f_1(X) = \sum_{i=1}^{N} \left| \sum_{p=1}^{M} r_{ip} - \sum_{p=1}^{M} x_{ip} \right| \tag{2}$$

$$f_2(X) = \sum_{k=1}^{M} \left| \sum_{p=1}^{N} c_{\mathrm{kp}} - \sum_{p=1}^{N} x_{\mathrm{kp}} \right| \tag{3}$$

$$f_3(X) = \sum_{i=1}^{N} \left| \left(M - \sum_{p=1}^{M} r_{ip} \right) - \left(M - \sum_{p=1}^{M} x_{ip} \right) \right| \tag{4}$$

$$f_4(X) = \sum_{k=1}^{M} \left| \left(N - \sum_{p=1}^{N} c_{\mathrm{kp}} \right) - \left(N - \sum_{p=1}^{N} x_{\mathrm{kp}} \right) \right| \tag{5}$$

where $f_1(X)$, given by (2) takes into account the difference between the desired number of 1s and the actual number of 1s in rows, for a given solution X, and $f_2(X)$, given by (3), does the same for columns. On the other hand, the term given in (4) stands for the differences in number of 0 s for rows, and in (5) for columns.

The Japanese puzzle can be redefined then, giving a puzzle grid $N \times M$ and a set of puzzle conditions (conditions matrices R with elements r_{ij} and C with elements c_{kj}), and finding an $N \times M$ binary matrix X, which minimizes function $f(X)$. This problem is a constrained combinatorial optimization problem.

III. Using Japanese Puzzles to Teach Evolutionary Algorithms

Following the definition of the problem in Section II, the application of a canonical GA to solve Japanese puzzles is straight forward. A first possible encoding of the puzzle can be accomplished by using a binary string of length $N \cdot M$, where the first N bits stand for the first row of the solution matrix X, the next N bits stand for the second row, etc. A small change in the objective function is also needed to use the standard selection mechanism (roulette wheel [1]). Just by defining a function $g(X) = N \cdot M - f(X)$, a minimization problem is translated into a maximization one: finding a matrix X which maximizes $g(X)$. With this encoding, the search space in which the EA has to operate is $\{0, 1\}^{N \cdot M}$.

In a first approach, a canonical GA can be run to solve the Japanese puzzle example in Fig. 1(a). The maximum number of function evaluations performed by the GA were fixed to 50 000 (100 individuals, 500 generations in the case of the canonical GA [1]), with a probability of crossover. $P_c = 0.6$ and probability of mutation $P_m = 0.01$. Fig. 2(a) shows the evolution of the canonical GA, i.e., best individual in different generations. As can be seen in Fig. 2(a), the canonical GA does not converge to the optimal solution of the problem within the number of function evaluations prefixed. The main problem with the canonical GA is the search space size. Because of the encoding used the algorithm has to perform search in rows and columns, which results in a huge search space size of $\{0, 1\}^{N \cdot M}$.

A. Advanced Encoding and Special Operators

For the case in which EA individuals (possible solutions to the puzzle) fulfill the conditions of the puzzle in columns, i.e., all the EA individuals are feasible in columns, but not necessarily in rows. In this special case, the search space would reduce to $\{0, 1\}^N$. Then, a new operator is needed since every individual in the EA population must fulfill the puzzle conditions in columns.

For a given column j, which is defined by a set of k numbers $n_l, l = 1, \ldots, k$, belonging to a Japanese puzzle with N rows and M columns. See Fig. 1(a) as an example: column 1 is defined by two numbers ($k = 2$), $c_{11} = 1, c_{12} = 1$ (the rest $c_{13}, \ldots, c_{1M}$ are 0). Column 2 is defined by 2 numbers ($k = 2$), $c_{21} = 1$ and $c_{22} = 2$, etc. For each number in a column, a first possible square and a last possible square is available to start

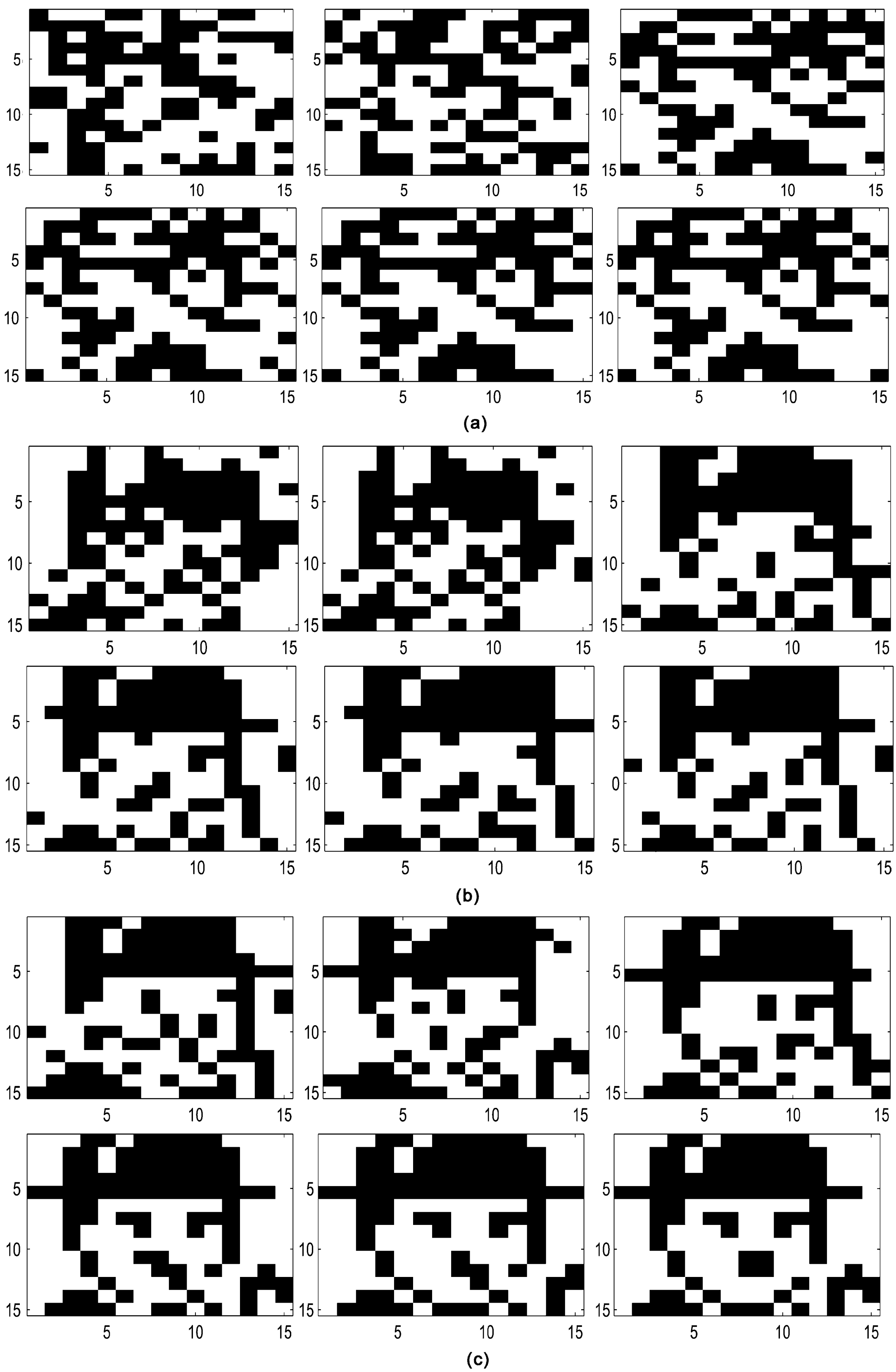


Fig. 2. Evolution of the best solution found by a (a) canonical GA; (b) EA with advanced encoding; and (c) EA with advanced encoding and local search in the puzzle "Chaplin." The initial solution found by the algorithm corresponds to the top left-hand corner figure, and the final solution found by the algorithm is depicted at the bottom right-hand corner figure in all cases.

filling. In the example, in column 1, the first set of filled squares can start in square 1 or, at most, in square

$$M - (\sum_i n_i + (k-1)) + 1 \tag{6}$$

which in this case takes the value $15 - 3 + 1 = 13$. Therefore, the filled squares defined by n_1 can start between square 1 and square 13. The starting point of the rest of the filled squares depends on the position assigned to the first set of filled squares. Based on these observations, an heuristic for initializing the population can be easily constructed in such a way that the columns of matrix X fulfil the puzzle constraints. The heuristic works choosing the initial-filled square at random between the first and last possible squares, (6). The rest of the squares of the column are filled taking into account the previous squares already filled. This heuristic produces a solution which fulfils the puzzle constraints in columns, but, in general, the solution does not fulfil the constraints in rows. This operator, which can be used to start the EA population, is known as *Inic_columns* heuristic. A new encoding can be introduced then: EA individuals are not binary strings any more, but binary matrices X. These matrices fulfil the puzzle conditions in columns after the application of the *Inic_columns* operator. This encoding has the drawback that the standard crossover and mutation operators would produce unfeasible solutions in columns. This problem can be avoided by introducing "alleles" in the encoding, following the structure of the problem. Consider again Fig. 1(a) as an example. Each column can be seen as an allele with several genes: allele 1 has 2 genes (two conditions in the puzzle); allele 2 has 2 genes; allele 4 has 3 genes, etc. If crossover and mutation operators are forced to operate between alleles and not between genes, they will keep the feasibility of the solution after their application.

The modified crossover operator used works in the following way. Given the population of the EA, the first step is to form couples at random with the individuals. Each couple has a probability of producing an offspring of P_c. Each couple produces then two new individuals by swapping alleles between them, i.e., swapping columns between individuals. This crossover does not swap genes, thus the feasibility in columns is ensured.

The mutation operator works on each individual, with a probability P_m. Once an individual must be mutated, this mutation is forced to change alleles, not genes. A number of alleles (columns) are selected at random, and the *Inic_columns* heuristic described above is applied to obtain a mutated (feasible) individual.

The fitness function must be also modified to cope with the new encoding of the EA as follows:

$$f(X) = f_1(X) + f_3(X) \tag{7}$$

where f_1 and f_3 are given by (2) and (4), respectively. In this case the part of the fitness function which works with errors in columns, (3) and (5), can be removed since there will be no errors in columns, only in rows. Now the search space size has been reduced to $\{0,1\}^N$.

The performance of the EA with advanced encoding and special operators in Japanese puzzles is shown in the following Chaplin puzzle. The number of function evaluations is maintained to be 50 000 as before. Fig. 2(b) shows the evolution of the EA with advanced encoding in the puzzles considered. All the solutions shown fulfil the puzzle's constraints in columns. The result obtained by the new EA is better than the result of the canonical GA in the puzzle; however, the optimal solution to the puzzle is not obtained this time.

B. Local Search: Memetic Algorithms

Hybridization of EAs with local search procedures is a common practice to improve the EAs performance [13]. This Section shows the improvement that can be obtained with a basic local search (hill-climbing), hybridized with the EA, for solving Japanese puzzles. The hill-climbing algorithm is one of the most basic local search procedures possible. Each individual in the EA is mutated, and if this mutation produces a better individual, this individual is selected to replace the current individual in the EA [13]. Mutated solutions with worse fitness than the current individual are discarded. In this case, the *Inic_columns* heuristic defined above is used to perform the mutation in the proposed EA for Japanese puzzles.

A hill-climbing procedure has been applied to the EA with advanced encoding and special operators. To carry on a fair comparison with the case of the EA without a local search, the number of function evaluations has been maintained to be 50 000 (as in previous experiments), including the function evaluations performed by the local search procedure. Fig. 2(c) shows the evolution of the memetic algorithm in puzzle "Chaplin." In this case the algorithm is able to solve completely the puzzle.

C. Educational Aspects of the Experiments With EAs and Japanese Puzzles

The study proposed in this paper has focused on three important concepts regarding advanced features of EAs[1]: encoding schemes; the relation of the encoding used with the search space size; constraint satisfaction; how to maintain the feasibility of solutions after applying the evolutionary operators; and, finally, the hybridization of EAs with local search procedures to fine-tune their search abilities.

The first and second experiments performed are related to encoding and constraint satisfaction issues in EAs. The experiments showed that the canonical GA is not able to solve highly constrained problems (such as Japanese puzzles) primarily because of the huge search space size ($\{0,1\}^{N \cdot M}$). However, the inclusion of an advanced encoding produces a reduction of the search space size ($\{0,1\}^N$), and the EA performs better. Also, a new operator must be applied to fulfil the conditions in columns (Section III-A). In addition, new crossover and mutation operators must be developed to keep the EA individuals feasible after the application of evolutive operators. Finally, experiments on the hybridization of the EA with a local search heuristic are provided to show that local search algorithms can enhance the performance of EAs.

[1]There are, of course, other important advanced features of EAs which have not been considered in this paper, and that can also be the subject of graduate and undergraduate courses on EAs.

IV. Teaching Method and Assessment

A course for teaching advanced features of EAs based on Japanese puzzles can be structured using the material presented in this paper. The first step is to make the students realize that they are working with a combinatorial optimization problem (Section II). The following steps consist of guiding the students through the process shown in the previous sections by proposing assignments. The first assignment is the application of a canonical GA to solve a Japanese puzzle. The second assignment requires a little effort to implement the advanced encoding and operators, explained in Section III. In the last assignment, the students must implement a local search to construct a memetic algorithm. The implementation of a simple local search as the hill-climbing proposed in Section III-B is suggested, but other possibilities such as a tabu search [18] or an heuristic local search can also be explored.

The presented method has been tested in the graduate course *Heuristics Methods for Optimization Problems in Engineering*, taught at the Universidad de Alcalá. This course is part of the Doctoral Program, "Computer Architecture and Signal Processing Techniques in Telecommunications," jointly taught by the *Signal Theory and Communications Department* and *Computers Architecture Department* at the Universidad de Alcalá. In this course, 10 postgraduate students learned about advanced features of EAs, among many other metaheuristics algorithms. In order to assess the proposed method, the students were asked to fill a small questionnaire about the method in which questions asked about the novelty of the method and their feelings about it. All the students agreed that they better understood concepts, such as advanced encoding, special operators, or memetic algorithms, with the proposed method than with studying them over difficult real applications or problems. The students also emphasized the point that the Japanese puzzles were really easy to understand, but very hard to solve, so that the application of an EA is an interesting possibility.

V. Conclusion

In this paper, Japanese puzzles have been proposed as appealing benchmark problems to teach advanced features of EAs. These puzzles are, in fact, constrained combinatorial optimization problems which can be solved using EAs. The interest of Japanese puzzles in education is that they are challenging problems to EAs and must incorporate new operators and local search procedures. In addition, these puzzles are really appealing to students, who can solve the puzzles by hand and compare the solution to the one obtained by the EA.

References

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

[2] Z. Michalewicz, *Genetic Algorithms + data structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1996.

[3] X. Yao, "A research-led and industry-oriented MSc program in Natural Computation," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 39–40, Feb. 2006.

[4] X. Yao, "How does evolutionary computation fit into IT postgraduate teaching," in *Proc. Congress Evolutionary Computation*, Washington, D.C., 1999, vol. 3, pp. 1707–1713.

[5] G. Pataki, "Teaching integer programming formulations using the traveling salesman problem," *SIAM Rev.*, vol. 45, no. 1, pp. 116–123, 2003.

[6] Y. H. Liao and C. T. Sun, "An educational genetic algorithms learning tool," *IEEE Trans. Educ.*, vol. 44, no. 2, pp. 20–30, May 2001.

[7] E. Bütün, "Teaching genetic algorithms in electrical engineering education: A problem-based learning approach," *Int. J. Electr. Eng. Educ.*, vol. 42, no. 3, pp. 223–233, 2005.

[8] Y. J. Cao and Q. H. Wu, "Teaching genetic algorithm using MATLAB," *Int. J. Electr. Eng. Educ.*, vol. 36, no. 2, pp. 139–153, 1999.

[9] D. Jackson and A. Fovorque, "The use of animation to explain genetic algorithms," in *Proc. Technical Symp. Computer Science Education*, San Jose, CA, 1997, pp. 243–247.

[10] C. Prince, R. Wainwright, D. A. Schoenefeld, and T. Tull, "GATutor: A graphical tutorial system for genetic algorithms," in *Proc. 25th Symp. Computer Science Education*, Phoenix, AZ, 1994, pp. 203–207.

[11] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1997.

[12] D. Beasley, D. R. Bull, and R. Martin, "An overview of genetic algorithms: Part 1 fundamentals," *University Computing*, vol. 15, no. 2, pp. 58–69, 1993.

[13] 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, Oct. 2005.

[14] G. Duncan, Puzzle Solving, Computer Science Dept., Univ. York, York, U.K., B.Sc. Degree Final Project Rep., 1999.

[15] B. J. Batenburg and W. A. Kosters, "A discrete tomography approach to Japanese puzzles," in *Proc. Belgian-Dutch Conf. Artificial Intelligence*, Groningen, The Netherlands, 2004, pp. 243–250.

[16] J. Benton, R. Snow, and N. Wallach, A Combinatorial Problem Associated With Nonograms, Artificial Intelligence Dept., Stanford Univ., Stanford, CA, Internal Rep., 2005.

[17] N. Ueda and T. Nagao, NP-Completeness Results for Nonograms Via Parsimonious Reductions, Computer Science Dept., Univ. Tokyo, Tokyo, Japan, Internal Rep., 1996.

[18] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

Sancho Salcedo-Sanz (S'00–M'03) was born in Madrid, Spain, in 1974. He received the B.S. degree in physics from the Universidad Complutense, Madrid, in 1998, and the Ph.D. degree in telecommunications engineering from the Universidad Carlos III, Madrid, in 2002.

He spent one year in the School of Computer Science, The University of Birmingham, Birmingham, U.K., as a Postdoctoral Research Fellow. Currently, he is an Associate Professor in the Department of Signal Theory and Communications at the Universidad de Alcalá, Madrid. He has coauthored more than 20 international journal and conference papers in the field of genetic algorithms and hybrid algorithms. His current research interests include optimization problems in telecommunications, genetic algorithms, hybrid algorithms, and neural networks.

Jose A. Portilla-Figueras (M'05) was born in Santander, Spain, in 1976. He received the B.S. and Ph.D. degrees in telecommunications engineering from Universidad de Cantabria, Cantabria, Spain, in 1999 and 2004, respectively.

He is currently an Associate Professor in the Department of Signal Theory and Communications at Universidad de Alcalá, Madrid, Spain. His main research interests are metaheuristics algorithms and their application to telecommunication problems.

Emilio G. Ortíz-García was born in Madrid, Spain, in 1984. He received the B.S. degree in telecommunications engineering from Universidad de Alcalá, Madrid, in 2006.

He is currently a Research Fellow in the Department of Signal Theory and Communications at Universidad de Alcalá, where he is working towards the Ph.D. degree. His research interests are genetic and evolutionary algorithms, and neural networks for optimization problems.

Ángel M. Pérez-Bellido was born in Madrid, Spain, in 1984. He received the B.S. degree in telecommunications engineering from Universidad de Alcalá, Madrid, in 2006.

He is currently working towards the Ph.D. degree as a Research Fellow in the Department of Signal Theory and Communications at Universidad de Alcalá. His current research topic concerns memetic algorithms and their applications in electrical engineering problems.

Xin Yao (M'91–SM'96–F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC, Hefei, in 1990.

He was an Associate Lecturer and Lecturer from 1985 to 1990 with USTC while working on the Ph.D. degree. His Ph.D. work on simulated annealing and evolutionary algorithms was awarded the President's Award for Outstanding Thesis by the Chinese Academy of Sciences. In 1990, he took up a postdoctoral fellowship at the Computer Sciences Laboratory, Australian National University, Canberra, A.C.T., Australia, and continued his work on simulated annealing and evolutionary algorithms. In 1991, he joined the Knowledge-Based Systems Group, Commonwealth Scientific and Industrial Research Organization (CSIRO) Division of Building, Construction, and Engineering, Melbourne, Vic., Australia, working primarily on an industrial project on automatic inspection of sewage pipes. In 1992, he returned to Canberra to take up a lectureship at the School of Computer Science, University College, University of New South Wales, Australian Defence Force Academy, Kensington, N.S.W., Australia, where he was later promoted to Senior Lecturer and Associate Professor. Attracted by the English weather, he moved to the University of Birmingham, Birmingham, U.K., as a Professor (Chair) of computer science on April 1, 1999. He is currently the Director of the Centre of Excellence for Research in Computational Intelligence and Applications, Birmingham, a Distinguished Visiting Professor of the University of Science and Technology of China, Hefei, and a Visiting Professor of three other universities. He has been invited to give more than 45 invited keynote and plenary speeches at conferences and workshops worldwide. He has more than 200 refereed research publications. His research has been supported by research councils, government organizations, and industry. His major research interests include evolutionary computation, neural network ensembles, and their applications.

Prof. Yao serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, an associate editor or editorial board member, of several other journals, and the Editor of the World Scientific book series on "Advances in Natural Computation." He is also a Distinguished Lecturer of the IEEE Computational Intelligence Society. He is the recipient of the 2001 IEEE Donald G. Fink Prize Paper Award for his work on evolutionary artificial neural networks.