

Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test

Francisco J. Rodriguez, Carlos García-Martínez, and Manuel Lozano

Abstract—The design of hybrid metaheuristics with ideas taken from the simulated annealing and evolutionary algorithms fields is a fruitful research line. In this paper, we first present an overview of the hybrid metaheuristics based on simulated annealing and evolutionary algorithms presented in the literature and classify them according to two well-known taxonomies of hybrid methods. Second, we perform an empirical study comparing the behavior of a representative set of the hybrid approaches based on evolutionary algorithms and simulated annealing found in the literature. In addition, a study of the synergy relationships provided by these hybrid approaches is presented. Finally, we analyze the behavior of the best performing hybrid metaheuristic with regard to several state-of-the-art evolutionary algorithms for binary combinatorial problems. The experimental studies presented provide useful conclusions about the schemes for combining ideas from simulated annealing and evolutionary algorithms that may improve the performance of these kinds of approaches and suggest that these hybrids metaheuristics represent a competitive alternative for binary combinatorial problems.

Index Terms—Combinatorial optimization, evolutionary algorithms (EAs), hybrid metaheuristics (HMs), simulated annealing (SA).

I. INTRODUCTION

OVER THE LAST few years, a large number of search algorithms have been presented that do not simply follow the concepts of one single classical metaheuristic [1], [2], but attempt to obtain the best from a set of metaheuristics (and even other kinds of optimization methods) that perform together and complement each other to produce a profitable synergy from their combination. These approaches are commonly referred to as hybrid metaheuristics (HMs) [3]–[5].

Simulated annealing (SA) [6]–[8] is commonly said to be the first algorithm extending local search methods with an explicit strategy to escape from local optima. The fundamental idea is to allow moves resulting in solutions of worse quality

Manuscript received March 17, 2011; revised June 27, 2011 and October 5, 2011; accepted November 15, 2011. Date of publication February 7, 2012; date of current version November 27, 2012. This work was supported by Research Projects TIN2008-05854 and P08-TIC-4173.

F. J. Rodriguez and M. Lozano are with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain (e-mail: fjrodriguez@decsai.ugr.es; lozano@decsai.ugr.es).

C. García-Martínez is with the Department of Computing and Numerical Analysis, University of Córdoba, Córdoba 14071, Spain (e-mail: cgarcia@uco.es).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2012.2182773

than the current solution in order to escape from local optima. The probability of making such a move is decreased during the search process. Although it was proposed in 1983, SA is still the object of further studies [9], applied to many optimization problems, or used as a component of other search algorithms [8], [10]. It is precisely because of its outstanding role in the metaheuristic field that further studies to obtain more effective SA models are encouraged.

Evolutionary algorithms (EAs) [11], [12] are stochastic search methods that mimic the metaphor of natural biological evolution. EAs rely on the concept of a *population* of individuals (representing search points in the space of potential solutions to a given problem), which undergo probabilistic operators such as mutation, selection, and (sometimes) recombination to evolve toward increasingly better fitness values of the individuals. There have been a variety of slightly different EAs that, basically, fall into three different categories that have been developed independently from each other. These are evolutionary programming [13], evolution strategies [14], and genetic algorithms (GAs) [15]. EAs offer practical advantages to researchers facing difficult optimization problems because they may locate high performance regions of vast and complex search spaces. Other advantages include the simplicity of the approach, their flexibility, and their robust response to changing circumstances.

The hybridization of EAs is becoming popular due to its ability to handle several real-world problems involving complexity, noise, imprecision, uncertainty, and vagueness [16]–[19]. A wide variety of metaheuristics such as tabu search [20], greedy randomized adaptive search procedure [21], and iterated local search [22], among others, have been employed to develop hybrid approaches with EAs. In this paper, we focus on the use of SA to design HMs with EAs (HMs-EA/SA) due to its prominent role in the field of the hybrid EAs [23]–[27].

The current relevance of HMs-EA/SA can be shown through the visibility of this topic at the ISI Web of Science. Fig. 1 shows an important number of publications and citations per year, as well as an increasing trend. We can conclude that, although the first items related to this topic appeared in 1992, nowadays HMs-EA/SA are subject of great interest and there is an important research community associated to their study. Moreover, in Fig. 2 we can observe that the number of works at the ISI Web of Science considering HMs-EA/SA is greater than those of hybrid metaheuristics combining EAs and other

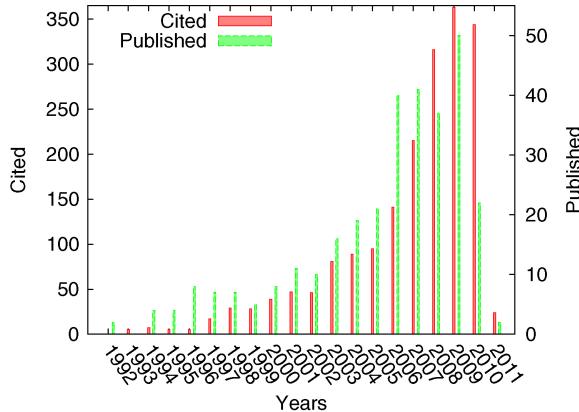


Fig. 1. Number of publications and citations per year for HMs-EA/SA (Web of Science).

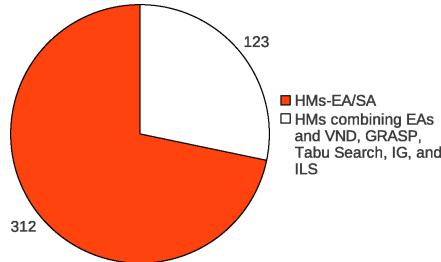


Fig. 2. Comparison between publications considering HMs-EA/SA and hybrid algorithms combining EAs and other metaheuristics (Web of Science).

metaheuristics (greedy randomized adaptive search procedure, iterated local search, variable neighborhood descent, iterated greedy, and tabu search). In addition, we can highlight that many different instantiations of HMs-EA/SA are presented to solve real-world problems covering domains that range from the design of combinational circuits [28], routing problems in telecommunications [29], cluster analysis [30], optimization of support vector machine parameters [31], scheduling problems [32], protein structure prediction [33], and image processing [34], to name but a few.

The goal in this paper is threefold. First, we attempt to paint a more complete picture of HMs-EA/SA than ever before. To do so, we structure and organize the knowledge about the HM-EA/SA approaches found in the literature by proposing a taxonomy for HMs-EA/SA based on those conceived by Talbi [3] and Raidl [4] for HMs. Our second objective is to study the empirical behavior of the different kinds of approaches according to the proposed taxonomy, analyzing what schemes provide a better performance. Finally, we want to study the synergistic relationships created by the hybridization of EAs and SA in these approaches. Suitably combining the complementary algorithm concepts can provide hybrid approaches with a better performance than that obtained by EAs or SA separately.

The remainder of this paper is organized as follows. In Section II, we present an overview of the HM-EA/SA approaches found in the literature and propose a taxonomy that characterizes them. In Section III, we describe the experimental framework employed in this paper. In Section IV, we compare the performance of a set of representative

TABLE I
TAXONOMY FOR HMs-EA/SA

General Categories		HM-EA/SA Categories	Instances
Collaborative	Teamwork	Multiple EAs and SAs	DCHCSA
		Multiple SAs	SSSA [23], CSA [35], ESA [36], GAMSA [37]
	Relay	EA then SA	HHSAGA [38], SAGA [39]
		SA then EA	GA-PSA
	TeamWork	MA with SA as local search	AGA [40], GASAHSA [31], IGA-SA [41], GSAAAL [42], GSAA [24]
		SA-based EA selection	HGA-BTS [43], GESA [44], HGA-BS[45]
		SA-based EA mutation and crossover	SAGACIA [46], ARSAGA [47], GSAAIA [48], HGA-SAM/R [49]
		SA-based EA replacement	PRSA [50], PGSA [25], GSA [51], NPOSA [52], MPGSAA [26], GSA-MLE [53]
Relay	EA-based SA component	SALGeS [54]	GAMSA [37]

HMs-EA/SA instances belonging to different categories. In Section V, we analyze the performance of the groups of HMs-EA/SA studied. In Section VI, we perform a synergy study of the HMs-EA/SA compared. In Section VII, we pit the best performing HM-EA/SA approach against some state-of-the-art EAs for binary combinatorial problems. Finally, in Section VIII, we present conclusions and future work.

II. HMs BASED ON EAS AND SA: OVERVIEW AND TAXONOMY

We have grouped different instances of HMs-EA/SA appearing in the literature into two broad categories (Table I summarizes the HMs-EA/SA found and the category they belong to). These two groups are specified following two well-known existing taxonomies for HMs [3], [4], which are based on the architecture of the algorithms.

- 1) *Collaborative HMs*. These are based on the exchange of information between different self-contained metaheuristics (and possibly other optimization techniques) running sequentially or in parallel.
- 2) *Integrative HMs*. In this case, one algorithm is considered a subordinate, embedded component of a master metaheuristic, which governs the search process.

A. Collaborative HMs-EA/SA

Collaborative HMs apply different self-contained metaheuristics. These metaheuristics can be regarded as black

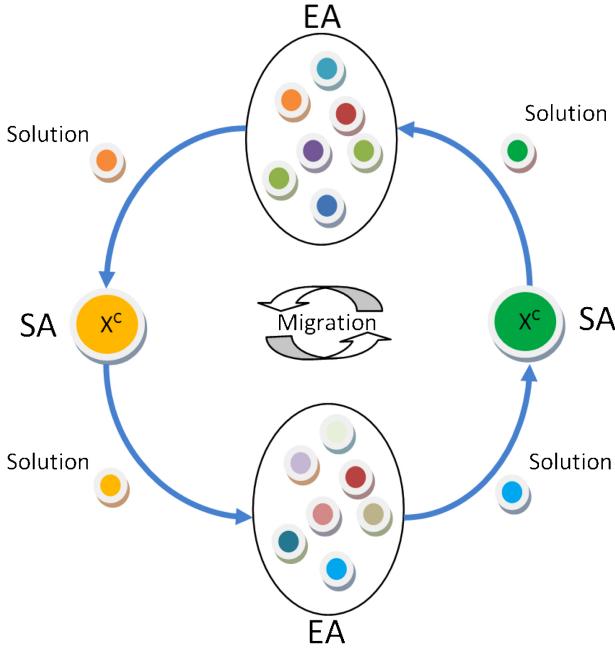


Fig. 3. Multiple EAs and SAs running in parallel.

boxes and the cooperation takes place through the exchange of some kind of information. According to the way metaheuristics are executed, collaborative HMs-EA/SA can be subdivided into teamwork or relay categories [3].

1) *Teamwork Collaborative HMs-EA/SA*: In *teamwork collaborative HMs*, there are several metaheuristics that work in parallel and exchange solutions, parameters, and so on from time to time. To the best of our knowledge, there is no proposed method arranging *multiple EAs and SAs* running in parallel. However, we may easily devise teamwork collaborative HM-EA/SA models based on distributed GAs [55]. In this scheme, a single population is decentralized by partitioning it into several subpopulations (islands or demes), where island GAs are run performing sparse exchanges (migrations) of individuals. It is easy to conceive a model that replaces some GAs with SAs. In this case, we can consider some subpopulations that are optimized by EAs and other subpopulations formed by only one individual and optimized by an SA process (Fig. 3). An instance of this kind of approach called *DCHCSA* will be presented in Section IV.

There are several proposals in the literature worthy of mention in this section, which consider the execution of multiple SAs that cooperate to explore the search space by exchanging solutions, parameters, and so on (Fig. 4). They do not consider the application of self-contained EAs, but evolutionary concepts underlie the general scheme. These algorithms consider a population of agents that evolve by means of neighborhood and stochastic selection operators. For this reason, authors may argue that they resemble EAs. Though these models are hardly seen as HM-EA/SA approaches, they are considered in this paper for the sake of coverage. They are referred to as *multiple SAs*.

The evolutionary SA (ESA) algorithm [36] considers a population of solutions. It selects one individual according to a running selection rule, operates on it with a neighborhood

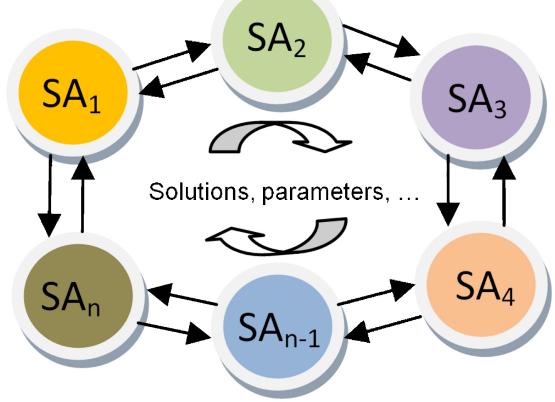


Fig. 4. Multiple cooperating SA processes.

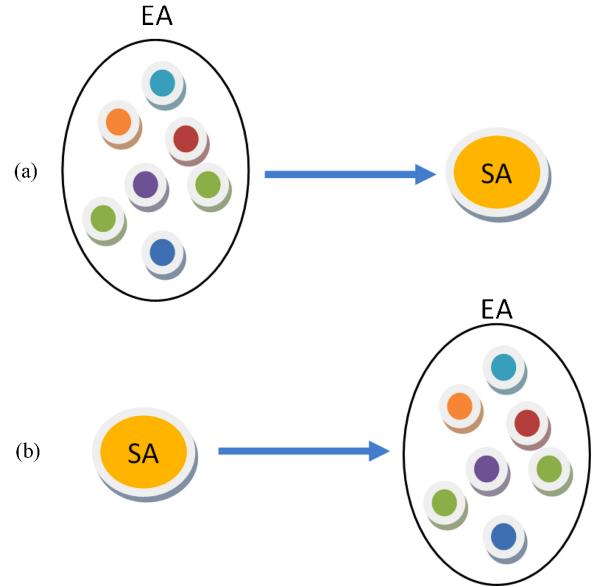


Fig. 5. EAs and SAs in a pipeline fashion. (a) EA then SA. (b) SA then EA.

operator, and evaluates whether to put it back into the population according to a particular replacement rule. The new individual provided by the neighborhood function is adopted according to the Metropolis acceptance criterion. The ESA algorithm can be seen as a population of multiple SA processes that exchange their current solutions at every iteration.

Sample-sort SA (SSSA) [23] maintains an array of samplers operating at static temperatures. At each iteration, each one first considers whether to accept any of the states of its neighboring samplers. Then, each sampler performs a standard iteration of SA. Coupled SA (CSA) [35] considers a population of samplers where the acceptance probability of any sampler making an uphill move depends on the states of the other samplers.

2) *Relay Collaborative HMs-EA/SA*: In relay collaborative HMs, several metaheuristics are executed in a pipeline fashion. The output of each algorithm is supplied as the input to the next one (Fig. 5). Depending on the order of execution, two kinds of basic approaches can be found:

- a) *EA then SA*: In [38], the highly hybrid GA+SA (HHSAGA) generates a number of random initial solutions

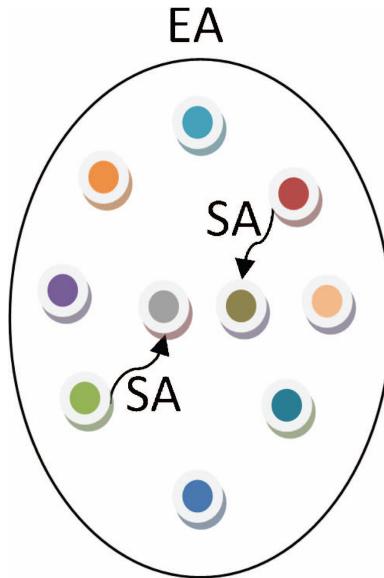


Fig. 6. MA with SA as local search procedure.

and runs a GA through a fixed number of iterations. After the GA ends, each individual in the final population is optimized by a SA [Fig. 5(a)]. At the end of the execution of all the SA processes, a new population is generated using the solutions obtained by the SA processes. Then, the GA starts again. This cycle is repeated until the termination condition is reached. This scheme also appears in the parallel heuristic SA+GA (SAGA) presented in [39].

b) *SA then EA*: By way of contrast, it is possible to devise an algorithm that starts from SA and uses EAs to enrich the solutions found [Fig. 5(b)]. The scheme that most directly fits this pattern is an EA that uses SA as a method to initialize the population. An implementation of this scheme called GA with population initialized by SA (GA-PSA) will be presented in Section IV.

B. Integrative HMs-EA/SA

Integrative HMs address the functional composition of a single optimization method. In this kind of HMs, a given function of a metaheuristic is replaced by another metaheuristic [3]. Integrative HMs can be also subdivided into teamwork or relay categories.

1) *Teamwork Integrative HMs-EA/SA*: In *teamwork integrative HMs*, one metaheuristic (subordinate) becomes a component of another population-based metaheuristic (master) [3]. In the case of HMs-EA/SA, we can find the following approaches.

a) *Memetic algorithms with SA as local search procedure*: Memetic algorithms (MAs) [56] combine a EA in charge of the global exploration with a local search procedure, which is executed within the EA run, looking for a synergy that takes benefits from both.

There are several instances of MAs in which the local search is performed by means of SA (Fig. 6). Annealing-GA (AGA) [40] is an EA where the best solutions are optimized by SA processes, which perform cooling at each iteration and apply the Metropolis acceptance mechanism.

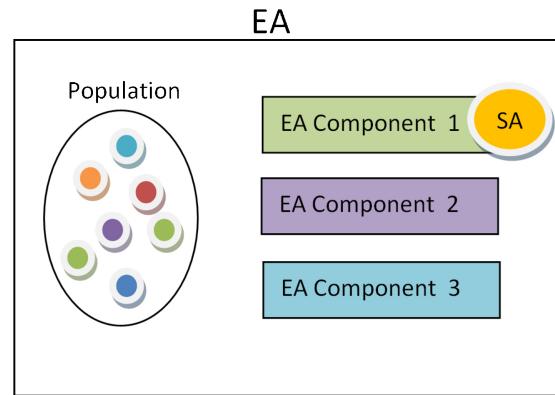


Fig. 7. EA with SA-based components.

Moreover, we can find several recent proposals that apply this scheme to solve real-world problems. In [31], a GA-SA hybrid algorithm (GASAHA) is proposed to optimize the parameters of a support vector machine. In each iteration, the GA operates on the population using three basic genetic operators to produce a new population. Afterwards, the GA applies the SA to the best individual of the GA population to further improve it. In [24], the genetic simulated annealing algorithm (GSAA) deals with a similar problem. In this case, an SA process improves all the solutions. More examples of MAs based on SA can be found in [41] [improved GA-SA (IGA-SA)] and [42] [genetic SA algorithm based localization (GSAAL)].

b) *SA-based EA component*: Another teamwork integrative HM-EA/SA approach consists of defining EA components by using principles of the SA algorithm (Fig. 7). Specifically, the EA components that have been replaced or extended in the literature by the SA are the selection mechanism, the crossover and mutation operators, and the replacement strategy.

SA-based EA selection

In [43], a hybrid GA uses a Boltzmann tournament selection (HGA-BTS) function to provide asymptotic convergence. This selection function uses pairwise probabilistic acceptance and anti-acceptance mechanisms on three individuals from the population. The anti-acceptance competition takes place between two of the three selected individuals. The acceptance competition takes place between the winner of this competition and the other individual.

In [45], the number of offspring that an individual can contribute to the next generation is calculated by implementing Boltzmann scaling on the fitness function [hybrid GA with Boltzmann scaling (HGA-BS)] by varying the selective pressure as a function of the temperature.

In the guided evolutionary SA (GES) algorithm [44], there are two levels of competition. In the first one, the children of the same family (i.e., generated from the same parents) compete with each other and only the one with the best fitness value survives. At the second level of competition, the best child is compared with its parents to find the members for the next generation. A Boltzmann probability is applied in order to decide whether the child will be accepted.

SA-based EA mutation and crossover

The approach proposed in [49] combines SA with GA by extending the mutation and crossover operators with SA (*hybrid GA with SAM/SAR, HGA-SAM/R*). Mutated and recombined solutions are accepted according to the standard SA acceptance condition.

In [48], the genetic SA algorithm-based inverse algorithm (GSAAIA) uses the SA technique to control the mutation operator in the EA. The member to be mutated is perturbed according to the Gaussian probabilistic distribution function and its variance is controlled by the SA technique.

Adaptive real-parameter SA-GA (ARSAGA) [47] with SA-based mutation operator samples a new solution from the neighborhood of the solution generated by the crossover operator. The new solution is accepted according to the SA acceptance criterion. The same procedure is applied in the SAGA-CIA algorithm (SA, GA, *chemotaxis algorithm, integrated, algorithm*) [46] to accept new solutions during the phase called rough search, which can be seen as a mutation operator.

SA-based EA replacement

Parallel recombinative SA (PRSA) [50] iterates over a population of solutions, employing a crossover operator and an unary neighborhood operator (mutation). Offspring replace their parents according to the acceptance mechanism of SA.

In [52], an algorithm called [new population-oriented SA (NPOSA)] employs, in addition, a replacement strategy based on the SA acceptance criterion. However, each individual has its own local temperature according to the individual's rank, settled by its cost value. If an individual finds that its own cost is higher than those of the other individuals, then it raises its local temperature to heighten the probability of uphill climbing in the cost space. Otherwise, it drops its temperature to lower this probability.

The genetic SA (GSA) is a parallel algorithm introduced in [51] that maintains small independent populations of solutions and performs periodic exchanges between the populations. Each subpopulation generates a direction and distance in the subpopulations grid to obtain a mating from another subpopulation. After that, the crossover and mutation operators are performed. A solution that replaces the resident candidate is selected from the current resident candidate and the offspring. An SA acceptance criterion is used to select this solution. Another parallel hybrid genetic SA approach that uses the SA acceptance mechanism to replace the solutions of the population is presented in [25] [parallel genetic SA (PGSA)].

More examples that follow this scheme to get HMs-EA/SA can be found in [26] [multipopulation parallel genetic SA algorithm (MPGSAA)] and [53] [genetic SA algorithm (GSA-MLE)].

2) *Relay Integrative HMs-EA/SA*: This kind of HMs represents algorithms in which a given metaheuristic is embedded into a trajectory-based metaheuristic [3]. Specifically, for the HMs-EA/SA, an EA is used to perform one or more functions in an SA process (Fig. 8). The SA based on local genetic search (SALGeS) [54] presents an EA designed specifically to play the role of the SA neighborhood operator. In particular, a steady-state EA creates one single candidate solution at

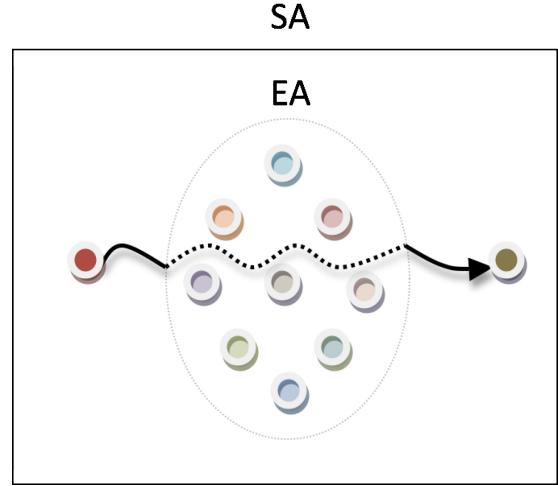


Fig. 8. SA with EA-based functions.

each iteration, by crossing over the current solution of the master SA and another one from the population. Afterwards, the master SA applies an acceptance mechanism to decide which solution becomes the new current solution, either the candidate solution or the current one. The other solution is inserted into the population by a replacement strategy.

The GA-based multiple SA (GAMSA) [37] extends the idea of SALGeS by considering the execution of multiple SA processes that share a unique steady-state EA. The existence of several SA processes promotes diversification by exploring different regions of the search space. On the other hand, the population of the EA allows the SA agents to communicate with one another in order to explore the search space. GAMSA can be seen as a population of SA processes that cooperate to explore the search space. In this sense, GAMSA can be classified as *teamwork collaborative*. On the other hand, each SA process in GAMSA is an HM-EA/SA that can be individually classified into the *integrative relay* category.

C. Grammar for HMs-EA/SA

The HMs-EA/SA are a specialized category of the HMs, and therefore any HM-EA/SA should be recognizable by the grammar proposed by Talbi [3]. In this section, we present a specialization of this grammar that recognizes only the HMs-EA/SA (Fig. 9). Undefined non-terminal and terminal symbols can be consulted in Talbi's grammar.

Notice that we have decided to consider the parallel evolution of a set of agents (<HTH>) as a kind of evolutionary method (<EA-metaheuristic>). Besides this, we have included the terminal symbol *null*, which does nothing, in order to simplify the <HRH-EA/SA> definition.

III. EXPERIMENTAL FRAMEWORK

A. Test Problems

In this section, we detail the test problems that were used for the upcoming empirical studies. The test suite is composed of 27 binary combinatorial optimization problems, 13 of which were artificial problems and the remaining 14 were obtained from real-world applications. Table II outlines their name,

<HM-EA/SA> → <design-issues-EA/SA><implementation-issue>
<design-issues-EA/SA> → <hierarchical-EA/SA><flat>
<hierarchical-EA/SA> → <LRH-EA/SA> <LTH-EA/SA> <HRH-EA/SA>
<HTH-EA/SA> LRH(<metaheuristic>(<HM-EA/SA>))
LTH(<metaheuristic>(<HM-EA/SA>))
<LRH-EA/SA> → LRH(<SA-metahuristic>(<EA-metahuristic>))
LRH(<LRH-EA/SA>(<metaheuristic>))
<LTH-EA/SA> → LTH(<EA-metahuristic>(<SA-metahuristic>))
LTH(<LTH-EA/SA>(<metaheuristic>))
LTH(<HTH-EA/SA>(<metaheuristic>))
<HRH-EA/SA> → HRH(<EA-metahuristic>+<SA-metahuristic>)
HRH(<met-or-null>+HRH(<HM-EA/SA>+<met-or-null>))
HRH(<SA-metahuristic>+<EA-metahuristic>)
<HTH-EA/SA> → HTH(<EA-metahuristic>, <SA-metahuristic>)
HTH(<metaheuristic>, <HM-EA/SA>)
<EA-metahuristic> → EA
LTH(<EA-metahuristic>(<metaheuristic>)) <HTH>
<SA-metahuristic> → SA
LRH(<SA-metahuristic>(<metaheuristic>))
<met-or-null> → <metaheuristic> null
<metaheuristic> → LS TS SA GA ES GP GH AC SS
NM CLP <HM-EA/SA>

Fig. 9. Grammar for HMs-EA/SA.

number of bits (D), a value (f^*) that stands for either the fitness value of the global optimum, known best solution, or upper bound presented in the literature, and reference. All of them have been formulated as maximization problems. BQP and Maxcut instances can be obtained from the corresponding files from the *BiqMacLibrary* (<http://biqmac.uni-klu.ac.at>), and Multiple knapsack problems, from the *SAC – 94Suite* (<http://elib.zib.de/pub/Packages/mp-testdata/ip/sac94-suite>).

B. Running Conditions

In order to perform a fair comparison between different search methods, we will run every algorithm with the same budget of fitness evaluations. Each run of a search algorithm on a test problem will perform at most 10^5 fitness evaluations. The performance measure is the average of the best fitness values found over 50 independent runs, because we are interested in the regular performance of the compared algorithms. Moreover, the evolution of the best solution found throughout the whole execution will be studied as well.

Nonparametric tests have been used to compare the results of different optimizers [68] given they do not require explicit conditions for being conducted. In particular, mean ranking for each algorithm is first computed according to the Friedman test [69]. This measure is obtained by computing, for each problem, the ranking r_j of the observed result for algorithm j , assigning to the best of them the ranking 1, and to the worst the ranking J (J is the number of algorithms). Then, an average measure is obtained from the rankings of this method for all the test problems. Second, the *Iman and Davenport* test [70] is applied to check the existence of performance differences between all the considered algorithms. Finally, the *Holm* test [71] is used to detect performance differences between the

TABLE II
TACKLED TEST PROBLEMS

Prob.	Name	D	f^*	Ref.
1	Royal road problem (400, 8)	400	1	[57]
2	Trap problem	36	220	[58]
3	Deceptive problem	39	390	[59]
4	Bipolar deceptive problem	396	1	[60]
5	Overlapping deceptive problem	399	1	[60]
6	M-Sat(100, 1200, 3)	100	1	[61]
7	M-Sat(100, 2400, 3)	100	1	[61]
8	NkLand(48, 4)	48	1	[62]
9	NkLand(48, 12)	48	1	[62]
10	HIFF(2, 5, true)	32	192	[63]
11	HIFF(3, 4, false)	81	211	[63]
12	PPeaks(10, 100)	100	1	[62]
13	PPeaks(100, 100)	100	1	[64]
14	PPeaks(50, 150)	150	1	[64]
15	PPeaks(50, 200)	200	1	[64]
16	BQP(bqp50-1)	50	2098	[65]
17	BQP(bqp500-1)	500	116586	[65]
18	BQP(bc120.3.3)	120	Not known	[65]
19	BQP(be200.8.5)	200	Not known	[65]
20	Maxcut(pm1s_80.6)	80	73	[66]
21	Maxcut(w09_100.2)	100	2738	[66]
22	Maxcut(g05_100.5)	100	1436	[66]
23	Maxcut(pw05_100.6)	100	8217	[66]
24	Maxcut(ising2.5_250_5555)	250	7919449	[66]
25	Multiple knapsack p. (weish03)	30	4115	[67]
26	Multiple knapsack p. (pet5)	28	12400	[67]
27	Multiple knapsack p. (pb4)	29	95168	[67]

best ranked algorithm and the remainder. Moreover, we have used the *Wilcoxon matched-pairs signed-ranks* test to perform pairwise comparisons.

IV. COMPARATIVE STUDY OF THE HMs BASED ON SA AND EAS

A representative set of the algorithms revised in Section II is chosen to study their behavior experimentally. This set has been built combining recent proposals and those that best fit the general scheme of each category. Parameters related to the SA process are fixed according to the study performed in [54], for those algorithms whose original publications were not clear: 100 iterations per cooling event, 0.99 as the cooling factor, geometric cooling, and logistic acceptance mechanism. Regarding the initial temperature, two random solutions are first generated. We set a desired probability p_d of accepting the worst solution from the best one. Then, we compute the corresponding T_0 value, according to the applied acceptance criterion. The p_d value is set to 0.4. We have followed the recommendations of the original publications for the remaining parameters. Next, we specify the rest of the parameter settings of the HMs-EA/SA considered, grouped attending to their corresponding category. Moreover the specification according to the grammar presented in Section II-C is detailed for each category or algorithm.

Teamwork Collaborative HMs-EA/SA (<HTH-EA/SA>) (Section II-A1)

- 1) Multiple EAs and SAs: we have developed an algorithm following the guidelines in Section II-A1 (Fig. 3) that is called *DCHCSA*. It considers two CHC

algorithms and two SA processes that are executed in parallel. The ring topology, alternating CHC and SA, is applied. Every certain number of fitness evaluations, the current solution of each SA is sent to the next CHC algorithm anticlockwise. This solution replaces the worst solution in the population of the CHC algorithm. At the same time, each CHC algorithm sends its best solution found so far to the next SA process anticlockwise. This solution becomes the new current solution of this SA process. The migration process takes place every 75 fitness evaluations of each algorithm.

- 2) Multiple SAs: two approaches that belong to this category are considered in the study. SSSA [23] maintains 100 samplers. The neighborhood size is set to 1, i.e., each sampler may accept any of the two states of its consecutive samplers (just one for the extreme samplers). CSA [35] has five samplers. Its parameter α , for the variance control procedure, is set to 0.05.

Relay Collaborative HMs-EA/SA (Section II-A2)

- 1) EA then SA (HRH(EA+SA)): HHSAGA [38] considers a generational GA. Its population size is set to 60. Two-point-crossover is used. In order to adapt the algorithm to the experimental framework described, only two steps (one GA step and another SA step) are considered and fitness evaluations are shared between GA and SA. The GA stage consumes 75% of fitness evaluations and the SA stage the remaining 25%.
- 2) SA then EA (HRH(SA+EA)): GA-PSA implements the approach presented in Section II-A2, in which the population of a GA is initialized by an SA [Fig. 5(b)]. The GA uses a population of 60 individuals. Mutation (one-flip) and crossover (two-point-crossover) are considered to evolve the individuals of the population. The method used by the SA to initialize the population is similar to that proposed in [40]. The population is filled with the solutions generated within the SA trajectory until it rejects the new solution proposed by the neighborhood operator. At that moment, a new SA trajectory starts from another random solution. This process is repeated until the population is filled up.

Teamwork Integrative HMs-EA/SA (LTH(EA(SA))) (Section II-B1)

- 1) *MA with SA as local search procedure:* AGA [40] uses a mutation probability equal to $1/(10 \cdot D)$. The population size is set to 60.
- 2) *SA-based EA selection:* GESA [44] maintains a population of 100 individuals distributed into 10 families. The offspring are generated using two-step crossover.
- 3) *SA-based EA mutation:* ARSAGA [47] sets the parameters as follows: population size equal to 40, crossover and mutation probability equal to 0.7 and 0.01, respectively, $\alpha = 0.9$, $\beta = 0.02$, and $N_{\text{frozen}} = 10$. Two-point crossover is applied.
- 4) *SA-based EA replacement:* PRSA [50] applies a mutation probability equal to $1/D$ and the population size is set to 60.

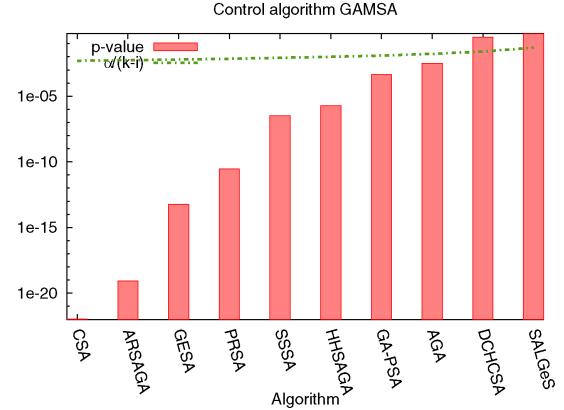


Fig. 10. Holm test results.

Relay Integrative HMs-EA/SA (LRH(SA(EA))) (Section II-B2)

- 1) *EA-based SA component:* for this paper we have implemented a simplified version of SALGeS [54] in which the individual from the population selected to perform the crossover is chosen randomly and the individual discarded from the current solution and the candidate one replaces the worst solution from the GA population (only if it is better). It considers a population size equal to 500 and a probability $p_y = 1/D$ associated with the crossover operator.

Teamwork Collaborative/Integrative Relay

- 1) *GAMSA* [37] (LTH(HTH(SA, HTH(SA, ...))(EA))): the number of simultaneous SA processes is set to 32, the maximum number considered in the original publication. The different SAs act sequentially, performing individual iterations, and use the same selection and replacement operators as the simplified SALGeS.

We have performed an experimental study comparing the previous HMs-EA/SA in the experimental framework described in Section III. In Fig. 10, we can see graphically the Holm test results (the Iman-Davenport test finds significant performance differences between the considered algorithms because its statistical value, 117.166, is greater than its critical one, 1.822). The Holm test finds that the best results are achieved by GAMSA. It detects significant differences between GAMSA and CSA, ARSAGA, GESA, PRSA, SSSA, HHSAGA, GA-PSA, and AGA.

Table III summarizes the results of applying the Wilcoxon test, with $p\text{-value} = 0.05$ and 0.1 , to compare the results of GAMSA and the algorithms for which the Holm test does not detect significant differences (DCHCSA and SALGeS). The last column indicates whether GAMSA performs statistically better (+), worse (-), or without significant differences (~) to its competitor. According to these results, the Wilcoxon test determines that GAMSA is statistically better than DCHCSA and SALGeS with $p\text{-value} = 0.1$. It is worth noting the good behavior of DCHCSA, a simple HM-EA/SA variant (teamwork collaborative approach) derived from distributed GAs that, as far as we know, had not been implemented before.

Table VI, in Appendix A, shows the averaged results for each test problem. Moreover t-test results, comparing GAMSA

TABLE III

GAMSA VERSUS OTHER HMs-EA/SA (WILCOXON'S TESTS WITH p -VALUE = 0.05 AND 0.1 AND CRITICAL VALUE = 107 AND 119, RESPECTIVELY)

Algorithms	R^+	R^-	Sig. differences? 0.05/0.1
GAMSA versus DCHCSA	262	116	~/+
GAMSA versus SALGeS	259	113	~/+

(the best performing HM-EA/SA) and the remaining approaches, indicate for each instance whether GAMSA results are statistically better (+), worse (-), or equal (~) to those of the other HMs-EA/SA. Regarding to these results, we see the following.

- 1) In 23 out of 27 instances, SALGeS and GAMSA reach the best solution found by any other HM-EA/SA approach.
- 2) GAMSA obtains better results than PRSA, ARSAGA, GESA, HHSAGA, SSSA, and CSA for almost every test problem. For the remainder, the test performed does not show statistical differences between its results and those of its competitors.
- 3) Only GA-PSA, AGA, DCHCSA, and SALGeS are able to significantly improve the GAMSA results in some instances. GA-PSA beats GAMSA on one problem, AGA on two, DCHCSA on four, and SALGeS on five.

Next, we study the averaged behavior of the HM-EA/SA instances along the whole run by means of convergence graphs, as follows (Fig. 11).

- 1) Taking into consideration the highest and lowest fitness values achieved by all the algorithms for each test problem, we have normalized every result, throughout all the runs, to the interval [0, 1].
- 2) Then, mean values over the 27 problems have been obtained for each algorithm throughout the 10^5 evaluations.

In addition, Fig. 12 shows the evolution of the averaged rankings over the 27 problems of all the algorithms, when their current best solutions according to the number of consumed evaluations are compared. Notice that Figs. 11 and 12 depict two different kinds of information; in particular, averaged convergence graphs may be strongly affected by distant fitness values in a few functions, e.g., Royal Road; by contrast, rankings are robust with regards to distant fitness values, but may be strongly affected by small changes to many functions.

According to Fig. 11, we may remark the following.

- 1) In the beginning of the execution, all the algorithms but PRSA, GESA, ARSAGA, CSA, and SALGeS move rapidly toward better solutions.
- 2) Over the course of the 40000 evaluations, SALGeS defeats all its competitors, considering the averaged fitness.
- 3) In the intermediate stage of the execution, GAMSA catches all its competitors but SALGeS and outperforms them.
- 4) In the latter stages, SALGeS seems to outperform GAMSA according to the convergence graph. This behavior can be explained by the fact that, as we mentioned before, averaged convergence graphs may be strongly

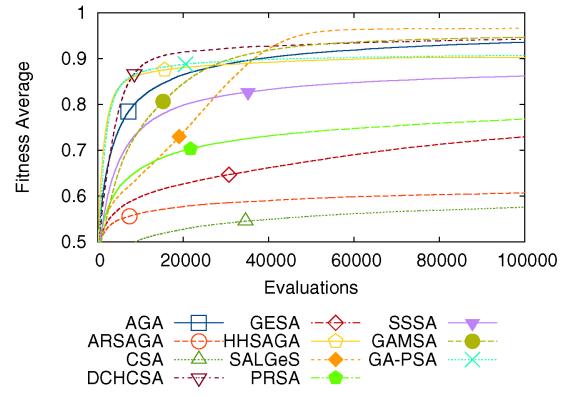


Fig. 11. Convergence graphs of the HMs-EA/SA.

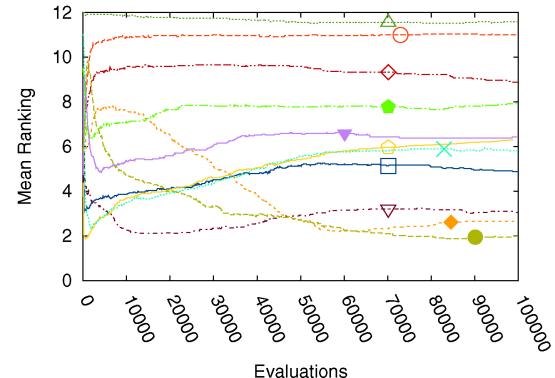


Fig. 12. Evolution of the ranking of the HMs-EA/SA.

affected by distant fitness values in a few functions. However, according to the Wilcoxon test results for p -value = 0.1 in Table III, we see that GAMSA outperforms SALGeS considering a statistical study.

According to Fig. 12, we observe the following.

- 1) All the algorithms but GAMSA, SALGeS, GA-PSA, and HHSAGA more or less maintain their respective rankings from the beginning of the execution until the end.
- 2) GAMSA and SALGeS show an improving trend from the beginning and reach the best rankings at the intermediate stages.
- 3) GA-PSA and HHSAGA have their best rankings at the very beginning of the execution, but lose competitiveness gradually.
- 4) At the end of the execution, GAMSA is the most competitive algorithm for all the problems in general.

V. PERFORMANCE OF HMs-EA/SA BY CATEGORIES

In this section, we study the behavior of the HMs-EA/SA by categories in order to assess the success of each group of algorithms. In Fig. 13, we can see the average ranking obtained by each algorithm and the category to which it belongs. HMs-EA/SA have been grouped by categories and the height of each column is proportional to the ranking. Therefore, the lower a column is, the better its associated algorithm is.

Taking into account this figure, we can conclude the following.

- 1) The “multiple SAs” category includes the CSA algorithm, which presents the poorest performance among all the studied algorithms. The other algorithm in this class, SSSA, appears in the group of the five algorithms with the worst average rankings. The three instances of the “SA-based EA component” category (ARSAGA, GESA, and PRSA) are among the algorithms forming this group as well. A feature shared by these two categories is that the hybridization EA/SA is built employing incomplete formulations of either EA or SA. Specifically, an effort is made to incorporate principles of one of these metaheuristics in the other (population of solutions in SA, multiple SAs, and SA acceptance criterion in EA, SA-based EA component)
- 2) Better performances are achieved by the algorithms in the other categories. In this case, they are conceived as HMs composed of self-contained EAs and SAs, and, cooperating somehow.
- 3) The integrative approach is present in three out of the four best ranked HMs-EA/SA. They are AGA (MA with SA as local search procedure), SALGeS, and GAMSA (both belonging to EA-based SA component). In these HMs, one metaheuristic is specialized to play a specific role inside the other metaheuristic. Specifically, in AGA, SA affords EA refined solutions and, in SALGeS and GAMSA, EA acts as an SA neighborhood operator.
- 4) The hybridization technique used by SALGeS and GAMSA, that involves replacing some components in metaheuristics with customized EAs (*evolutionary components*) to develop the same work more effectively and with a relatively low computational cost, was also studied in [22]. As an example, the authors contributed an iterated local search algorithm with an evolutionary perturbation method, which is a micro EA that effectively explores in the neighborhood of particular solutions. This algorithm turned out to be very competitive with state-of-the-art iterated local search metaheuristics for binary optimization problems. The suitable results shown by GAMSA and SALGeS provide additional evidence for the effectiveness of this novel hybridization paradigm.
- 5) The group of the four best performing algorithms is completed by an HM-EA/SA belonging to the team-work collaborative approach (DCHCSA). Interestingly, it has better results than the two algorithms representing the relay collaborative approach (HHSAGA and GA-PSA).

We can highlight that the ranking obtained when comparing the performance of the different HM-EA/SA approaches (Fig. 13) may be used as a guideline for researchers attempting to tackle specific optimization problems by building HMs-EA/SA, or even with other trajectory-based metaheuristics (tabu search, local search procedures, guided local search, and so on).

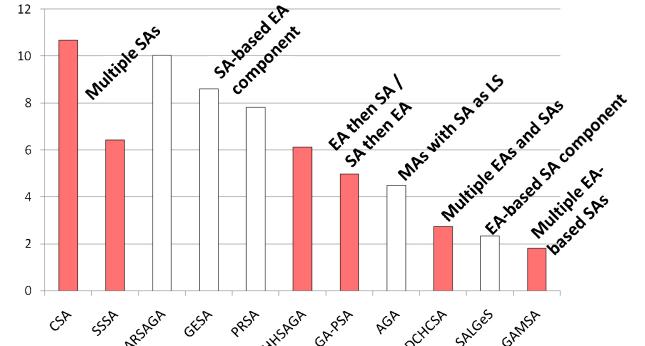


Fig. 13. Average rankings of the HMs-EA/SA versions.

VI. SYNERGY TEST

One of the most important aspects of studying the behavior of an HM is to consider the synergy produced by the combination of the composing metaheuristics. In fact, exploring the complementary character of the different optimization strategies involved in the hybrid metaheuristics is the main motivation behind the hybridization, that is, hybrids are believed to benefit from synergy [72]. In order to assess the amount of synergy that appears when combining two or more components, or whether it does or does not appear, the usual practice involves the comparison between the hybrid algorithm and the sole usage of its components [73]–[75]. In this section, we study the synergy produced by combining SA and EAs of previous HMs-EA/SA with regard to the following algorithms.

- 1) Standalone SA (SA) [6], [7]. The parameter values are those detailed in Section IV.
- 2) Canonical generational GA (CGGA). This considers a population of 60 individuals, binary tournament selection, one-flip mutation (probability 0.006), and two-point crossover (probability 1).
- 3) Canonical steady-state GA (CSSGA). The parameters and operators are the same as in CGGA. In each iteration, two offspring that replace their parents are generated.
- 4) Cross-generational elitist selection, heterogeneous recombination, and cataclysmic mutation (CHC) [76]. This is an EA that involves the combination of a selection strategy with a very high selective pressure, and several components inducing diversity.

Table IV (column “Comp. on results”) summarizes the results of applying the Wilcoxon test with p -value = 0.05, where the values of R^+ (associated to the HM-EA/SA) and R^- (associated to the single metaheuristic) of the test are specified. According to these results, we may conclude the following.

- 1) SALGeS and GAMSA are the only HMs that perform better than the sole application of SA, any canonical EA, and CHC. So, we can state that this hybrid approach (EA-based SA component) supplies a better performance, exploiting and uniting the advantages of the individual metaheuristics and, therefore, showing a profitable synergy.
- 2) Considering the remaining hybridizations, only DCHCSA presents a positive result in terms of synergy.

The other ones do not produce better results, for the considered experimental framework, than the sole application of one of their intrinsic metaheuristics. These results allow us to conclude that the simple combination of several metaheuristics does not ensure success. It is necessary to study the way the composing metaheuristics are combined in order to achieve a positive synergy between them.

Finally, it is interesting to analyze the computational time required for the HMs with regards to their standard counterparts. Table IV (column “Comp. on time”) shows the results of applying the Wilcoxon test, using as performance measure the average over 50 independent runs of the time employed to perform 10^5 fitness evaluation on each problem. The last column indicates whether the corresponding HMs require statistically more time (–), less (+), or there are no significant differences (~) with regards to their standard counterparts. The Wilcoxon test results show that all the HMs, but HHSAGA, take statistically more time than SA to solve the considered problems. On the contrary, it is significant that there are no HMs exceeding the computational time required by the most computationally expensive counterparts, the evolutionary algorithms (CGGA, CSSGA, and CHC). In particular, we can highlight that SALGeS and GAMSA, which are the only ones that reach a profitable synergy and outperform the quality of the solutions provided by all their standard counterparts, do not exceed computational time required by their most expensive counterparts, CGGA, CSSGA, and CHC.

VII. BEST PERFORMING HM-EA/SA VERSUS STATE-OF-THE-ART EAS FOR BINARY COMBINATORIAL PROBLEMS

In this section, we intend to assess the performance of GAMSA, the best performing HM-EA/SA, with regards to other relevant EAs for binary combinatorial problems found in the literature. The experiments are carried out on the test suite detailed in Section III-A.

- 1) CHC [76] was tested against different GAs, giving better results, especially for hard problems [77]. So, it has arisen as a reference point in the literature. Its population consists of 50 individuals.
- 2) Variable assortative mating GA (VDMGA) [78]. This is a steady-state EA, similar to CHC, in which the number of new chromosomes entering the population in each generation is controlled on-line by a threshold value, genetic diversity, and the population’s state of convergence. The results in [78] show the superior performance of VDMGA when compared to other GAs. Its population size is set to 100. The mutation probability is set to 0.006.
- 3) Context-independent scatter search (CISS) [79]. CISS is a proposal explicitly designed to tackle general binary optimization problems. Its performance was compared with that of several general-purpose commercial optimization tools, obtaining promising results. Population and reference set sizes have been set to 300 and 6,

TABLE IV
HMS-EA/SA VERSUS STANDALONE SA, CANONICAL GAS, AND CHC
(COMPARISON ON RESULTS AND TIME) (WILCOXON’S TEST WITH
 p -VALUE = 0.05 AND CRITICAL VALUE = 107)

HMs	Single Metah.	Comp. on Results			Comp. on Time		
		R^+	R^-	Diff	R^+	R^-	Diff
CSA	SA	0	378	–	64	314	–
	CGGA	0	378	–	321	57	+
	CSSGA	100	278	–	358	20	+
	CHC	0	378	–	309	69	+
ARSAGA	SA	0	378	–	24	354	–
	CGGA	0	278	–	262	116	~
	CSSGA	16	362	–	60	318	–
	CHC	0	378	–	277	101	+
GESA	SA	0	378	–	38	340	–
	CGGA	0	378	–	298	80	+
	CSSGA	4	374	–	378	0	+
	CHC	0	378	–	301	77	+
PRSA	SA	0	378	–	27	351	–
	CGGA	10	368	–	288	90	+
	CSSGA	0	378	–	279	99	+
	CHC	1	377	–	277	101	+
SSSA	SA	0.5	377.5	–	48	330	–
	CGGA	33.5	344.5	–	329	49	+
	CSSGA	378	0	+	378	0	+
	CHC	1	377	–	367	11	+
HHSAGA	SA	8	370	–	116	262	~
	CGGA	88	290	–	378	0	+
	CSSGA	377	1	+	378	0	+
	CHC	46	332	–	378	0	+
GA-PSA	SA	7.5	370.5	–	68	310	–
	CGGA	0	378	–	359.5	18.5	+
	CSSGA	158.5	219.5	~	210	168	~
	CHC	52.5	325.5	–	355	23	+
AGA	SA	30.5	347.5	–	81	297	–
	CGGA	237.5	140.5	~	339	39	+
	CSSGA	378	0	+	323	55	+
	CHC	129.5	248.5	~	345	33	+
DCHCSA	SA	206.5	171.5	~	32	346	–
	CGGA	330.5	47.5	+	276	102	+
	CSSGA	378	0	+	306	72	+
	CHC	328.5	49.5	+	325	53	+
SALGeS	SA	287.5	90.5	+	27	352	–
	CGGA	374.5	3.5	+	174	204	~
	CSSGA	378	0	+	192	186	~
	CHC	343.5	34.5	+	160	218	~
GAMSA	SA	277.5	101.5	+	27	351	–
	CGGA	352.5	25.5	+	188	190	~
	CSSGA	378	0	+	214	164	~
	CHC	333.5	44.5	+	199	179	~

respectively. The other parameters have been set as in [79].

- 4) Sawtooth GA (Saw-GA) [80]. This uses a variable population size and periodic partial reinitialization of the population in the form of a saw-tooth function. For a wide range of problems, the performance of Saw-GA was statistically superior to a standard GA and a micro GA. The average population size is set to 80. Crossover and mutation probabilities are set to 0.85 and 0.05, respectively. Period and amplitude parameters are adjusted to 40 and 75, respectively.
- 5) Versatile quantum-inspired EA (vQEA) [81]. vQEA is a novel EA approach based on quantum computing principles. It considers the quantum bit (Qbit) as the smallest information unit, which is defined by the probability

TABLE V
GAMSA VERSUS STATE-OF-THE-ART EAS (WILCOXON'S TEST WITH
 p -VALUE = 0.05 AND CRITICAL VALUE = 107)

Algorithms	R^+	R^-	Sig. Differences?
GAMSA versus CHC	338.5	39.5	+
GAMSA versus VDMGA	286	92	+
GAMSA versus CISS	320	58	+
GAMSA versus Saw-GA	365.5	12.5	+
GAMSA versus vQEA	362.5	15.5	+

at which the corresponding state (0 or 1) is likely to appear when it is collapsed, i.e., read or measured. vQEA considers a population of quantum individuals (a quantum individual is composed of a Qbit string, a realization, and an attractor) that evolve through quantum gate operations. The population of vQEA is divided into g groups each containing d individuals. Attractors are periodically synchronized between individuals in the same group and between different groups. vQEA considers a population of ten individuals distributed into a unique group. The synchronization of attractors takes place every generation and $\Delta\Theta$, associated to quantum gate, is set to $\pi/100$.

Table V summarizes the results of applying the Wilcoxon test for p -value = 0.05, where the values of R^+ (associated to GAMSA) and R^- of the test are specified. We notice that GAMSA obtains statistically better results than those of the other EAs (R^- values are lower than both R^+ ones and critical values).

Table VI, in Appendix A, shows the results of the optimizers when tackling each test problem. Moreover t-test results, comparing GAMSA (the best performing approach) and the other algorithms, show whether there are significant differences for each problem instance. Regarding to these results, we observe the following.

- 1) In 19 out of 27 instances, GAMSA reaches the best solution found by any algorithm studied in this section.
- 2) Saw-GA and vQEA are able to reach better results than GAMSA in one instance (Deceptive and Royal Road respectively). CHC and CISS do the same in NK-Land(48, 12). Moreover, CHC outperforms GAMSA in Maxcut(ising2.5-250_5555) and CISS, in Overlapping Deceptive.
- 3) VDMGA, the most competitive algorithm with regards to GAMSA, attains better results on five instances (Royal Road, HIFF(3, 4, false), BQP(bqp500-1), Maxcut(ising2.5-250_5555), and Deceptive instances).

As we see in Fig. 14, at the initial stages, GAMSA experiences lower convergence speed than most of its competitors, except Saw-GA. However, GAMSA continues improving its results at the intermediate and even at the final stages. This fact allows GAMSA to catch its competitors up and to overtake them at the latter stages. It is important to note that, according to the convergence graph, though at the end of the execution VDMGA seems to reach similar solutions to GAMSA, the Wilcoxon test showed that GAMSA defeats VDMGA. This can be explained by the fact that the differences achieved by

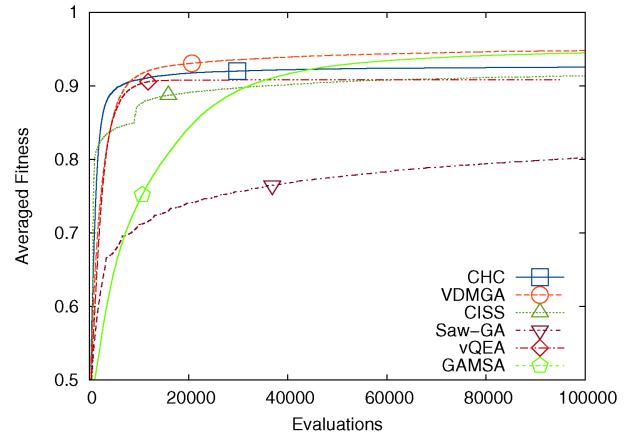


Fig. 14. Convergence graphs of state-of-the-art EAs and GAMSA.

TABLE VI
RESULTS OF THE BINARY OPTIMIZERS ON EACH TEST PROBLEM

Pr.	CHC	VDMGA	Saw-GA	CISS	vQEA	GAMSA
1	0.337+	1-	0.192+	0.130+	0.536-	0.394
2	220-	220~	219~	210+	219+	220
3	378~	382-	385-	378~	366+	378
4	0.878+	0.850+	0.853+	0.892+	0.886+	0.899
5	0.899+	0.896+	0.753+	0.915-	0.894+	0.906
6	0.957+	0.957+	0.943+	0.955+	0.954+	0.959
7	0.937+	0.938+	0.926+	0.937+	0.936+	0.939
8	0.760+	0.765~	0.755+	0.761+	0.747+	0.766
9	0.746-	0.701+	0.713+	0.751-	0.71+	0.741
10	164+	162+	155+	148+	136+	185
11	170~	176-	155+	152+	154+	169
12	0.994+	1~	0.941+	0.997~	0.999-	1
13	0.990+	0.990+	0.929+	0.991+	0.988+	0.998
14	0.990+	0.992+	0.870+	0.995+	0.996+	0.999
15	0.990+	0.995+	0.831+	0.993+	0.993+	0.998
16	5174-	5176~	5159+	5176~	5168+	5176
17	117 453~	120 948-	51 714+	119 056~	117 402+	119 264
18	11 548+	11 589+	9420+	11 583+	11 530+	11 599
19	49 852+	50 428+	34 427+	50 407+	50 162+	50 465
20	68+	69+	61+	68+	65+	73
21	2623+	2647+	2267+	2681+	2552+	2716
22	1422+	1423+	1385+	1430+	1407+	1435
23	8153+	8168+	7923+	8190+	8102+	8204
24	7526 181-	7 522 897-	4 949 495+	7 333 148+	7 249 980+	7 422 326
25	4005+	4003+	4074~	4096-	3955+	4081
26	12 396+	12 398~	12 385+	12 400-	12 393+	12 400
27	118 204~	118 204~	118 204~	118 204~	118 204~	118 204

GAMSA are not too great, but it reaches positive differences more frequently. The latter causes the Wilcoxon test to detect significant differences.

In conclusion, this paper shows that GAMSA really provides a competitive alternative in the combinatorial binary optimization field, obtaining promising results on the considered testbed.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we provided an overview of the ways EAs and SA may be combined with each other to obtain HMs-EA/SA. We organized the approaches found in the literature by proposing a taxonomy based on those introduced by Talbi and Raidl for HMs [3], [4]. Moreover, we developed, to our knowledge, the first experimental study analyzing a large

TABLE VII
RESULTS OF THE HMS-EA/SA ON EACH TEST PROBLEM

Pr.	PRSA	GA-PSA	ARSAGA	GESA	AGA	HHGASA	SSSA	CSA	DCHCSA	SALGeS	GAMSA
1	0.366 ~	0.579 —	0.085 +	0.104 +	0.690 —	0.548 +	0.123 +	6.52e-02 +	0.338 ~	0.997 —	0.394
2	209 +	217 +	188 +	201 +	216 +	211 +	214 +	179 +	220 ~	220 ~	220
3	375 +	378 ~	363 +	376 +	370 +	374 +	373 +	365 +	382 —	379 —	378
4	0.860 +	0.883 +	0.826 +	0.875 +	0.893 +	0.893 +	0.875 +	0.813 +	0.88 +	0.895 +	0.899
5	0.768 +	0.864 +	0.671 +	0.777 +	0.894 +	0.896 +	0.845 +	0.635 +	0.896 +	0.9 +	0.906
6	0.933 +	0.955 +	0.916 +	0.932 +	0.953 +	0.953 +	0.952 +	0.912 +	0.957 +	0.958 +	0.959
7	0.921 +	0.936 +	0.906 +	0.917 +	0.936 +	0.935 +	0.932 +	0.903 +	0.938 +	0.939 ~	0.939
8	0.727 +	0.744 +	0.673 +	0.72 +	0.759 +	0.737 +	0.741 +	0.674 +	0.766 ~	0.766 ~	0.766
9	0.713 +	0.721 +	0.675 +	0.71 +	0.714 +	0.709 +	0.718 +	0.692 +	0.743 ~	0.737 ~	0.741
10	151 +	126 +	113 +	120 +	179 ~	128 +	160 +	116 +	187 ~	176 +	185
11	159 +	164 +	124 +	144 +	171 —	152 +	156 +	120 +	178 —	176 —	169
12	0.841 +	0.987 +	0.725 +	0.842 +	0.988 +	0.993 +	0.939 +	0.680 +	0.998 ~	0.989 ~	I
13	0.828 +	0.976 +	0.73 +	0.857 +	0.976 +	0.979 +	0.935 +	0.709 +	0.993 +	0.995 +	0.998
14	0.805 +	0.979 +	0.687 +	0.79 +	0.967 +	0.98 +	0.937 +	0.656 +	0.996 +	0.994 +	0.999
15	0.747 +	0.987 +	0.659 +	0.759 +	0.980 +	0.979 +	0.923 +	0.627 +	0.997 +	0.992 ~	0.998
16	4641 +	5164 +	3771 +	4325 +	5176 ~	5146 +	5152 ~	3378 +	5176 ~	5176 ~	5176
17	59 834 +	95 978 +	26 523 +	46 188 +	117 034 +	111 884 +	80 116 +	16 728 +	120 215 —	121 266 —	119 264
18	8140 +	11 588 ~	5022 +	7537 +	11 597 ~	11 408 +	10 792 +	3958 +	11 597 ~	11 599 ~	11 599
19	34 193 +	49 759 +	19 042 +	27 767 +	50 311 +	50 298 +	44 338 +	14 305 +	50 410 +	50 456 +	50 465
20	51 +	68 +	34 +	52 +	70 +	67 +	68 +	31 +	70 +	72 ~	73
21	2122 +	2623 +	1564 +	2076 +	2675 +	2564 +	2529 +	1510 +	2667 +	2700 +	2716
22	1366 +	1420 +	1324 +	1366 +	1422 +	1414 +	1412 +	1319 +	1431 +	1433 +	1435
23	7819 +	8150 +	7514 +	7826 +	8125 +	8123 +	8074 +	7463 +	8187 +	8202 ~	8204
24	5 153 437 +	7 285 193 +	3 230 924 +	4 975 365 +	7 369 494 +	7 275 664 +	6 546 047 +	2 812 404 +	7 545 615 —	7 563 823 —	7 422 326
25	3915 +	3922 +	3732 +	3620 +	3965 +	3793 +	3975 +	3720 +	4053 +	4052 +	4081
26	12 189 +	12 094 +	12 084 +	11 934 +	11 994 +	11 913 +	12 244 +	12 090 +	12 400 ~	12 398 +	12 400
27	116 566 +	118 204 ~	111 258 +	112 481 +	118 204 ~	117 156 ~	118 204 ~	109 247 +	118 204 ~	118 204 ~	118 204

spectrum of HM-EA/SA models from three points of view.

- 1) We compared the performance of the HM-EA/SA models, individually and by categories, extracting relevant conclusions regarding the categories presented in the presented taxonomy.
- 2) We performed a synergy test that has identified two HM-EA/SA models that really present synergistic properties.
- 3) We compared the best performing HM-EA/SA model with the state-of-the-art evolutionary algorithms for binary combinatorial optimization.

Our study allowed us to draw an interesting conclusion: the hybridization of EAs and SA becomes a prospective research area for finding more effective search algorithms.

Finally, we should remark that the research line focused on in this paper is indeed worthy of further studies. We will intend to explore three interesting avenues of research. First, *multiobjective and constrained problems* are subject of great interest, existing in the literature a great number of proposals for dealing with this kind of problems based on EAs [82]–[84] and SA [85], [86]. Therefore, it is possible to adapt the design of HMs-EA/SA to this type of problems. Second, teamwork collaborative HMs-EA/SA are able to take advantage of parallel hardware (multicore processors, clusters, and so on) and software [87] that has become very affordable and widely available nowadays. This clearly favors the *implementation on parallel hardware* of HMs-EA/SA [25], [50], [51] that may lead to improved results due to the speed-up in the search process, which becomes a very appealing option for dealing with large-scale optimization tasks. Finally, an important issue when combining different algorithms concerns the number of evaluations that each algorithm should consume throughout the run to create the conditions for the appearance of collabora-

tive synergies among all the composing algorithms. Adaptive strategies that identify the best performing technique at each phase of the evolution with a minimum overhead [18], [19], [88] may be used to built *adaptive HMs-EA/SA* with the aim of allowing profitable synergies to arise from the adjusted intervention of EAs and SA.

APPENDIX A RESULTS

Tables VI and VII show the average fitness values obtained by the studied search algorithms. The best results for each problem are presented in boldface. Moreover, we have carried out a statistical analysis with *p*-value equal to 0.05 to measure the performance differences between GAMSA, the best performing HM-EA/SA, and the other algorithms for each problem separately. This statistical analysis is based on *t-test*, when normality and heteroscedasticity conditions are satisfied, and *Mann-Whitney*, otherwise [69], [89].

REFERENCES

- [1] F. Glover and G. Kochenberger, Eds., *Handbook of Metaheuristics*. Dordrecht, The Netherlands: Kluwer, 2003.
- [2] P. Siarry and Z. Michalewicz, Eds., *Advances in Metaheuristics for Hard Optimization* (Natural Computing). Berlin, Germany: Springer, 2008.
- [3] E. Talbi, “A taxonomy of hybrid metaheuristics,” *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.
- [4] G. Raidl, “A unified view on hybrid metaheuristics,” in *Hybrid Metaheuristics*, LNCS 4030, F. Almeida, M. B. Aguilera, C. Blum, J. M. Vega, M. P. Pérez, A. Roli, and M. Sampels, Eds. Berlin, Germany: Springer, 2006, pp. 1–12.
- [5] C. Blum, “Hybrid metaheuristics—guest editorial,” *Comput. Oper. Res.*, vol. 37, no. 3, pp. 430–431, 2010.
- [6] S. Kirkpatrick, C. Gelatt, Jr., and M. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

- [7] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. New York: Wiley, 1989.
- [8] D. Henderson, S. Jacobson, and A. Jacobson, "The theory and practice of simulated annealing," in *Handbook of Metaheuristics*. Dordrecht, The Netherlands: Kluwer, 2003, pp. 287–319.
- [9] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.
- [10] P. Salamon, P. Sibani, and R. Frost, *Facts, Conjectures and Improvements for Simulated Annealing* (Monographs on Mathematical Modeling and Computation). Philadelphia, PA: SIAM, 2002.
- [11] T. Bäck, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. London, U.K.: Institute of Physics, 1997.
- [12] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Berlin, Germany: Springer-Verlag, 2003.
- [13] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. New York: IEEE Press, 1995.
- [14] H. Beyer and H. Schwefel, "Evolution strategies: A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, 2002.
- [15] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Harlow, U.K.: Addison-Wesley Longman, 1989.
- [16] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: Methodologies, architectures, and reviews," in *Hybrid Evolutionary Algorithms*, C. Grosan, A. Abraham, and H. Ishibuchi, Eds. Berlin, Germany: Springer, 2007, pp. 1–17.
- [17] P. Preux and E. Talbi, "Toward hybrid evolutionary algorithms," *Int. Trans. Oper. Res.*, vol. 6, no. 6, pp. 557–570, 1999.
- [18] R. Mallipeddi and P. Suganthan, "Ensemble of constraint handling techniques," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 561–579, Aug. 2010.
- [19] J. Vrugt, B. Robinson, and J. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 243–259, Apr. 2009.
- [20] Z. Lü, F. Glover, and J. Hao, "A hybrid metaheuristic approach to solving the UBQP problem," *Eur. J. Oper. Res.*, vol. 207, no. 3, pp. 1254–1262, Dec. 2010.
- [21] C. Cotta and A. Fernandez, "A hybrid GRASP: Evolutionary algorithm approach to Golomb ruler search," in *Proc. 8th PPSN*, LNCS 3242. 2004, pp. 481–490.
- [22] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 481–497, Mar. 2010.
- [23] D. Thompson and G. Bilbro, "Sample-sort simulated annealing," *IEEE Trans. Syst. Man Cybern. B*, vol. 35, no. 3, pp. 625–632, Jun. 2005.
- [24] G. Liu, D. Zhou, H. Xu, and C. Mei, "Model optimization of SVM for a fermentation soft sensor," *Expert Syst. Applicat.*, vol. 37, no. 4, pp. 2708–2713, Apr. 2010.
- [25] Z. Wang, Y. Wong, and M. Rahman, "Development of a parallel optimization method based on genetic simulated annealing algorithm," *Parallel Comput.*, vol. 31, nos. 8–9, pp. 839–857, 2005.
- [26] H. Cheng, X. Wang, S. Yang, and M. Huang, "A multipopulation parallel genetic simulated annealing-based QoS routing and wavelength assignment integration algorithm for multicast in optical networks," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 677–684, 2009.
- [27] L. Tang and X. Wang, "An improved particle swarm optimization algorithm for the hybrid flowshop scheduling to minimize total weighted completion time in process industry," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1303–1314, Nov. 2010.
- [28] I. Obregon and A. Pawlowsky, "A hybrid SA-EA method for finding the maximum number of switching gates in a combinational circuit," *IEICE Electron. Express*, vol. 5, no. 18, pp. 756–761, 2008.
- [29] Y. Xu and R. Qu, "Solving multiobjective multicast routing problems by evolutionary multiobjective simulated annealing algorithms with variable neighborhoods," *J. Oper. Res. Soc.*, vol. 62, pp. 313–325, Feb. 2011.
- [30] B. Firouzi, M. Sadeghi, and T. Niknam, "A new hybrid algorithm based on PSO, SA, and k-means for cluster analysis," *Int. J. Innov. Comput. Inf. Control*, vol. 6, no. 7, pp. 3177–3192, 2010.
- [31] W. Hong, Y. Dong, L. Chen, and C. Lai, "Taiwanese 3G mobile phone demand forecasting by SVR with hybrid evolutionary algorithms," *Expert Syst. Applicat.: An Int. J.*, vol. 37, no. 6, pp. 4452–4462, Jun. 2010.
- [32] R. M'Hallah, "Minimizing total earliness and tardiness on a single machine using a hybrid heuristic," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 3126–3142, Oct. 2007.
- [33] A. Tantar, N. Melab, and E. Talbi, "A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction," *Soft Comput. A Fusion Found. Methodol. Applicat.*, vol. 12, no. 12, pp. 1185–1198, 2008.
- [34] S. Bhandarkar and H. Zhang, "Image segmentation using evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 1–21, Apr. 1999.
- [35] S. Xavier-de-Souza, J. Suykens, J. Vandewalle, and D. Bollé, "Cooperative behavior in coupled simulated annealing processes with variance control," in *Proc. Symp. Nonlinear Theory Its Applicat.*, 2006, pp. 114–119.
- [36] M. Aydin and V. Yigit, *Parallel Simulated Annealing* (Parallel Metaheuristics: A New Class of Algorithms). New York: Wiley, 2005, pp. 267–288.
- [37] F. Rodriguez-Diaz, C. Garcia-Martinez, and M. Lozano, "A GA-based multiple simulated annealing," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 195–201.
- [38] D. Chen, C. Lee, C. Park, and P. Mendes, "Parallelizing simulated annealing algorithms based on high-performance computer," *J. Global Optimization*, vol. 39, no. 2, pp. 261–289, 2007.
- [39] D. Brown, C. Huntley, and A. Spillane, "A parallel genetic heuristic for the quadratic assignment problem," in *Proc. 3rd Int. Conf. Genet. Algorithms*, 1989, pp. 406–415.
- [40] F. Lin, C. Kao, and C. Hsu, "Applying the genetic approach to simulated annealing in solving some NP-hard problems," *IEEE Trans. Syst. Man. Cybern.*, vol. 23, no. 6, pp. 1752–1767, Nov.–Dec. 1993.
- [41] X. Li and X. Wei, "An improved genetic algorithm-simulated annealing hybrid algorithm for the optimization of multiple reservoirs," *Water Res. Manag.*, vol. 22, no. 8, pp. 1031–1049, 2007.
- [42] Q. Zhang, J. Wang, C. Jin, and Q. Zeng, "Localization algorithm for wireless sensor network based on genetic simulated annealing algorithm," in *Proc. 4th Int. Conf. WiCOM*, 2008, pp. 1–5.
- [43] D. Goldberg, "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing," *Complex Syst.*, vol. 4, no. 4, pp. 445–460, 1990.
- [44] P. Yip and Y. Pao, "Combinatorial optimization with use of guided evolutionary simulated annealing," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 290–295, 1995.
- [45] M. De la Maza and B. Tidor, "Increased flexibility in genetic algorithms: The use of variable Boltzmann selective pressure to control propagation," in *Proc. ORSA CSTS Conf. Comput. Sci. Oper. Res.: New Develop. Their Interfaces*, 1992, pp. 425–440.
- [46] B. Li and W. Jiang, "A novel stochastic optimization algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. 30, no. 1, pp. 193–198, Feb. 2000.
- [47] S. Hwang and R. He, "A hybrid real-parameter genetic algorithm for function optimization," *Adv. Eng. Inform.*, vol. 20, no. 1, pp. 7–21, 2006.
- [48] W. Han and P. Que, "Defect reconstruction of submarine oil pipeline from MFL signals using genetic simulated annealing algorithm," *J. Jpn. Petr. Inst.*, vol. 49, no. 3, pp. 145–150, 2006.
- [49] D. Adler, "Genetic algorithm and simulated annealing: A marriage proposal," in *Proc. IEEE Int. Conf. Neural Netw.*, Mar.–Apr. 1993, pp. 1104–1109.
- [50] S. Mahfoud and D. Goldberg, "Parallel recombinative simulated annealing: A genetic algorithm," *Parallel Comput.*, vol. 21, no. 1, pp. 1–28, Jan. 1995.
- [51] H. Chen, N. Flann, and D. Watson, "Parallel genetic simulated annealing: A massively parallel SIMD algorithm," *IEEE Trans. Parallel Distributed Syst.*, vol. 9, no. 2, pp. 126–136, Feb. 1998.
- [52] H. Cho, S. Oh, and D. Choi, "A new evolutionary programming approach based on simulated annealing with local cooling schedule," in *Proc. Congr. Evol. Comput.*, 1998, pp. 598–602.
- [53] Z. Yang, Z. Tian, and Z. Yuan, "GSA-based maximum likelihood estimation for threshold vector error correction model," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 109–120, 2007.
- [54] C. García-Martínez and M. Lozano, "Simulated annealing based on local genetic search," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 2009, pp. 2569–2576.
- [55] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 1, pp. 43–63, Apr. 2000.
- [56] 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.
- [57] S. Forrest and M. Mitchell, "Relative building block fitness and the building block hypothesis," in *Foundations of Genetic Algorithms 2*, L. Whitley, Ed. San Francisco, CA: Morgan Kaufmann, 1993, pp. 109–126.

- [58] D. Thierens, "Population-based iterated local search: Restricting neighborhood search by crossover," in *Proc. Genet. Evol. Comput. Conf.*, LNCS 3103, 2004, pp. 234–245.
- [59] D. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Syst.*, vol. 3, no. 4, pp. 493–530, 1989.
- [60] M. Pelikan, D. Goldberg, and E. Cantú-Paz, "Linkage problem, distribution estimation, and Bayesian networks," *Evol. Comput.*, vol. 8, no. 3, pp. 311–340, 2000.
- [61] K. Smith, H. Hoos, and T. Stützle, "Iterated robust tabu search for MAX-SAT," in *Proc. Canad. Soc. Comput. Stud. Intell. Conf.*, LNCS 2671, 2003, pp. 129–144.
- [62] S. Kauffman, "Adaptation on rugged fitness landscapes," in *Lectures in the Sciences of Complexity*, vol. 1. Reading, MA: Addison-Wesley, 1989, pp. 527–618.
- [63] R. Watson and J. Pollack, "Hierarchically consistent test problems for genetic algorithms," in *Proc. Congr. Evol. Comput.*, vol. 2, 1999, p. 1413.
- [64] W. Spears, *Evolutionary Algorithms: The Role of Mutation and Recombination*. Berlin, Germany: Springer, 2000.
- [65] J. Beasley, "Heuristic algorithms for the unconstrained binary quadratic programming problem," Manage. School, Imperial College, London, U.K., Tech. Rep., 1998 [Online]. Available: <http://people.brunel.ac.uk/~mastjjb/jeb/bqpabs.html>
- [66] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. New York: Plenum, 1972, pp. 85–103.
- [67] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms," in *Proc. Congr. Evol. Comput.*, 2002, pp. 980–985.
- [68] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristic*, vol. 15, no. 6, pp. 617–644, 2008.
- [69] J. Zar, *Biostatistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [70] R. Iman and J. Davenport, "Approximations of the critical region of the Friedman statistic," *Commun. Stat.*, vol. 9, no. 6, pp. 571–595, 1980.
- [71] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Stat.*, vol. 6, no. 2, pp. 65–70, 1979.
- [72] C. Blum, J. Puchinger, G. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *App. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.
- [73] C. Antonio, "A study on synergy of multiple crossover operators in a hierarchical genetic algorithm applied to structural optimization," *Structural Multidisciplinary Optimiz.*, vol. 38, no. 2, pp. 117–135, 2009.
- [74] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 782–800, Oct. 2010.
- [75] F. Herrera, M. Lozano, and A. Sánchez, "Hybrid crossover operators for real-coded genetic algorithms: An experimental study," *Soft Comput.*, vol. 9, no. 4, pp. 280–298, 2005.
- [76] L. Eshelman and J. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," in *Proc. Int. Conf. Genet. Algorithms*, 1991, pp. 115–122.
- [77] D. Whitley, S. Rana, J. Dzubera, and E. Mathias, "Evaluating evolutionary algorithms," *Artif. Intell.*, vol. 85, nos. 1–2, pp. 245–276, 1996.
- [78] C. Fernandes and A. Rosa, "Self-adjusting the intensity of assortative mating in genetic algorithms," *Soft Comput.*, vol. 12, no. 10, pp. 955–979, 2008.
- [79] F. Gortazar, A. Duarte, M. Laguna, and R. Martí, "Black box scatter search for general classes of binary optimization problems," *Comput. Oper. Res.*, vol. 37, no. 11, pp. 1977–1986, 2010.
- [80] V. Kourousis and C. Katsaras, "A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 19–28, Feb. 2006.
- [81] M. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1218–1232, Dec. 2009.
- [82] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
- [83] C. Coello, G. Lamont, and D. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer-Verlag, 2006.
- [84] C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Comput. Meth. Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, 2002.
- [85] C. Pedamallu and L. Ozdamar, "Investigating a hybrid simulated annealing and local search algorithm for constrained optimization," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1230–1245, 2008.
- [86] A. Hedar and M. Fukushima, "Derivative-free filter simulated annealing method for constrained continuous global optimization," *J. Global Optim.*, vol. 35, no. 4, pp. 521–549, 2006.
- [87] S. Cahon, N. Melab, and E. Talbi, "ParadisEO: A framework for the reusable design of parallel and distributed metaheuristics," *J. Heuristics*, vol. 10, no. 3, pp. 357–380, 2004.
- [88] A. LaTorre, S. Muelas, and J. Peña, "A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test," *Soft Comput. A Fusion Found. Methodol. Applicat.*, vol. 15, no. 11, pp. 1–13, Sep. 2010.
- [89] D. Sheskin, *The Handbook of Parametric and Nonparametric Statistical Procedures*. Boca Raton, FL: Chapman and Hall/CRC, 2000.



Francisco J. Rodriguez received the M.Sc. degree in computer sciences from the University of Granada, Granada, Spain, in 2005. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Artificial Intelligence, University of Granada.

His current research interests include the analysis of search algorithms for optimization problems, including evolutionary algorithms, hybrid metaheuristics, simulated annealing, iterated greedy algorithms, and greedy randomized search procedures.



Carlos García-Martínez received the M.Sc. and Ph.D. degrees in computer sciences from the University of Granada, Granada, Spain, in 2003 and 2008, respectively.

He is currently an Associate Professor with the Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain. His current research interests include mainly the analysis of search methods for optimization problems. Among his publications, it is worth mentioning his studies on evolutionary algorithms, ant colony optimization, and hybrid metaheuristics.

Dr. García-Martínez regularly acts as a reviewer for remarkable international journals and conferences.



Manuel Lozano received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1992 and 1996, respectively.

He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. His current research interests include evolutionary algorithms, metaheuristics (simulated annealing, iterated greedy, GRASP, and so on), hybrid metaheuristics, and combination of fuzzy logic and evolutionary algorithms.