



ORIGINAL ARTICLE

Bayesian network as an adaptive parameter setting approach for genetic algorithms

Guillaume Corriveau¹ · Raynald Guilbault¹ · Antoine Tahar¹ · Robert Sabourin²

Received: 24 March 2015 / Accepted: 10 February 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Parameter setting currently ranks among the most actively researched topics in the evolutionary algorithm (EA) community. This can be explained by the major impact EA parameters have on search performance. However, parameter setting has been shown to be both problem dependent and evolution dependent. Moreover, because parameters interact in complex ways, developing an efficient and beneficial parameter setting approach is not an easy feat, and no broadly recognized solution has emerged to date. In this paper, we borrow the notion of parameter adaptation with the objective of addressing the parameter setting dependencies mentioned above, using a strategy based on a Bayesian network. The adaptive framework is elaborated for a steady-state genetic algorithm (SSGA) to control nine parameters. To judge parameter state productivities, we consider the population's fitness improvement, as well as exploration/exploitation balance management. The performance of this proposal, a Bayesian network for genetic algorithm parameter adaptation (BNGA), is assessed based on the CEC'05 benchmark. BNGA is compared to a static parameter setting, a naïve approach, three common adaptive systems (PM, AP, and FAUC-RMAB), and two state-of-the-art EAs (CMA-ES and G-CMA-ES). Our results statistically demonstrate that the performance of BNGA is equivalent to that of FAUC-RMAB, CMA-ES, and G-CMA-ES, and overall is superior to that of all the other SSGA parameter setting

approaches. However, these results also reveal that all the approaches considered have great difficulty finding global optima in a multimodal problem set. This suggests a lack of complementarity and/or synergy among parameter states.

Keywords Bayesian network · Exploration/exploitation balance · Genetic algorithms · Parameter adaptation

1 Introduction

Metaheuristics inspired by nature, such as evolutionary algorithms (EAs), form a major class of modern optimization methods [1]. One of the reasons underlying their success is that they provide the flexibility needed to solve diverse engineering problems [2]. However, this flexibility comes at the cost of defining and setting multiple internal parameters, which represents a difficult task. This is because:

1. the parameters are problem dependent [3];
2. they often change as the process evolves [4];
3. the interactions among them can be complex [5].

The problem dependency is expressed by the no free lunch theorem, as individual parameter configurations represent different search path optimizers. The evolution dependency and interaction dependency issues are mostly acknowledged through empirical studies.

Parameter setting approaches first appeared on the scene with the emergence of the EA field, with the aim of providing guidelines to practitioners [6–8]. In the early 1990s, as it became clear that general recommendations were hardly useful [9], the emphasis shifted from guidelines to control systems and finely tuned methodologies. With a widely

✉ Guillaume Corriveau
guillaume.eng.corriveau@gmail.com

¹ Department of Mechanical Engineering, École de technologie supérieure, 1100 Notre-Dame Street West, Montreal H3C 1K3, Canada

² Department of Automated Manufacturing Engineering, École de technologie supérieure, 1100 Notre-Dame Street West, Montreal H3C 1K3, Canada

recognized and satisfactory solution yet to be found even after more than 35 years, parameter setting continues to be an important and active field of research [10]. Not only do parameter setting approaches help alleviate practitioners' efforts, but they also play a major role in the achievement of enhanced EA performance. In fact, EA parameters are responsible for providing a specific exploration/exploitation balance (EEB) over a given problem [11]. This balance dictates the search path to be followed, and as a result, parameter setting constitutes a stepping stone providing an optimization framework to practitioners that is both robust (high-quality solutions for a wide range of problems) and efficient (high-quality solutions delivered as quickly as possible).

The most comprehensive taxonomy of parameter setting approaches to date has been provided by Eiben, Hinterding, and Michalewicz [9]. They define four approaches: parameter tuning, deterministic control, adaptive control, and self-adaptive control. Parameter tuning methods are designed to fix parameters prior to the optimization process; alternatively, parameter control methods change parameters as the process evolves. Deterministic control methods are iteration-based formulations, which do not take feedback from the process into account, while adaptive control methods use feedback gathered from the search and somehow apply it to the selection of parameter states. Finally, self-adaptive control refers to techniques in which parameters are directly encoded in individuals with the aim of allowing evolution to identify the most promising parameter states. Obviously, each of these categories has its own advantages and drawbacks [9]. Based on the preceding description, it could be concluded that parameter tuning and deterministic control are not capable of characterizing the three parameter setting dependency issues listed above. In contrast, the parameter setting capability of self-adaptive control is based entirely on the individual's fitness. Nevertheless, these raw values are often not sufficient to characterize the search behaviour. For instance, population diversity remains crucial in guiding the search process [12]. Therefore, we should recognize that adaptive control, which is the only remaining option, represents the most attractive class to guide the search towards an optimal path.

Over the years, numerous adaptive parameter control approaches have been proposed. However, these have been restricted to a limited number of EA parameters. In fact, a

review of more than 100 papers covering parameter adaptation reveals that more than 88 % of the approaches examine the adaptation of at most two parameters, while only 3 % involve four parameters. To the authors' knowledge, no study has proposed the adaptation of more than four parameters. Furthermore, only a few of these approaches have the potential of handling parameter interactions, while most of them consider fixed interactions. This limits their adaptation effectiveness in terms of achieving an optimal search path, as parameter interactions may change when faced with different problems or during process evolution. This indicates that EA parameter adaptation is a rather complex task, and one that is yet to be fully resolved. This study proposes an adaptive control framework that enables the management of all three parameter setting dependencies.

The paper is organized as follows. Sect. 2 reviews the adaptive frameworks available in the literature; Sect. 3 presents the proposed adaptive control system; Sect. 4 describes the methodology introduced to compare the examined parameter setting approaches; Sect. 5 presents the results of this comparative study, and, finally, Sect. 6 provides a concluding discussion.

2 Review of adaptive parameter control strategies

Many aspects of an adaptive control strategy (listed below) need to be defined, while Fig. 1 illustrates the adaptive control process. Since each of these elements plays a specific role, the following subsections review them separately. Note that the present research focuses mainly on genetic algorithms (GA) due to their extensive use and the requirement of many parameters in their formulations [13].

- The type and states of the parameters involved.
- The feedback indicator used to evaluate the impact of the current state (j) of parameter i .
- The window interval (W) over which the adaptation is conducted;
- The credit assignment scheme required to convert feedback information into a suitable reward.
- The parameter selection rule used to update parameter states.

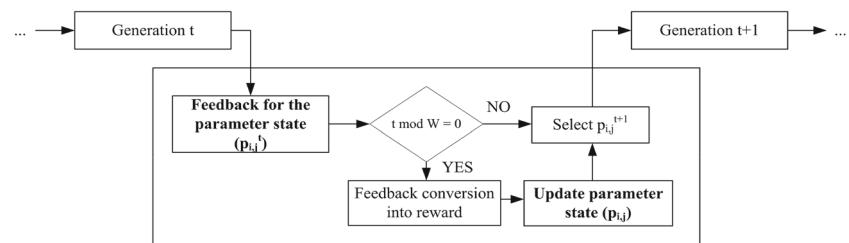


Fig. 1 General framework describing the adaptive process. Steps in ***boldtype*** refer to fundamental components



2.1 Parameters involved

In general, six parameters must be set to conduct a GA search: (1) the population size (N), (2) the selection mode, genetic operators through (3) crossover and (4) mutation types, as well as their respective activation probabilities, (5) p_c and (6) p_m . This ensemble becomes even larger if we consider multi-parent selection (μ) and/or multiple offspring creation per couple (λ). In addition, supporting the steady-state evolution model (SSGA) adds replacement and update plans to parameter selection. Throughout this paper, parameter setting is looked at in the broadest sense of the term, which includes both parameters and operators.

One of the strongest motivations for conducting parameter adaptation is to control the EEB throughout the search. Each parameter has its own impact on this fundamental search characteristic. For instance, increasing N favours the exploration of the landscape, whereas increasing selection pressure directs the search towards the exploitation of the most promising solutions. Depending on their formulation, any of the genetic operators may be seen as having an exploration or an exploitation function [11]. Adapting all the parameters therefore becomes important.

2.2 Feedback indicators

A feedback indicator may be defined as the evidence on which adaptation is based [14]. In other words, by assessing the productivity of the parameter states, it acts as the driving force for adaptation. Sending the wrong advice to the update mechanism may adversely affect the adaptation process [15], which makes feedback indicator selection truly fundamental for any adaptive framework.

Since most adaptive studies cover genetic operators and their activation probabilities, we must point out that many feedback formulations are based on the improvement achieved by the offspring ($f_{\text{offspring}}$) relative to a reference fitness value. This reference could be the best individual in the population (f_{best}) [16], the parents (f_{parent}) [15], or any population-based statistics [17, 18]. In almost all applications, no penalty is transferred to the associated parameter state when no improvement is registered from $f_{\text{offspring}}$.

Another widely applied approach consists in considering diversity as a feedback indicator. Indeed, diversity is a measure commonly used to express the EEB of a search process, and here, it can be formulated based on the location of the individuals in the search space (i.e. genotypic formulation) or based on their fitness distribution (i.e. phenotypic formulation). Various proposals have been developed regarding genotypic diversity measures (GDMs) [12] and phenotypic diversity measures (PDMs) [19]. It is important to note, however, that PDMs are inherently limited when they are used alone [20]. Consequently, feedback indicators using both

GDMs and PDMs have also been proposed, with the aim of simultaneously describing both the quantity and the quality of the population diversity [19]. Reference [21] considered up to three aspects: a GDM, a quality measure defined as the average fitness (f_{avg}) variation, and the mean execution time of the selected parameter state.

2.3 Credit assignment scheme

Credit assignment is used to convert a feedback indicator into a form supported by the adaptation mechanism (i.e. selection rule) and/or for aggregating multiple feedback indicators. The normalization of a feedback indicator can be regarded as a simple credit assignment scheme. In fact, normalization is helpful in reducing dependency related to feedback indicators built from raw values [22]. The concept of ranking has also been proposed as a way of eliminating the concern regarding raw values [22]. Reference [23] focuses on rare but substantial improvements based on the probability of producing exceptionally good solutions.

Regardless of the credit assignment scheme promoted, the frequency rate of the adaptation schedule must be defined. This is reflected by the window size (W), a hyper-parameter investigated in [15]. Although it was found that W is relatively robust overall, the published results suggest that W may be sensitive and problem dependent. Three streams are employed for conferring a reward to a selection rule: (1) instantaneously conferring the reward ($W = 1$), (2) averaging the rewards over W , or (3) conferring the extreme reward encountered along W .

2.4 Parameter selection rule

The parameter selection rule can be considered as the core of any adaptive system. It is responsible for automatically incorporating past performance into current parameter selection, and numerous proposals for this rule have been presented over the years. For the sake of clarity, they are grouped below into six families.

2.4.1 Heuristic rule

This family encompasses various kinds of selection rules, all specifically designed for a given parameter. Because of this limitation, only the most famous of them is presented here: Rechenberg's "1/5 success rule", which was proposed within the evolution strategy (ES) paradigm. This rule is defined as follows [24]: "The ratio of successful mutations to all mutations should be 1/5. If it is greater than 1/5, increase the standard deviation, if it is smaller, decrease the standard deviation."

2.4.2 Fuzzy logic controller

Fuzzy logic controller (FLC) allows the definition of fuzzy IF–THEN rules. This approach is useful when knowledge is vague, incomplete, or ill-structured [19]. Well-defined fuzzy sets are, however, quite difficult to form, and membership functions can be challenging to value. So, in general, FLCs rely on the knowledge and experience of experts [19].

2.4.3 Probability matching

The probability matching (PM) method was proposed by Goldberg [25] as a reasonable adaptive strategy for an environment characterized by uncertainty. PM is based on translating reward values into a corresponding selection probability, and is formalized in the following mechanism:

$$\hat{q}_j^{t+1} = (1 - \alpha)\hat{q}_j^t + \alpha \cdot r_j^t, \quad (1)$$

$$p_{i,j}^{t+1} = p_{i,\min} + (1 - J \cdot p_{i,\min}) \frac{\hat{q}_j^{t+1}}{\sum_{j=1}^J \hat{q}_j^{t+1}}. \quad (2)$$

At a generation $t+1$, state j of parameter i is selected following the probability stored in $p_{i,j}^{t+1}$. This selection probability is built from the past empirical quality estimate of state j (\hat{q}_j^t) and the current reward (r_j^t) provided by the credit assignment scheme. These two components are connected through an additive relaxation mechanism described by (1) and controlled by an adaptation rate parameter ($0 < \alpha \leq 1$). This parameter allows the system memory to be reduced, as past knowledge is increasingly forgotten with the rise of α . To ensure that no parameter state is lost in the process, a minimum probability ($p_{i,\min}$) is granted to all states (J) of parameter i . This last feature is essential to cope with the evolution dependencies. The PM selection rule has been used extensively [15–18, 23].

2.4.4 Adaptive pursuit

Adaptive pursuit (AP) was introduced as an adaptive selection rule by Thierens [26]. It is based on a decision theory approach in which a winner-takes-all strategy is adopted, as the emphasis is placed on the parameter state with the highest reward. AP uses the same additive relaxation mechanism as PM (1). However, the updated probability mechanism is given by:

$$p_{i,j}^{t+1} = \begin{cases} p_{i,j}^t + \beta(p_{i,\max} - p_{i,j}^t) & \text{if } j = j^* \\ p_{i,j}^t + \beta(p_{i,\min} - p_{i,j}^t) & \forall j \neq j^* \end{cases}, \quad (3)$$

where $j^* = \arg \max_{j=1, \dots, J} \{\hat{q}_j^{t+1}\}$ and $p_{i,\max} = 1 - (J - 1)p_{i,\min}$. The learning rate ($\beta \in [0, 1]$) is used to control the dominance of the best parameter state with respect to the concurrent states. It has been shown that AP outperforms PM [22, 26], at least over artificial scenarios.

2.4.5 Multi-armed bandit

The multi-armed bandit (MAB) approach was introduced as an EA adaptive selection rule by DaCosta et al. [27]. It allows the management of the exploration/exploitation balance among parameter states. The approach does not allocate any probability to an individual parameter state; rather, the allocation decision is made by selecting the state that provides the highest expectation. The general form is given by:

$$p_i^{t+1} = \operatorname{argmax}_{j=1, \dots, J} \left(\hat{q}_j^{t+1} + C \cdot \sqrt{\frac{2 \cdot \log \left(\sum_{j=1}^J n_j \right)}{n_j}} \right), \quad (4)$$

where n_j represents the number of times state j was selected from the latest periods. Within this formulation, the first term describes the exploitation aspect, while the second term is intended to promote exploration. The balance between these two aspects is controlled through the scaling factor C .

The first MAB implementation was dynamic MAB (DMAB). However, because of certain issues related to DMAB hyper-parameter setting, MAB variants, such as the sliding MAB (SLMAB) and the rank-based MAB (RMAB) [22], were developed. The main difference between RMAB and SLMAB is the \hat{q}_j^{t+1} definition. Instead of using raw values from the credit assignment scheme, RMAB ranks them through the area under the curves (AUC) concept or through the sum of the ranks (SR). Variants of these two ranking processes (FAUC and FSR) have also been introduced. Compared to AUC and SR, these latter approaches are shown to be invariant to monotonous fitness transformation. Furthermore, a decay factor (D) is integrated into RMAB to allow greater influence to top-ranked parameter states.

In all the MAB forms, (4) is applied only when at least one trial is allocated to each parameter state. Otherwise, the unselected states are chosen randomly. FAUC–RMAB has been shown to be the best MAB variant for adapting genetic operators [22], and additionally, has also shown better performance than PM or AP. Finally, it is worth noting that none of the MAB variants has succeeded in solving most of the multimodal functions considered. To explain this, Fialho [22] concluded: “[...] the maintenance of some level of diversity in the population should also be accounted somehow for the rewarding of operator applications [...]”.



2.4.6 Covariance matrix adaptation

The covariance matrix adaptation (CMA) approach was developed within the ES framework (CMA-ES) by Hansen and Ostermeier [28]. The purpose of this strategy is to maximize the search progress rate by reproducing the successful evolution path through the adaptation of mutation (i.e. step size and direction). The underlying assumption of CMA-ES is that the mutation step size (σ) must be adapted as quickly as possible, whereas the covariance matrix (\mathbf{C}) of the mutation distribution should be modified more slowly. This is reflected in the use of an additive relaxation mechanism, as defined in (1), for \mathbf{C} . σ is defined globally within the population, but applied locally, as it is weighted with respect to the principal axis identified by \mathbf{C} . This characteristic gives CMA-ES an invariant property with respect to the rotational and linear transformation of the search space. CMA-ES was introduced to improve the local search performance of ES, but it was shown in [29] that increasing N improves CMA-ES performance in multimodal problems. Based on this observation, G-CMA-ES was proposed in [30] with a restart feature that doubles the population size each time it is triggered. This could be seen as a second adaptive mechanism, since the restart decision is made based on five criteria characterizing search performance. To date, G-CMA-ES has been considered as the state-of-the-art EA [31].

2.5 Discussion

Regarding the three parameter setting dependencies, it is clear that all the adaptive control proposals take into account the problem and evolution dependencies by default. However, not every method can manage a parameter interaction dependency. In fact, among the previously defined families, only the heuristic rule and FLC are able to consider this aspect, as the other families are parameter-independent approaches. Nevertheless, methods relying on user-defined adaptive formulations are prone to encountering generalization issues with parameter interaction owing to the problem dependency issue.

Furthermore, almost all the adaptive methods rely on some hyper-parameters. This would be justifiable if it could be demonstrated that these hyper-parameters are less sensitive than the controlled parameters of EAs. However, if the ratio of hyper-parameters to controlled parameters (here referred to as the H/C ratio) is greater than 1, it is easy to lose track of the intended adaptive goal. In fact, the H/C ratio is 3 for SLMAB, 4 for PM and RMAB, and 5 for AP and DMAB, when hyper-parameters related to the adaptation schedule (window size and type of reward conferred) are included.

3 Parameter adaptation through Bayesian network

We propose the use of a Bayesian network (BN) as a GA parameter adaptation scheme (referred to as BNGA) to tackle the parameter setting dependencies more effectively. The BN is a coherent probabilistic framework drawn from the machine learning field; it is in fact a causal network based on Bayes' rule:

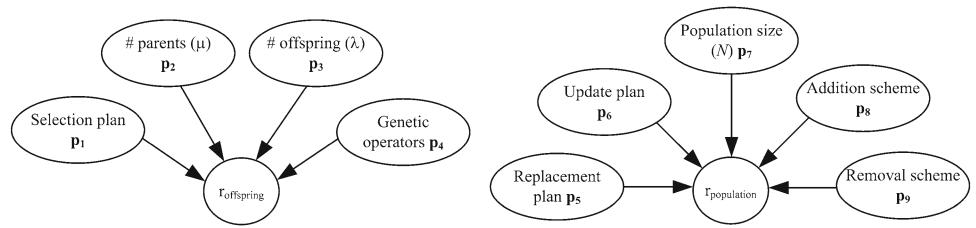
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (5)$$

Within this formulation, $P(A)$ represents the prior probability of event A , whereas $P(A|B)$ describes the posterior probability of event A , knowing event B . $P(B|A)$ represents a conditional probability or the likelihood of A , given B . The marginal probability of B ($P(B)$) is taken into account for normalization purposes. It guarantees that final probabilities will respect the law of probability ($\sum_j P(a_j|B) = 1.0$). The consistency of Bayes' rule is easily demonstrated for independent events A and B , as the joint distribution representing the numerator in (5) becomes: $P(A, B) = P(B|A)P(A) = P(A)P(B)$. Thus, a BN allows for the belief update on a parameter state based on new evidence from the search process. However, as the number of parameters increases, it becomes difficult to perform inferences directly from Bayes' rule. To address this problem, BN uses a graphical model, which can efficiently represent the relationships among the parameters. Inference from it is conducted with algorithms developed to take advantage of the graphical representation.

Bayesian network has been applied to various domains, particularly in EA, where BN is used in the estimation of distribution algorithm (EDA) search paradigms to carry out probability distribution updates of landscape variables [32]. BN has also been used to tune GA parameters based on the number of evaluations required to achieve suitable solutions [33]. It has further been extended, with case-based reasoning (CBR), to cope with the inherent limitations of tuning [34]. To the authors' knowledge, BN has never been used as an adaptive parameter control system.

The construction of a BN for a particular application usually follows two steps. First, learning is conducted to define the best structure for describing the relationship among variables or parameters in the explored context. Then, network data or conditional probability tables (CPT) are learned from that BN structure, with the aim of quantifying the strength of those relationships. Thus, the BN framework tackles parameter interaction directly through the definition of the graphical model. The problem and evolution dependencies are settled with the help of the CPTs.

Fig. 2 Graphical model of BNGA



3.1 Graphical model of BNGA

The BN structure for a specific application can be learned using scoring metrics or search algorithms. However, to reduce computational costs, a graphical model is defined a priori in this study. Consequently, BNGA is based on the simplest BN category, where the structure is known and the data are all observable. Figure 2 illustrates this graphical model. The structure is developed for a real-coded steady-state GA (SSGA). The structure is decoupled into two-parameter sets to allow the parameters to act on two different canvases. In fact, as can be observed in Sect. 2.2, two kinds of evidence are used in the literature to assess parameter state productivities: (1) solution improvement and (2) population diversity. As they have a direct impact on search performance, both of these indicators should be used to assist in adaptation. Therefore, in BNGA, parameters related to the creation of new solutions ($p_i, i \in 1 \text{ to } 4$) are judged by their ability to improve solutions. In contrast, parameters involved in population governance ($p_i, i \in 5 \text{ to } 9$) are judged by their ability to manage the EEB through population diversity. Obviously, parameters from the former set have an impact on the EEB. However, the benefit of the steady-state model is that replacement and update plans constitute a gateway for deciding who will take part in the population.

Regarding the parameters involved, no activation probabilities (p_c or p_m) are considered here, as offspring are automatically created through a genetic operator. Furthermore, p_8 and p_9 are added to control the dynamics of the population size.

In addition to BNGA being able to encompass all the SSGA parameters, the simplicity of its structures provides three advantages [35]. First, the V converging structures, corresponding to singly connected polytrees, allow closed-form inferences to be performed through a local updating scheme (the message passing algorithm [35]). The complexity of this model is linear with respect to the number of variable nodes [36,37], whereas in general, BN inference is NP-hard [38]. This is an essential characteristic of BNGA, as it allows the minimization of the run-time impact of the adaptive system on the GA process. Second, it is possible to gather data incrementally from this two-level structure and refine our knowledge on parameter states. This characteristic constitutes the core of the present proposal for building

CPTs. Section 3.3 below completes the description. Finally, the decoupling of the parameters into two sets enables the use of belief propagation without any approximation technique, such as noisy OR and AND gates. This is because of the small number of parent nodes in each structure. Thus, BNGA is aligned with the statement by Druzdzel and van der Gaag [39]: “Building a probabilistic network requires a careful trade-off between the desire for a large and rich model on the one hand and the costs of construction, maintenance, and inference on the other hand.”

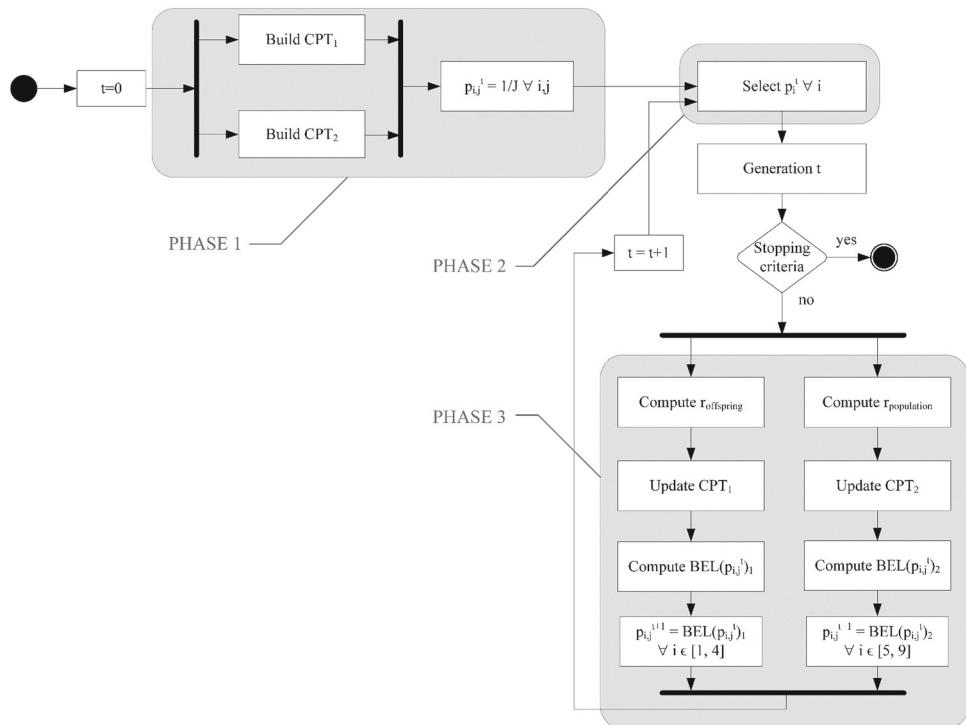
The process of BNGA may be summarized in the three phases illustrated in Fig. 3. The first phase is related to the initialization through the definition of the CPTs and the prior probabilities of each parameter. Since no initial knowledge on the problem at hand is available, both CPTs and prior probabilities are assumed to be uniform distributions ($p_{i,j} = 1/J$). In the second phase, the parameter state for the next generation is selected. This selection is based on the established prior probabilities. Once the new generation is settled by the GA process, reward computations and CPTs updates for the parameters sets associated with new solution creation ($r_{\text{offspring}}$) and to population governance ($r_{\text{population}}$) are carried out in the third phase of BNGA. Thereafter, the probabilities of the parameters ($\text{BEL}(p_i^t)$) are computed. The last step of this third phase is to update the prior probabilities for the upcoming generation ($p_{i,j}^{t+1}$); these probabilities are set equal to the last probabilities computed at the current generation. Phases 2 and 3 are repeated until the GA stop criteria are met.

To sum up, BNGA allows control of all the SSGA parameters, while taking into account the three dependencies:

1. Problem dependency—Initial CPTs and prior probabilities defined as uniform distributions.
2. Evolution dependency— $\text{BEL}(p_i^t)$ used as prior probabilities for the next generation.
3. Parameters interaction dependency—parameters p_m to p_n are conditionally dependent on r .

Furthermore, no additive relaxation or fading mechanism is needed by BNGA, as the memory of past knowledge is automatically managed through the modification of the prior probabilities at each generation. Finally, BNGA does not require any adaptation schedule, since it takes advantage of new evidence as soon as it appears.

Fig. 3 Flowchart of the BNGA process



3.2 Credit assignment schemes

Rewards for p_1 to p_4 ($r_{\text{offspring}}$) are based on the fitness improvement of the offspring, while rewards for p_5 to p_9 ($r_{\text{population}}$) reflect the search EEB by means of population diversity measures.

Regarding $r_{\text{offspring}}$, preliminary experiments not included in this article showed that comparing average fitness of the offspring ($\bar{f}_{\text{offspring}}$) to the average fitness of the parents (\bar{f}_{parent}) provides the most suitable results. Therefore, for minimization problems, the reward over the former parameter set is given by:

$$r_{\text{offspring}} = 0.5 + 0.5 * \frac{(\bar{f}_{\text{parent}} - \bar{f}_{\text{offspring}})}{|\bar{f}_{\text{parent}}|} \quad (6)$$

$$P(r_{\text{offspring}} | p_1, p_2, p_3, p_4) = \begin{cases} r_{\text{offspring}} & \text{if } p_{\min} \leq r_{\text{offspring}} \leq 1 - p_{\min}; \\ p_{\min} & \text{elseif } r_{\text{offspring}} < p_{\min}; \\ 1 - p_{\min} & \text{otherwise.} \end{cases} \quad (7)$$

The 0.5 multiplication in (6) allows a better representation of the reward response between -0.5 and 0.5. Therefore, parameter states that are not able to improve parent solutions are penalized. In other words, a bad parameter choice also provides relevant knowledge that must not be ranked at the same level as an unused parameter state. This contradicts the

mainstream view presented in Sect. 2.2. Finally, conditions are proposed to ensure that $r_{\text{offspring}}$ remains in the $[p_{\min}, 1 - p_{\min}]$ range.

The $r_{\text{population}}$ factor relies both on a genotypic and on a phenotypic measure to describe the quantity and quality of the population diversity simultaneously. It has been shown that the EEB orthogonal framework is a generalization of the exploration/exploitation opposing force concept [12]. As such, the GDM is responsible for characterizing the exploration axis, while the exploitation axis is represented by a phenotypic convergence measure (PCM). Note that the PCM is the inverse of the normalized PDM. A PCM is used instead of a PDM to comply with the *useful diversity* concept proposed in [40], which favours the maintenance of appropriate diversity, which potentially brings in good individuals. From this standpoint, both GDM and PCM must be maximized. Equations (8) and (9) formulate the measures employed in this paper for GDM and PCM evaluation. Even though they do not guarantee a perfect evaluation in all cases, the descriptors were compared to other available formulations and showed significant performance increases globally.

$$\text{GDM} = \frac{\sum_{i=1}^{N-1} \ln \left(1 + \min_{j \in [i+1, N]} \frac{1}{n} \sqrt{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2} \right)}{\text{NMDF}}, \quad (8)$$

$$\text{PCM} = 1 - \frac{\sum_{i=1}^{N-1} \ln(1 + |f_i - f_{i+1}|)}{\text{VMD}}. \quad (9)$$

The subscripts used in these formulations are related to the individuals in the population. Therefore, $x_{i,k}$ and $x_{j,k}$ refer to gene k of individuals i and j . The GDM formulation is developed for a real-coded representation, where n corresponds to the dimensionality of the landscape. In the PCM formulation, f_i and f_{i+1} correspond to the neighbour fitness derived from the sorted fitness distribution. Finally, NMDF and VMD are normalization techniques. The former performs normalization with the maximum diversity achieved so far and simply yields the maximum value attained by the numerator since the beginning of the search process. In contrast, the latter performs normalization with virtual maximum diversity. VMD is computed through the diversity part, or numerator, of the PCM formula using a virtual distribution of N samples uniformly distributed between f_{worst} and f_{best} . Therefore, VMD needs to be updated when the population size and/or absolute fitness range change(s).

From there, we need to determine how these two measures can be combined to produce $r_{\text{population}}$. A similar framework was applied in [21], where it was proposed that genetic operators be rewarded based on their ability to bring the EEB closer to 45° (equal exploration and exploitation levels). A commonly adopted EEB management scheme involves promoting full exploration at the beginning of the evolution process ($\text{EEB} \rightarrow 90^\circ$) and moving this balance towards full exploitation as the process evolves ($\text{EEB} \rightarrow 0^\circ$) [9]. However, these EEB management schemes are biased and have not been proven to provide an optimal search path over any problem. This issue was summarized in [41] as follows: “The type and amount of diversity required at different evolutionary times remains rather unclear.” Consequently, to make EEB management as flexible as possible, the proposed strategy encourages parameter states that maximize both exploration and exploitation (i.e. GDM $\rightarrow 1$ and PCM $\rightarrow 1$). Equation 10 formulates the operation:

$$r_{\text{population}} = \frac{\sqrt{\text{GDM}^2 + \text{PCM}^2}}{\sqrt{2}} \quad (10)$$

$$P(r_{\text{population}} | p_5, p_6, p_7, p_8, p_9) = \begin{cases} r_{\text{population}} & \text{if } (p_{\min} \leq r_{\text{population}} \leq 1 - p_{\min}) \wedge (\text{GDM} > p_{\min} \wedge \text{PCM} < 1 - p_{\min}); \\ p_{\min} & \text{elseif } (r_{\text{population}} < p_{\min}) \vee (\text{GDM} < p_{\min} \wedge \text{PCM} > 1 - p_{\min}); \\ 1 - p_{\min} & \text{otherwise.} \end{cases} \quad (11)$$

Equation 11 adds a condition by penalizing parameter states that produce convergence (i.e. $\text{GDM} < p_{\min}$ and $\text{PCM} > 1 - p_{\min}$). This condition allows moving away from a premature convergence state without relying on a restart feature.

Finally, as observed in (7) and (11), a minimum probability (p_{\min}) constrains the rewards. The meaning of p_{\min} in this

case is similar to that involved in the PM and AP adaptive systems; p_{\min} ensures that parameter state ($p_{i,j}$) is not lost during the search process. However, the role of p_{\min} is slightly different in BNGA, in that certainty states (i.e. $p_{i,j} = 0.0$ or $p_{i,j} = 1.0$) are prohibited by avoiding the certainty of conditional probabilities. However, $p_{i,j}$ can approach these values without restriction. In addition, this hyper-parameter serves miscellaneous functions, depending on each parameter set: for $r_{\text{offspring}}$, p_{\min} allows the reward value to be bounded in a probabilistic range, while in contrast, for $r_{\text{population}}$, p_{\min} is involved in the definition of a hazardous area and penalizes parameter states that navigate into it.

3.3 Conditional probability table (CPT)

In BNGA, CPTs are built from a Bayesian type estimation. At each generation, a uniform probability is assigned to all the parameter configurations in the CPTs ($P(r|p_i) = 0.5$), and the reward ($r_{\text{offspring}}$ or $r_{\text{population}}$) is subsequently used to update the parameter configuration selected ($j = s$) in the CPT ($P(r|p_{i,s}) = r$). This CPT building scheme ensures that knowledge gathered through the reward is considered only once, as its impact is transferred to the parameter state probabilities by setting the prior probabilities of the next generation ($t+1$) equal to the posterior probabilities of the previous generation (t). This approach is valid because the CPTs are built from scratch at each generation.

Finally, the size of the CPTs increases exponentially with the number of parameters involved. Consequently, a mechanism is added to increase the impact of the single configuration reward ($P(r|p_{i,s})$) in the CPTs. This is achieved by allocating a probability equal to $1 - P(r|p_{i,s})$ to all configurations that do not involve parameter states included in the selected parameter configuration (i.e. $\forall P(r|p_{i,j}) = 1 - P(r|p_{i,s})$, where $j \neq s$). The assumption underlying this reinforcement mechanism is that if a parameter state performs well over a given generation, other choices become undesirable, and vice-versa. In fact, this reinforcement mechanism increases the rate of adaptation.

3.4 Posterior probabilities (BEL)

To ensure the tractability of the adaptive system, the posterior probabilities are computed at each generation according to a closed-form inference strategy, which is the message passing algorithm [35] given by:

BEL Algorithm: local updating scheme(param,prior,CPT)

Input: List of the parameters with their states, prior probabilities of the parameters states, conditional probability table

Ouput: Posterior probabilities of the parameters states

```

1: for  $i = m, \dots, n$  do
2:   SUM_PROD(1:J)  $\leftarrow 0$ 
3:   for  $j = 1, \dots, J$  do
4:     iSTATE  $\leftarrow \text{param}(i,j)$ 
5:     for  $k = 1, \dots, \text{size}(\text{CPT})$  do
6:       if  $\text{CPT}(k,i) = \text{iSTATE}$ 
7:         /*store the conditional probabilities
8:          *corresponding to the reward  $r$ 
9:        */
10:      PROD  $\leftarrow \text{CPT}(k,J+1)$ 
11:      for  $u = m, \dots, n$  do
12:        if  $u \neq i$ 
13:          uSTATE  $\leftarrow \text{CPT}(k,u)$ 
14:          PROD  $\leftarrow \text{PROD} * \text{prior}(u, \text{uSTATE})$ 
15:        end if
16:      end for
17:      SUM_PROD(j)  $\leftarrow \text{SUM\_PROD}(j) + \text{PROD}$ 
18:    end for
19:  end for
20:  /*bottom-up propagation (message passed by the child)
21:   * “.” represent element by element division
22:   * normalization
23:   */
24:  mlambda  $\leftarrow \text{SUM\_PROD} / \sum_{j=1}^J \text{SUM\_PROD}$ 
25:  /*top-down propagation (message passed by the parent)
26:   * directly equal to prior, since parents node have no parent
27:   */
28:  mpi  $\leftarrow \text{prior}(i,1:J)$ 
29:  /*posterior probabilities computation
30:   * “.” represent element by element multiplication
31:   */
32:  BEL_tmp(i,1:J)  $\leftarrow \text{mlambda} * \text{mpi}$ 
33:  /*normalization
34:   * “.” represent element by element division
35:   */
36:  BEL(i,1:J)  $\leftarrow \text{BEL\_tmp} / \sum_{j=1}^J \text{BEL\_tmp}$ 
37: end for
38: Return (BEL)

```

Fig. 4 Pseudocode of the inference algorithm used by BNGA

$$\lambda_r(p_i) = \alpha \sum_{p_k: k \neq i} P(r | p_m, \dots, p_n) \prod_{k \neq i} \pi_r(p_k), \quad (12)$$

$$\text{BEL}(p_i) = \alpha \cdot \lambda_r(p_i) \cdot \pi_r(p_i), \quad (13)$$

where $\lambda_r(p_i)$ is the message passed by the child node ($r_{\text{offspring}}$ or $r_{\text{population}}$) to the parent nodes, and $\pi_r(p_i)$ is the message passed by the parent nodes p_i to the child node. Since nodes p_i do not have any parent, $\pi_r(p_i)$ is exactly equal to their prior probabilities. α is a normalization factor based on the summation of all p_i states. In (12), $m = 1$ and $n = 4$ for the structure converging towards $r_{\text{offspring}}$, and $m = 5$ and $n = 9$ for the structure involving $r_{\text{population}}$. Figure 4 presents the pseudocode summarizing the inference algorithm.

4 Comparative study

We conducted a two-step comparative study to assess the performance of BNGA. First, we evaluated BNGA by means of an SSGA, looking at static parameter setting, a naïve adaptive method, PM, AP, and FAUC-RMAB. For static parameter setting, the parameters were defined from the most promising states identified by BNGA for each problem considered. For the naïve approach, the parameter states were defined with uniform probabilities throughout the course of the evolution, while for the other approaches, a hyper-parameter study was performed prior to the comparison to determine the impact of these parameters and selecting the best configuration for them.

We then compared the best parameter setting approach found in the first step to CMA-ES and G-CMA-ES in their default settings.¹ We propose this two-step approach because CMA-ES and G-CMA-ES follow different EA paradigms with different genetic operators and may thus overshadow the objectives of the analysis over the SSGA framework.

4.1 Methodology

The comparison included 25 functions (F1–F25) defined in the CEC'05 benchmark [42]. However, since this comparison represented a first validation phase aimed mainly at demonstrating the inherent ability of BNGA, while rigorously adopting the methodology provided by the CEC'05 benchmark, the following procedure remains limited to ten-dimensional experiments ($n = 10$). Even though this restriction already offers high complexity levels, a complete description of the BNGA potential would later require a second experimentation phase with n greater than 10. Twenty-five runs were conducted for each problem. The search process stopped if the optimizer located the global optimum within the prescribed tolerance of 10^{-8} or if a 100,000-function evaluation limit was reached.

4.2 Parameter states involved

The following summarizes the SSGA process: λ offspring are generated at each generation using a specific crossover operator over the set of μ selected parents; following a chosen replacement plan, r individuals are removed from the population to make room for the new individuals; subsequently, an update plan is used to reinsert individuals from the temporary pool into the next generation.

Table 1 summarizes the parameters considered in this study. In terms of the selection plan, the analysis includes five alternative states: (1) the best individual is selected as

¹ CMA-ES version 3.51.beta was used to conduct this analysis. It can be accessed at <http://www.lri.fr/~hansen/cmaes.m>.

Table 1 Parameters involved and their potential states

State j	Parameter i	1 Selection plan	2 Number of parents (μ)	3 Number of offspring (λ)	4 Genetic operator	5 Replacement plan (r)	6 Update plan	7 Population size (N)	8 Addition scheme	9 Removal scheme
1	Best + random	2 ^a		2	PCX	Parents	Best	50	Random	Worst
2	Group-10			5	UNDX	1 random	BT1	100	Child + random	Tournament-2
3	Random			10	UX	2 random	BT2	300		Tournament-5
4	Tournament-2				BLX-0.5	5 random		500		Tournament-10
5	FUSS				PNX					
$p_{i,j}$		0.2	1.0	0.33	0.2	0.25	0.33	0.25	0.5	0.25

^aWithin this comparative study, μ is kept at its minimal value: 2 with $p_{4,j}$, where $j \in \{3, 4, 5\}$, and 3 with $p_{4,j}$, where $j \in \{1, 2\}$

the parent, and the remaining individuals are chosen randomly; (2) one individual is selected randomly from among the best individuals in ten groups in the fitness range, and the remaining parents ($\mu - 1$) are picked at random from the population [43]; (3) parents are randomly selected from the population; (4) parents are selected from a binary tournament; and (5) parents are selected using the fitness uniform selection scheme (FUSS) [44]. For the genetic operator, only crossovers are considered here, as they all have the ability to introduce new genetic material. Again, five potential states are taken into account: (1) a parent-centric crossover (PCX) with $\sigma_\eta = \sigma_\zeta = 0.1$ [45]; (2) a unimodal normal distribution crossover (UNDX) with $\sigma_\xi = 1/\sqrt{\mu - 2}$ and $\sigma_\eta = 0.35/\sqrt{n - \mu - 2}$ [46]; (3) a uniform crossover (UX) with the addition of normally distributed noise $N(0, 0.01)$; (4) a blended crossover (BLX-0.5); and (5) a parent-centric normal crossover (PNX) with $\eta = 1.0$ [47]. The analysis also provides four choices for the replacement plan: (1) parents are added to the temporary pool, and (2)–(4) $x \in \{1, 2, 5\}$ individuals are randomly selected from the population. For the update plan, three strategies are proposed: (1) the best individuals are reinserted into the population; (2) the best individual is selected and the remaining individuals are picked from binary tournaments (BT1); and (3) the same as BT1, except that a diversity control mechanism is considered, since the selected individuals replace the nearest genotypic individual already in the temporary pool (BT2). Finally, addition and removal schemes are required to manage the population size dynamics. Two addition methods are considered when N increases: (1) randomly created individuals are added to the population; and (2) all the non-inserted offspring of the current generation are added to the population, and the remaining required individuals are randomly created. Four removal methods are considered when N decreases: (1) the worst individuals are removed from the population; (2)–(4) a tournament is made up of $x \in \{2, 5, 10\}$ competitors, and the worst is removed from the population.

All the SSGA adaptive control systems (naïve, PM, AP, FAUC-RMAB, and BNGA) integrate these parameter states.

At the beginning of each run, the probability of application of each parameter state ($p_{i,j}$) is uniformly set, as presented in the last row of Table 1. The adaptive controls of those systems, except for the naïve approach, are based on the credit assignment schemes described in Sect. 3.2. For PM, AP, and FAUC-RMAB, the parameters are controlled independently, since these systems ignore the parameter interactions. In contrast, as depicted in Fig. 2, BNGA completely integrates the parameter interaction.

4.3 Hyper-parameter sensitivity analysis

To fairly compare the adaptive systems, a sensitivity analysis is conducted over the hyper-parameter setting as a prerequisite. Table 2 summarizes the hyper-parameter values considered for each adaptive system. For decoupled parameter control systems (PM, AP, FAUC-RMAB), the same hyper-parameters are used for each parameter, even though different optimal settings are likely. This choice maintains a suitable H/C ratio. Regarding p_{min} , the 1/2J option comes from the Thierens' proposal, which seeks at most, the best parameter state applied in half the time [26].

The sensitivity analysis includes five repetitions, from F1 to F10, by considering all the hyper-parameter combinations. Since the goal is to find robust hyper-parameters over various problems, as well as configurations that provide high-quality solutions, we suggest a multi-objective approach. Both objectives are evaluated by the following equation:

$$\text{obj} = \sum_{j=1}^{10} 1 - \frac{\min_{k \in [1, TC]} f_{k,j}}{f_{i,j}}. \quad (14)$$

For the first objective, f represents the mean best fitness error (MBFE), whereas for the second, f corresponds to the best fitness error (BFE). Both are defined from all the repetitions over the evaluated problem j . Subscript i stands for the studied hyper-parameter configuration, which ranges from 1 to the total number of combinations (TC). Using this normal-

Table 2 Hyper-parameter values considered for the sensitivity analysis

Adaptive control system	H/C ratio	Hyper-parameter	p_{\min}	α	β	C	D	Reward type	Window size (W)	Total combination (TC)
PM	0.44	{1/(2D), 0.01}	{0.1, 0.3, 0.5, 0.7, 0.9}					{0. instantaneous, 1. mean, 2. argmax}	{10, 100, 250}	70
AP	0.56	{1/(2D), 0.01}	{0.1, 0.3, 0.5, 0.7, 0.9}	{0.1, 0.3, 0.5, 0.7, 0.9}				{0. instantaneous, 1. mean, 2. argmax}	{10, 100, 250}	350
FAUC-RMAB	0.33					{0.01, 0.1, 0.5, 1, 5, 10}	{0.25, 0.4, 0.5, 0.75, 0.9, 1}			8
BNGA	0.11					{0.001, 0.005, 0.01, 0.03, 0.05, 0.1, 0.15, 0.2}				

ized formulation, responses can be aggregated, regardless of the complexity of the test functions. Consequently, each hyper-parameter configuration is assigned a result ranging from 0 to 10.

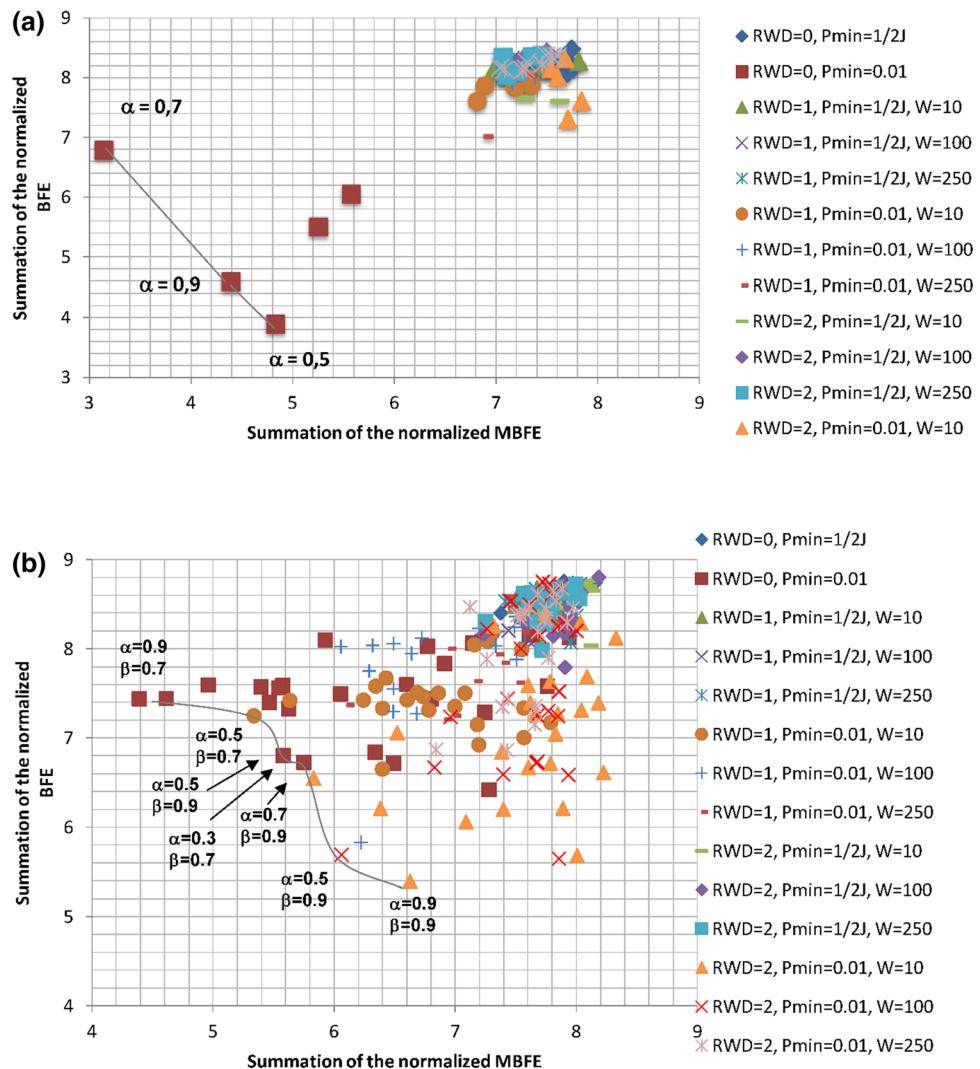
Figures 5 and 6 present the results for the four adaptive systems. For PM, the configurations with instantaneous reward application and $p_{\min} = 0.01$ stand out compared to all hyper-parameter choices. Among them, three are non-dominated configurations ($\alpha = \{0.5, 0.7, 0.9\}$). To perform the comparative study, PM with $p_{\min} = 0.01$, $\alpha = 0.9$, and RWD = instantaneous (MBFE = 4.4, BFE = 4.6) is selected. For AP, no hyper-parameter family emerges. However, seven configurations present non-dominated behaviours. Among them, AP with $p_{\min} = 0.01$, $\alpha = 0.3$, $\beta = 0.7$, and RWD=instantaneous (MBFE = 5.8, BFE = 6.7) is favoured. For FAUC-RMAB, favoured configurations suggest large window size, but only four configurations are found to be non-dominated. Among them, FAUC-RMAB with $C = 0.5$, $D = 0.5$, and $W = 250$ (MBFE = 5.3, BFE = 5.9), is favoured. Finally, the performance of BNGA shows an improvement with a decreasing p_{\min} value. Based on this result, and because it is one of the non-dominated configurations, and it better reintroduces configurations that had almost been forgotten, BNGA with $p_{\min} = 0.01$ (MBFE = 1.9, BFE = 3.5) is adopted.

5 Results

To analyze the results of the parameter setting approaches, the statistical procedure proposed by García et al. [48] is followed, according to which Friedman's non-parametric test is used to reveal any significant performance differences from at least one of the approaches. Then, a post hoc test, following Hochberg's procedure, is applied to identify any difference between each parameter setting approach and a control algorithm. This control algorithm is represented by the best ranked approach over the sample considered. For each approach, the sample is built from the MBFE characteristics over a predefined group of functions. As proposed in [48], three groups are examined: F1–F14, F15–F25, and F1–F25. This statistical procedure is also independently applied to each test function, where BFEs obtained from all the repetitions serve as samples.

Table 3 presents the format used to display the results of each parameter setting approach over each individual test function. The MBFE with respect to the global optimum is recorded with its standard deviation (STD) and the best fitness error (BFE) achieved over the 25 repetitions. Furthermore, the success rate (SR), as well as the success performance (SP), is also registered. The former are defined as the percentage of runs performing better than the prescribed tolerance, whereas the latter describes the mean function

Fig. 5 Results of the hyper-parameter sensitivity study: **a** PM. **b** AP



evaluation number required by the successful runs divided by SR. SP is intended to compare, on the same basis, algorithms with low probabilities of success but high convergence rates, and vice-versa. The last entry corresponds to the statistical test p values. This information is used for both individual test functions and groups of functions. Shaded p values indicate Friedman's test results and are located in the control algorithm column. The other p values represent Hochberg's post hoc test. An underlined p value indicates that the performance of a given approach is worse than that of the control algorithm, following a 0.05 significance level. Finally, bold entries designate the best values achieved for a given test function.

5.1 Results of the SSGA parameter setting approaches

Table 4 compares the results of the six parameter setting approaches within the SSGA framework. Overall, the statistical results show that the static parameter setting approach is outperformed by other approaches over 44 % of the test

functions (number of underlined p value/number of functions). This value increases to 48, 64, and 68 % for the naïve approach, PM, and AP, respectively. In terms of the FAUC-RMAB, poorer performances are observed over 40 % of the test functions. In this comparison, BNGA comes out as the most robust approach, since compared to the other parameter setting approaches, only 16 % of the test functions demonstrated weaker performances. This result confirms the importance of considering parameter interaction dependency.

Regarding the unimodal or low multimodal test functions (F1–F6), the results indicate that the static parameter setting approach performs the best. In reality, BNGA shows comparable results over those problems. However, it can be observed from the SP measures that BNGA converges more slowly to the optimum. This behaviour was expected, since, like any other adaptive parameter control method, BNGA needs to identify the most promising parameter states, whereas the static approach takes advantage of “off-line tuning”. It is interesting to note, however, that the high con-



Fig. 6 Results of the hyper-parameter sensitivity study: **a** FAUC-RMAB. **b** BNGA

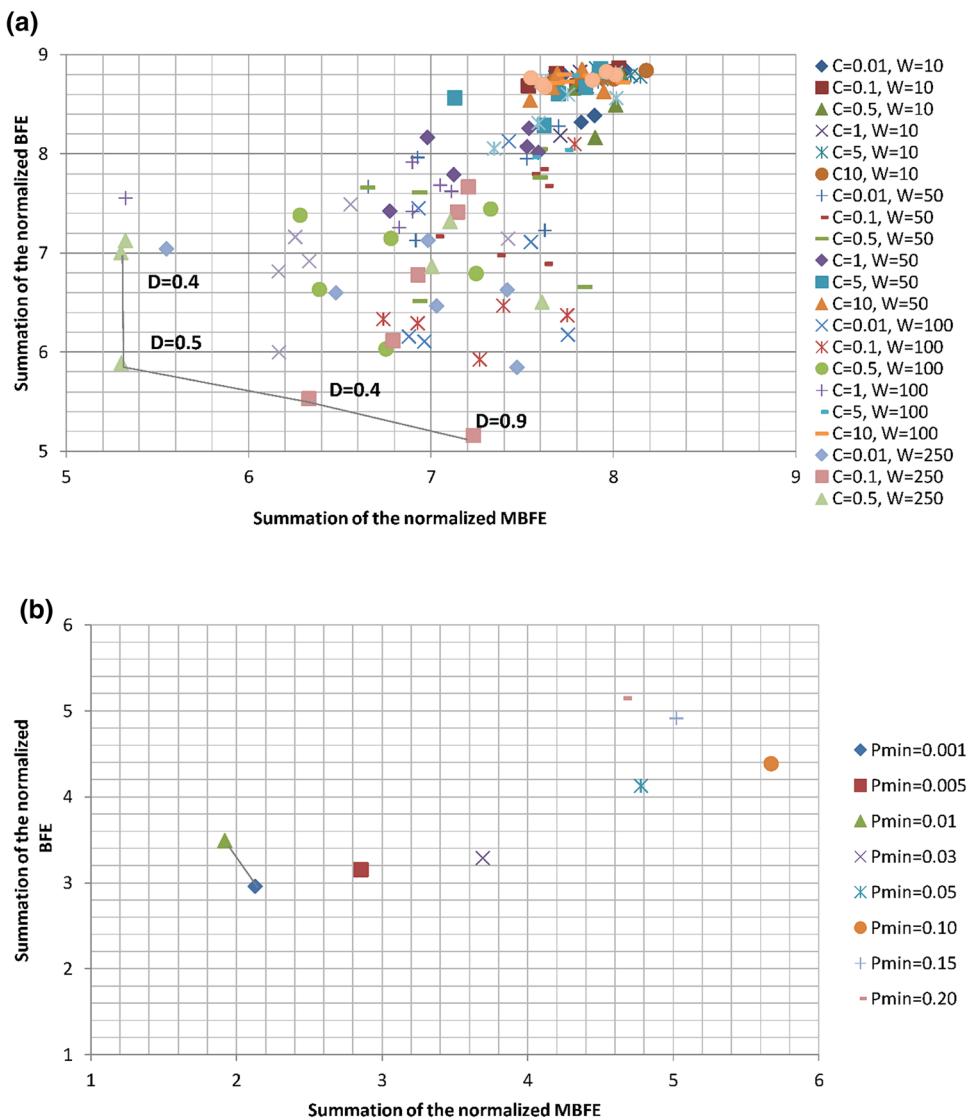


Table 3 Results format

F_j	Approach		
	MBFE	SR	p value
	STD	SP	
(tolerance)	BFE		

ditioning characteristic of F_3 poses great difficulties to all adaptive parameter control systems.

Comparing the general behaviour trends among the approaches is more difficult with the multimodal problems (F_7 – F_{25}). In fact, FAUC-RMAB and BNGA turned out to be the best approaches over most of these test functions. Nevertheless, except for F_{12} , none of the approaches were able to locate the global optimum within the prescribed tolerance. For the most complex multimodal subsets, which are the hybrid composition functions (F_{15} – F_{25}), it is interesting to note that the naïve approach appears to rank among the best. This suggests that over highly multimodal problems,

SSGA adaptive systems are no better than a pure exploration strategy in selecting parameter states. However, an investigation of the population EEB throughout the simulations indicates that none of the approaches were able to allocate resources following a useful diversity pattern ($GDM \rightarrow 1.0$ and $PCM \rightarrow 1.0$). This lack of adequate EEB states suggests a missing complementarity and synergy among the parameter states involved. This introduces the exploration/exploitation dilemma, but this time, at the parameter level. On the one hand, too many parameter states could reduce their participation in the search and dilute their performances. On the other hand, with too few parameter states, the probability of finding robust and efficient parameter configurations diminishes. In other words, balance will yet again have to be sought. However, further investigating this aspect would be beyond the scope of the present paper.

Table 5 presents the performance of the SSGA parameter setting approaches from the perspective of the function

Table 4 Results of the SSGA parameter setting approaches over the CEC'05 benchmark

	STATIC		NAIVE		PM		AP		FAUC-RMAB		BNGA	
F1	8.38E-09	100%	4.56E-01	0%	8.72E-09	100%	8.39E-09	100%	3.64E-06	96%	8.53E-09	100%
(< 1e-6)	1.63E-09	8.28E+03	8.33E-01	-	1.11E-09	9.01E+03	1.63E-09	1.37E+04	1.81E-05	3.60E+04	1.28E-09	7.39E+04
F2	8.80E-09	100%	9.57E+01	0%	9.81E-09	100%	1.24E-05	76%	1.61E-01	60%	1.05E-08	100%
(< 1e-6)	1.04E-09	6.50E+03	6.78E+01	-	2.46E-10	3.10E+04	4.52E-05	8.02E+04	5.02E-01	6.22E+04	1.13E-08	7.02E+04
F3	5.83E-09	<1E-3	1.05E+01	<u><1E-3</u>	8.94E-09	<u>0.010</u>	8.94E-09	<u><1E-3</u>	8.60E-09	<u><1E-3</u>	5.25E-09	0.940
(< 1e-6)	7.66E+00	0%	1.07E+06	0%	8.00E+04	0%	2.13E+05	0%	3.51E+05	0%	1.43E+05	0%
F4	1.29E+01	-	8.37E+05	-	9.13E+04	-	1.49E+05	-	4.08E+05	-	9.73E+04	-
(< 1e-6)	2.01E-02	<1E-3	2.02E+05	<u><1E-3</u>	4.84E+03	<u>0.001</u>	1.24E+04	<u><1E-3</u>	4.11E+04	<u><1E-3</u>	1.28E+04	<u><1E-3</u>
F5	2.91E-05	96%	4.12E+02	0%	5.94E-02	0%	1.22E+01	0%	2.45E+01	4%	9.10E-09	100%
(< 1e-6)	1.45E-04	9.16E+03	2.37E+02	-	7.55E-02	-	5.46E+01	-	8.91E+01	9.93E+05	7.74E-10	7.37E+04
F6	6.93E-09	8.80E-01	9.38E+01	<u><1E-3</u>	7.35E-04	<u><1E-3</u>	9.10E-06	<u><1E-3</u>	9.57E-09	<u><1E-3</u>	7.02E-09	<1E-3
(< 1e-6)	8.70E-09	100%	3.65E+02	0%	6.93E-02	0%	2.93E-03	12%	1.17E-01	76%	3.33E-07	96%
F7	1.17E-09	8.47E+03	7.37E+02	-	2.00E-01	-	8.09E-03	6.34E+05	4.83E-01	5.77E+04	1.47E-06	8.15E+04
(< 1e-6)	1.55E-09	<1E-3	7.51E+00	<u><1E-3</u>	1.58E-05	<u><1E-3</u>	1.23E-08	<u><1E-3</u>	3.49E-09	0.425	5.42E-09	0.545
F8	1.12E+00	72%	4.66E+03	0%	1.58E+02	0%	7.70E+01	4%	9.66E+01	16%	2.22E+01	40%
(< 1e-2)	1.83E+00	1.74E+04	1.46E+04	-	3.13E+02	-	2.33E+02	2.22E+06	1.84E+02	4.14E+05	7.40E+01	2.15E+05
F9	6.04E-09	<1E-3	2.30E+01	<u><1E-3</u>	7.61E-02	<u><1E-3</u>	3.47E-03	<u>0.003</u>	9.03E-09	<u><1E-3</u>	6.56E-09	0.096
(< 1e-2)	2.20E-01	0%	2.16E+00	0%	6.22E-01	0%	6.15E-01	0%	4.71E-01	0%	4.82E-01	0%
F10	1.59E-01	-	1.41E+00	-	4.58E-01	-	5.24E-01	-	3.12E-01	-	3.45E-01	-
(< 1e-2)	8.36E-02	<1E-3	4.54E-01	<u><1E-3</u>	1.48E-01	<u>0.002</u>	1.08E-01	<u>0.003</u>	9.10E-02	<u>0.010</u>	1.13E-01	<u>0.010</u>
F11	2.03E+01	0%	2.04E+01	0%	2.03E+01	0%	2.04E+01	0%	2.03E+01	0%	2.04E+01	0%
(< 1e-2)	8.00E-02	-	7.08E-02	-	7.29E-02	-	7.74E-02	-	7.59E-02	-	7.39E-02	-
F12	3.20E+01	3.78E-01	2.03E+01	0.545	2.01E+01	0.545	2.02E+01	0.521	2.01E+01	0.450	2.02E+01	0.206
(< 1e-2)	3.33E+01	0%	1.35E+01	0%	3.42E+01	0%	2.66E+01	0%	1.30E+01	0%	7.24E+00	0%
F13	1.39E+01	-	7.15E+00	-	1.28E+01	-	1.31E+01	-	6.38E+00	-	4.08E+00	-
(< 1e-2)	1.29E+01	<u><1E-3</u>	3.18E+00	<u>0.028</u>	8.95E+00	<u><1E-3</u>	9.95E+00	<u><1E-3</u>	1.99E+00	<u>0.025</u>	2.98E+00	<u><1E-3</u>
F14	3.77E+01	0%	1.85E+01	0%	4.32E+01	0%	4.01E+01	0%	2.41E+01	0%	1.93E+01	0%
(< 1e-2)	1.65E+01	-	6.92E+00	-	1.81E+01	-	1.90E+01	-	1.09E+01	-	7.38E+00	-
F15	1.29E+01	<u><1E-3</u>	7.36E+00	<u><1E-3</u>	1.59E+01	<u><1E-3</u>	1.39E+01	<u><1E-3</u>	9.95E+00	0.225	5.97E+00	0.597
(< 1e-2)	6.44E+00	0%	5.71E+00	0%	6.34E+00	0%	5.86E+00	0%	5.24E+00	0%	4.03E+00	0%
F16	1.62E+00	-	1.14E+00	-	1.44E+00	-	1.23E+00	-	2.16E+00	-	1.22E+00	-
(< 1e-2)	3.02E+00	<u><1E-3</u>	3.52E+00	<u>0.002</u>	3.92E+00	<u><1E-3</u>	3.73E+00	<u>0.002</u>	1.29E+00	<u>0.002</u>	1.73E+00	<u><1E-3</u>
F17	2.81E+03	24%	1.52E+03	0%	2.96E+03	0%	2.67E+03	8%	7.13E+02	4%	9.21E+02	24%
(< 1e-2)	4.39E+03	2.65E+04	1.54E+03	-	5.34E+03	-	4.55E+03	5.81E+05	1.06E+03	4.04E+05	2.31E+03	3.54E+05
F18	3.67E+00	0%	2.01E+00	0%	2.17E+00	0%	2.16E+00	0%	1.41E+00	0%	1.31E+00	0%
(< 1e-2)	4.16E-01	0.514	1.02E+00	<u>0.004</u>	8.95E-01	<u>0.004</u>	8.63E-01	<u>0.001</u>	5.02E-01	0.762	4.30E-01	<u><1E-3</u>
F19	3.31E-01	-	2.30E-01	-	3.48E-01	-	2.88E-01	-	3.67E-01	-	3.50E-01	-
(< 1e-2)	3.21E+00	<u><1E-3</u>	2.89E+00	<u><1E-3</u>	2.37E+00	<u>1.30E-02</u>	2.54E+00	<u>1.22E-02</u>	2.55E+00	0.059	2.32E+00	<u><1E-3</u>
F20	3.79E+02	0%	2.26E+02	0%	4.02E+02	0%	3.84E+02	0%	3.44E+02	0%	1.94E+02	0%
(< 1e-2)	1.02E+02	<u><1E-3</u>	1.19E+02	0.450	7.72E+01	-	1.05E+02	-	1.34E+02	-	1.10E+02	-
F21	1.81E+02	0%	1.42E+02	0%	1.87E+02	0%	1.87E+02	0%	1.61E+02	0%	1.52E+02	0%
(< 1e-2)	1.13E+02	<u><1E-3</u>	1.04E+02	<u><1E-3</u>	1.18E+02	<u><1E-3</u>	1.23E+02	<u><1E-3</u>	9.70E+01	<u>0.025</u>	1.04E+02	0.450
F22	1.72E+02	0%	1.59E+02	0%	1.87E+02	0%	1.96E+02	0%	1.70E+02	0%	1.42E+02	0%
(< 1e-1)	3.39E+01	-	2.33E+01	-	3.55E+01	-	5.06E+01	-	3.75E+01	-	2.18E+01	-
F23	1.19E+02	<u>0.010</u>	1.25E+02	0.082	1.22E+02	<u><1E-3</u>	1.28E+02	<u><1E-3</u>	1.23E+02	<u>0.008</u>	1.11E+02	<u><1E-3</u>
(< 1e-1)	8.01E+02	0.063	3.10E+02	0.650	3.57E+02	0.650	4.07E+02	0.650	3.00E+02	0.022	5.06E+02	0.650
F24	9.54E+02	0%	8.56E+02	0%	9.21E+02	0%	8.84E+02	0%	7.68E+02	0%	8.19E+02	0%
(< 1e-1)	1.16E+02	-	1.83E+02	-	1.77E+02	-	1.78E+02	-	2.58E+02	-	2.19E+02	-
F25	5.86E+02	0.063	3.72E+02	0.650	3.72E+02	0.113	3.80E+02	0.392	3.00E+02	0.083	3.00E+02	0.650
(< 1e-1)	9.53E+02	0%	8.80E+02	0%	9.23E+02	0%	8.91E+02	0%	8.40E+02	0%	7.92E+02	0%
F26	1.58E+02	-	1.67E+02	-	1.26E+02	-	2.49E+02	-	2.22E+02	-	2.17E+02	-
(< 1e-1)	3.00E+02	<u>0.002</u>	4.11E+02	0.099	5.15E+02	<u>0.047</u>	3.00E+02	<u>0.021</u>	3.10E+02	0.199	3.00E+02	0.009
F27	9.97E+02	0%	8.17E+02	0%	9.22E+02	0%	9.32E+02	0%	8.09E+02	0%	7.49E+02	0%
(< 1e-1)	2.56E+02	-	2.97E+02	-	2.73E+02	-	2.87E+02	-	2.89E+02	-	3.13E+02	-
F28	5.00E+02	<u>0.001</u>	5.00E+02	0.104	5.00E+02	<u>0.030</u>	5.00E+02	<u>0.023</u>	5.00E+02	0.104	3.00E+02	0.007

groups. The outcomes of these statistical tests are similar to those obtained from the independent function analyses sampled. The results confirm that the static parameter setting approach is as good as the best adaptive control systems over the most simple test functions (F1–F14), whereas the naïve approach is competitive over complex

test functions (F15–F25). However, this group perspective does not reveal the differences between the FAUC–RMAB and BNGA performances. What it clearly demonstrates is that over all the considered problems (F1–F25), these two methods stand out as the best SSGA parameter setting approaches.



Table 4 continued

	STATIC	NAIVE	PM	AP	FAUC-RMAB	BNGA						
F22 (<1e-1)	8.74E+02 7.00E+01	0% -	8.08E+02 3.56E+01	0% -	8.09E+02 4.20E+01	0% -	8.05E+02 1.19E+02	0% -	7.87E+02 3.00E+02	0% <1E-3	7.91E+02 7.50E+02	0% 0.705
	7.62E+02	0.001	7.69E+02	0.705	7.64E+02	0.705	3.00E+02	0.705	3.00E+02	<1E-3	7.50E+02	0.705
F23 (<1e-1)	1.10E+03 2.38E+02	0% -	9.65E+02 2.62E+02	0% -	8.89E+02 2.70E+02	0% -	8.70E+02 2.81E+02	0% -	8.38E+02 4.25E+02	0% 1.000	8.55E+02 5.59E+02	0% 1.000
	5.59E+02	0.078	5.59E+02	0.235	5.59E+02	1.000	4.25E+02	1.000	5.59E+02	0.027	5.59E+02	1.000
F24 (<1e-1)	4.70E+02 3.70E+02	0% -	2.63E+02 2.07E+02	0% -	4.35E+02 3.32E+02	0% -	2.68E+02 1.81E+02	0% -	2.48E+02 1.12E+02	0% -	2.55E+02 1.91E+02	0% -
	2.00E+02	0.124	2.00E+02	0.483	2.00E+02	0.678	2.00E+02	0.093	2.00E+02	<1E-3	2.00E+02	<1E-3
F25 (<1e-1)	1.23E+03 3.24E+02	0% -	2.15E+02 6.20E+01	0% -	3.51E+02 2.58E+02	0% -	2.85E+02 2.20E+02	0% -	2.45E+02 1.62E+02	0% -	2.28E+02 1.40E+02	0% -
	2.00E+02	0.047	2.00E+02	<1E-3	2.00E+02	0.096	2.00E+02	0.007	2.00E+02	0.096	2.00E+02	<1E-3

Table 5 Statistical comparison by function group

	STATIC	NAIVE	PM	AP	FAUC-RMAB	BNGA
F1-F14	0.480	0.002	0.014	0.079	0.259	0.003
F15-F25	<1E-3	0.610	0.001	0.004	0.909	<1E-3
F1-F25	0.001	0.002	<1E-3	0.001	0.226	<1E-3

5.2 Results of the EA parameter setting approaches

Since FAUC-RMAB and BNGA both demonstrated the highest global performance over F1 to F25 within the SSGA comparison framework, their results are now compared to the state-of-the-art EAs (CMA-ES and G-CMA-ES). Table 6 presents the comparison results.

Overall, Table 6 shows that FAUC-RMAB is statistically outperformed by other approaches over 60 % of the test functions, compared to 36 % for BNGA, 40 % for CMA-ES, and 20 % for G-CMA-ES. This confirms the robustness of G-CMA-ES and ranks BNGA second.

The recorded performance of G-CMA-ES is different from that indicated for functions F9 to F11 by Auger and Hansen [30]. This difference may be explained by different initialization seed numbers and the stochastic nature of EAs.

As for the SSGA parameter setting approaches, CMA-ES and G-CMA-ES are not able to allocate resources in accordance with the useful diversity concept. This may partially explain their difficulty in reaching the global optimum over highly multimodal test functions.

Table 7 presents the statistical comparison from the group of function point of view. FAUC-RMAB is the only adaptive control system dominated by the other approaches on F1–F14. It is interesting to note that, this time, G-CMA-ES is not statistically better than CMA-ES or BNGA, regardless of the group of functions considered. This is somewhat surprising, as G-CMA-ES was introduced specifically to improve the CMA-ES performance over multimodal problems. However, the lack of performance enhancement may be related to the strong asymmetry of the hybrid composition functions [29] and the maximum number of prescribed function evaluations preventing the identification of an effective λ value [49]. Finally, the fact that BNGA is statistically equivalent to

G-CMA-ES clearly suggests that with an appropriate parameter setting approach, SSGA represents a competitive EA strategy.

5.3 Influence of diversity on search method responses

The previous results indicated that none of the evaluated approaches successfully found the global optimum of the complex multimodal problems. We can conjecture that this may be related to a lack of useful diversity in the search process. To support this hypothesis, Figs. 7, 8, and 9 compare the EEB spectra achieved by all the parameter setting approaches over the test functions F6, F8, and F21, respectively. These test functions provide a general, but concise picture of the EEB management offered by each parameter setting strategy. In these charts, each color corresponds to a specific run. Accordingly, the points represent the EEB footprints of the parameter setting strategies, rather than their path history. The EEB exploration and exploitation axes were computed from the GDM and PCM formulations given by Eqs. 8 and 9, respectively.

Overall, the EEB of the static parameter setting approach (Figs. 7a, 8a, 9a) aptly describes the conventional EA evolution path; the search process is automatically directed towards the best individual found so far. Interestingly, this makes the EEB footprint compatible with its path history. Only F8 (Fig. 8) presents a different response. The results obtained for this test function will be discussed separately below. Surprisingly, the naïve adaptive approach (Figs. 7b, 8b, 9b) often settles in three specific EEB regions that are well balanced with respect to the exploration axis. On the other hand, PM, AP, FAUC-RMAB, and BNGA (Figs. 7c–f, 8c–f, 9c–f) demonstrate fairly similar EEB footprints.

CMA-ES and G-CMA-ES show very similar EEB footprints: few generations are allocated to extensive exploration, and most of their search processes are rapidly dedicated to the exploitation of a narrow landscape region. However, G-CMA-ES demonstrates more exploration capabilities over high multimodal problems (F21) than CMA-ES. This condition results from the restart feature embedded

Table 6 Results of the EA parameter setting approaches over the CEC'05 benchmark

		FAUC-RMAB	BNGA	CMA-ES	G-CMA-ES
F1	3.64E-06 (< 1e-6)	96% 5.83E-09	8.53E-09 3.60E+04	100% 1.28E-09	7.98E-09 7.39E+04
F2	1.81E-05 (< 1e-6)	60% 8.60E-09	1.05E-08 5.25E-09	100% 1.37E-09	1.87E+03 2.62E+03
F3	3.51E+05 (< 1e-6)	0% -	1.43E+05 9.73E+04	0% -	8.00E-09 6.47E+03
F4	4.08E+05 (< 1e-6)	1.28E+04 ≤1E-3	1.28E+04 ≤1E-3	100% 5.25E-09	8.07E-09 1.40E-09
F5	3.51E+05 (< 1e-6)	4% 9.57E-09	9.10E-09 ≤1E-3	100% 7.02E-09	7.82E-09 2.85E+03
F6	4.83E-01 (< 1e-2)	76% 9.03E-09	7.74E-10 ≤1E-3	96% 6.56E-09	100% 4.25E-09
F7	5.77E+04 (< 1e-2)	1.77E-01 ≤1E-3	7.37E+04 ≤1E-3	100% 3.01E-09	6.18E-09 2.02E-09
F8	4.13E+05 (< 1e-2)	16% 2.03E+01	2.22E+01 ≤1E-3	40% 2.00E+01	92% 8.86E-10
F9	4.14E+05 (< 1e-2)	0% 6.38E+00	4.14E+05 ≤1E-3	0% 2.00E+01	100% 5.42E+00
F10	5.77E+04 (< 1e-2)	0% 1.99E+00	5.77E+04 ≤1E-3	0% 2.98E+00	100% 5.09E+00
F11	4.14E+05 (< 1e-2)	1.96E+01 2.16E+00	4.14E+05 ≤1E-3	0% 1.87E+01	0% 1.50E+00
F12	5.77E+04 (< 1e-2)	4% 1.06E+03	9.21E+02 ≤1E-3	24% 3.21E+03	12% 9.47E+03
F13	5.77E+04 (< 1e-2)	0% 1.09E+01	1.93E+01 ≤1E-3	0% 3.54E+05	0% 1.63E+01
F14	5.77E+04 (< 1e-2)	0% 3.44E+02	5.77E+04 ≤1E-3	0% 7.97E-09	0% 5.48E-09
F15	5.77E+04 (< 1e-2)	0% 1.34E+02	0% 1.05E+02	0% 8.87E+00	0% 1.37E+02
F16	5.77E+04 (< 1e-2)	0% 3.18E+01	0% 5.02E-01	0% 3.47E+01	0% 9.86E-01
F17	5.77E+04 (< 1e-1)	0% 1.70E+02	0% 1.23E+02	0% 1.11E+02	0% 4.33E+00
F18	5.77E+04 (< 1e-1)	0% 3.00E+02	0% 0.178	0% 5.06E+02	0% 3.75E+02
F19	5.77E+04 (< 1e-1)	0% 2.58E+02	0% 2.19E+02	0% 8.20E+02	0% 7.95E+02
F20	5.77E+04 (< 1e-1)	0% 3.00E+02	0% 0.121	0% 3.00E+02	0% 2.13E+02
F21	5.77E+04 (< 1e-1)	0% 8.09E+02	0% 2.98E+02	0% 7.49E+02	0% 4.34E+02
F22	5.77E+04 (< 1e-1)	0% 7.87E+02	0% 1.15E+02	0% 7.91E+02	0% 4.65E+01
F23	5.77E+04 (< 1e-1)	0% 8.38E+02	0% 5.59E+02	0% 8.55E+02	0% 3.46E+02
F24	5.77E+04 (< 1e-1)	0% 2.48E+02	0% 1.12E+02	0% 2.55E+02	0% 4.76E+02
F25	5.77E+04 (< 1e-1)	0% 2.45E+02	0% 1.62E+02	0% 2.28E+02	0% 4.09E+02

Table 7 Statistical comparison by function group

	FAUC-RMAB	BNGA	CMA-ES	G-CMA-ES
F1-F14	0.005	0.094	0.633	0.008
F15-F25	1.000	0.086	0.842	1.000
F1-F25	0.119	0.421	0.474	0.061

within G-CMA-ES. Nevertheless, as shown in Table 6, the higher exploration capability of G-CMA-ES remains insufficient to prevent the premature convergence issue observed over the higher multimodal problems (i.e. $GDM \rightarrow 0$ and $PCM \rightarrow 1$).

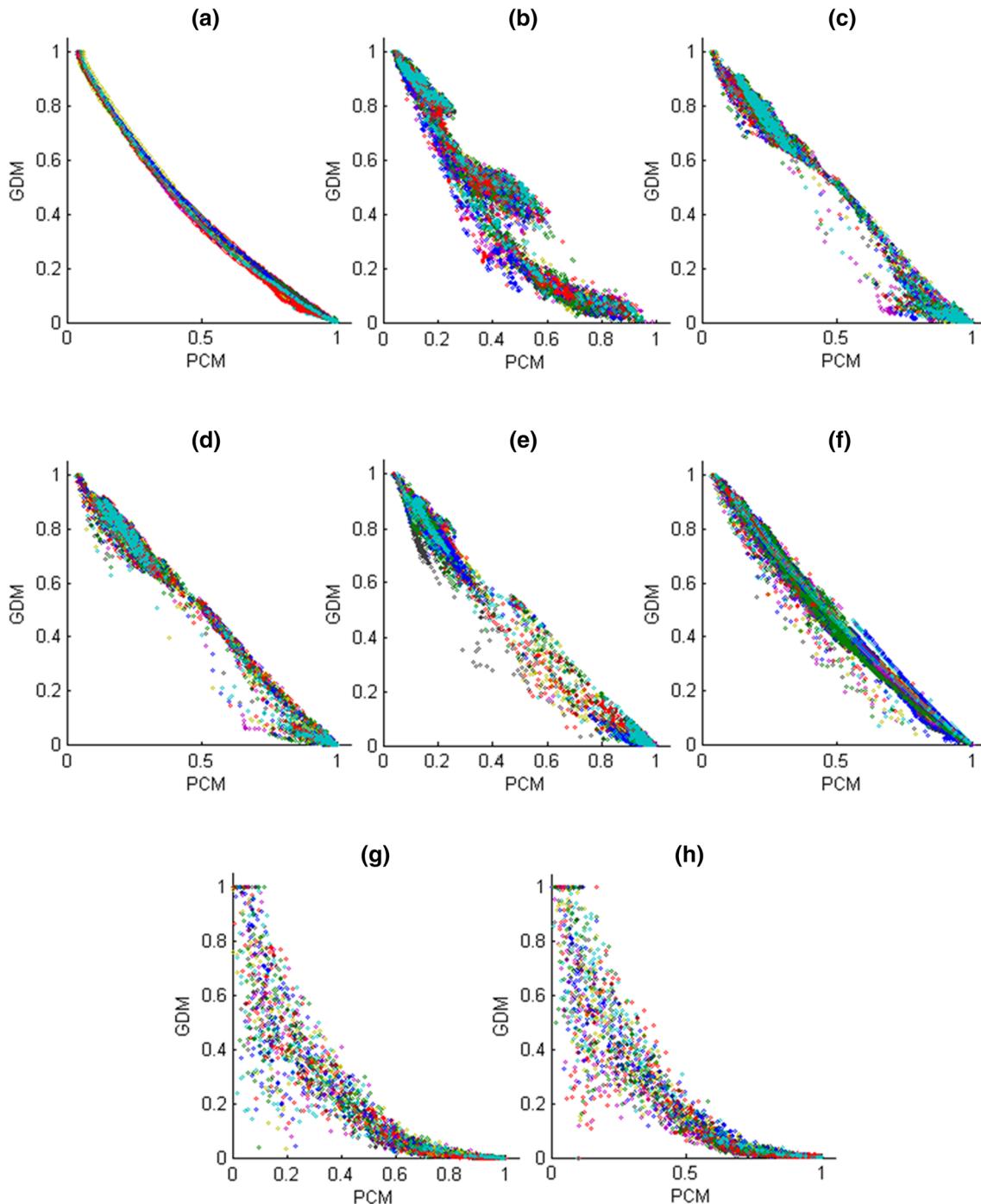


Fig. 7 EEB history over F6 (shifted Rosenbrock's function) with: **a** static parameter setting, **b** naïve approach, **c** PM, **d** AP, **e** FAUC-RMAB, **f** BNGA, **g** CMA-ES, and **h** G-CMA-ES

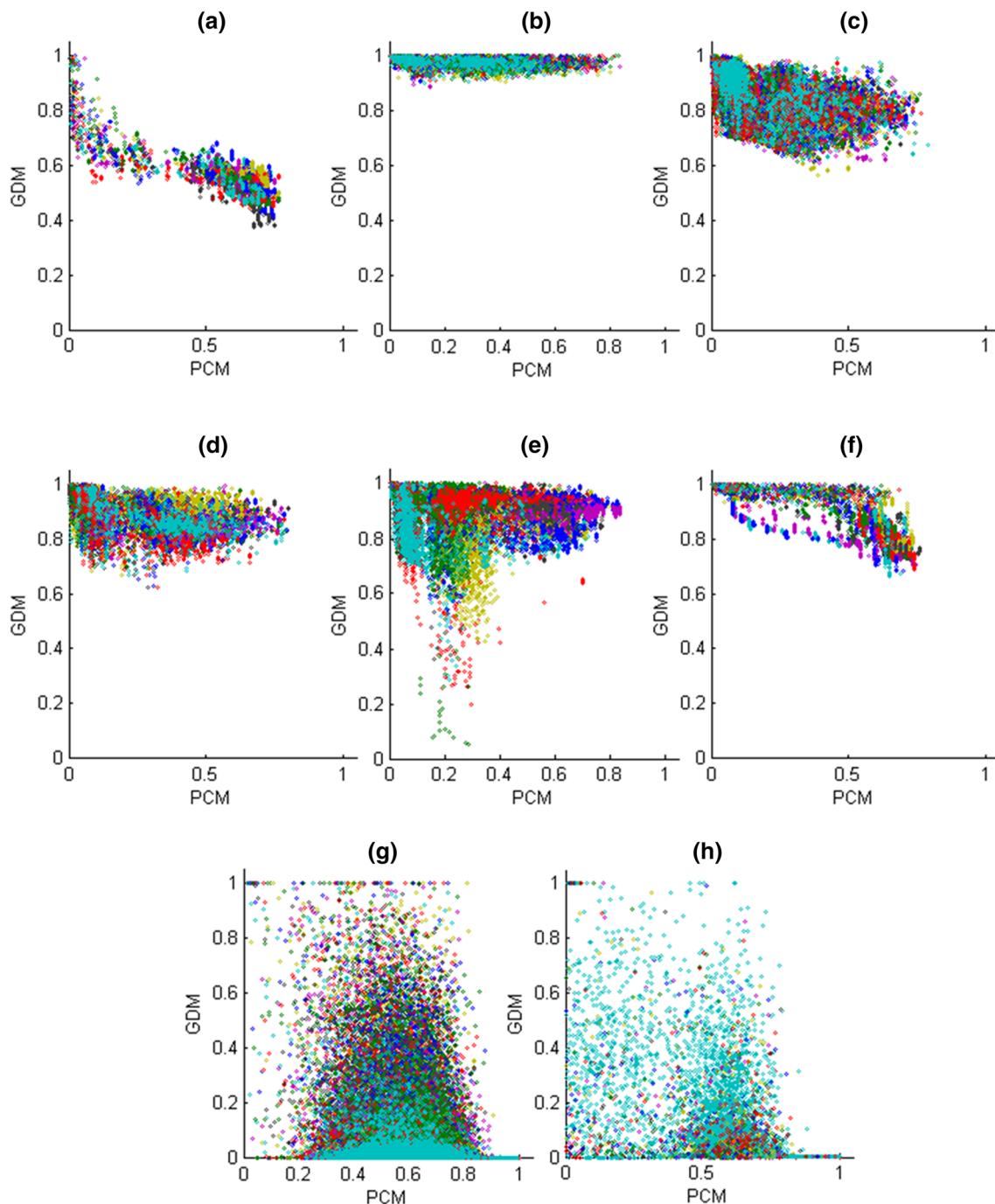


Fig. 8 EEB history over F8 (shifted rotated Ackley's function with global optimum on bound) with: **a** static parameter setting, **b** naïve approach, **c** PM, **d** AP, **e** FAUC-RMAB, **f** BNGA, **g** CMA-ES, and **h** G-CMA-ES

Test function F8 (Fig. 8) offers interesting information, as each parameter setting strategy presents a very different behaviour from the other test functions considered. In fact, F8 describes an “egg box” landscape, with the global optimum basin located on its bound. Except for rare FAUC-RMAB generations, where individuals seem to be spread throughout one or more local basins of attraction (Fig. 8e), most adap-

tive approaches continuously spread their resources across the landscape ($GDM \rightarrow 1$). In contrast, the CMA-ES and G-CMA-ES EEB footprints suggest that they concentrate the vast majority of their resources in a very few local basins of attraction. Interestingly, a differentiating feature of these two strategies is highlighted in this test function; they cover up to approximately 80 % of the EEB space. Nevertheless,

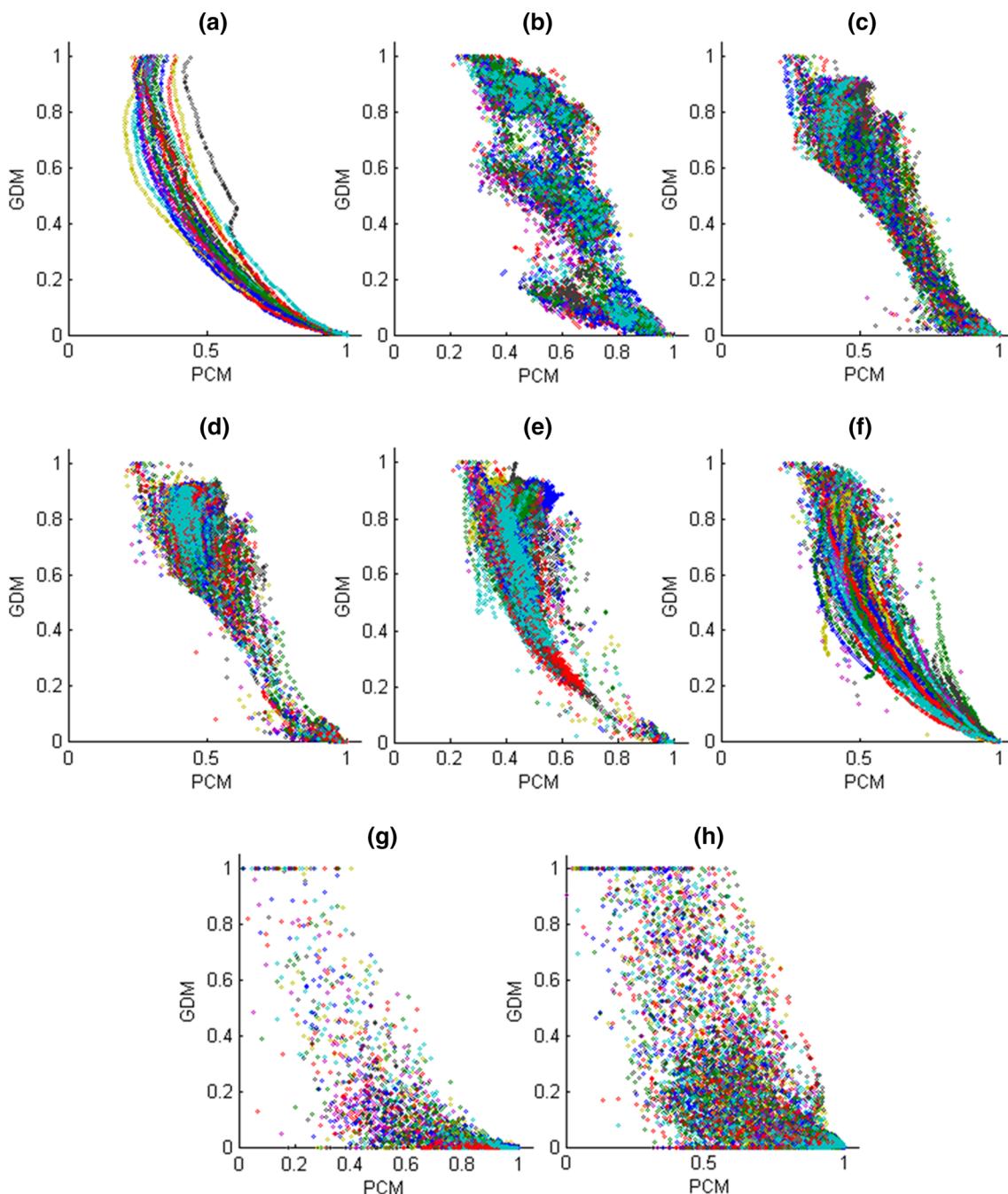


Fig. 9 EEB history over F21 (rotated hybrid composition function) with: **a** static parameter setting, **b** naïve approach, **c** PM, **d** AP, **e** FAUC–RMAB, **f** BNGA, **g** CMA-ES, and **h** G-CMA-ES

practically all of them remain stuck in a local optimum basin (Table 6).

The preceding observations on the population EEB tend to verify the opening hypothesis on useful diversity, and, in other words, suggest an apparent lack of complementarity and synergy among parameter states, especially over multimodal landscapes, where premature convergence is generally observed (i.e. $GDM \rightarrow 0$ and $PCM \rightarrow 1$). Determining the

best combination of parameter states involved may result in marked performance gain.

6 Concluding discussion

Parameter setting is a longstanding issue in the EA field: specific parameters are responsible for the search path followed

and, therefore, drive the performance of the optimizer. On the other hand, these parameters are very difficult to determine, since optimal parameter states are problem and evolution dependent.

The present paper proposes an adaptive system capable of coping with all the parameter setting dependencies. This adaptive approach is based on a Bayesian network and is applied to genetic algorithms (BNGA). The proposed strategy addresses all three dependencies: (1) problem dependency, by ensuring uniform parameter state probabilities at the start of the search; (2) evolution dependency, by updating parameter state probabilities as new evidence appears; and (3) parameter interaction, by supporting relationships through the graphical model. Compared to other available adaptive parameter setting approaches, BNGA appears to be the only one capable of integrating parameter interactions, while autonomously managing interactions strength as the search evolves. Furthermore, BNGA requires only one hyper-parameter, compared to as many as five for comparable systems. For the authors, these characteristics seem clearly promising in terms of enhancing search robustness and efficiency.

The presented study initiated a comparative evaluation following the CEC'05 benchmark. In the first step, BNGA is compared to a static parameter setting, a naïve approach, and three recognized adaptive methods (PM, AP, and FAUC-RMAB). This comparison was performed within an SSGA framework involving the control of nine parameters. All the adaptive control approaches were based on the same credit assignment scheme. This means that four parameters related to offspring creation were judged on their ability to improve the fitness, and five parameters related to population governance were evaluated on their ability to manage the EEB.

In summary, the statistical results obtained led to three main conclusions. First, when considering the management of parameter interactions, BNGA shows the highest level of robustness. Second, BNGA and FAUC-RMAB stand out as the best SSGA adaptive approaches. Their performances were similar to those of static parameter setting over unimodal test functions, and statistically equivalent to that of the naïve approach over complex multimodal problems. For all the test functions, they were superior to all the parameter setting approaches included in the comparison. These observations clearly attest to the importance of the role of an adaptive control system. Adaptive systems must indeed be proficient in matching exploitative or explorative strategies. Finally, the hyper-parameter sensitivity study confirmed the implementation advantage of BNGA over other adaptive systems, such as FAUC-RMAB. In fact, the minimal number of hyper-parameters involved gives BNGA the greatest generalization power of all the adaptive systems.

The study further compared BNGA and FAUC-RMAB to state-of-the-art EAs: CMA-ES and G-CMA-ES. The statistical results demonstrated that BNGA performs as well as CMA-ES and G-CMA-ES. From a practitioner standpoint, this general conclusion is attractive, since it indicates that with the appropriate parameter setting, SSGA is highly competitive, while offering a more flexible framework than CMA-ES for customizing search components. This last observation certainly describes the central contribution of the present investigation, but more importantly suggests that SSGA remains a rich research avenue that is yet to be exploited. It is also important to mention that while associated with GA in the present investigation, the proposed adaptive approach is not limited to GA applications, but rather describes a generic control system that can be effective for any population-based search algorithm.

This analysis also revealed that none of the studied approaches found the global optimum over complex multimodal problems. Additional information on the EEB evolution gathered during the search progression and presented for all studied parameter setting approaches working on specific test functions indicated that a lack of complementarity and synergy among the parameter states may plausibly be responsible for premature convergence.

The intrinsic BNGA attributes, combined with the presented results, suggest that BNGA should also be efficient over large-scale global optimization (LSGO) problems. Actually, over the last few years, several algorithms have been developed to deal with LSGO problems [50]. These algorithms may be divided into two general categories: on the one hand, there are algorithms using landscape decomposition strategies, commonly known as cooperative coevolution (CC) methods, aimed at reducing the search complexity of LSGO problems. The main challenge associated with these algorithms resides in the optimal decomposition scheme, markedly over non-separable problems. On the other hand, the second group encompasses algorithms focusing on the improvement of conventional metaheuristics. This algorithm group faces the challenge of preventing premature convergence, which is exacerbated over LSGO problems, by providing enough exploration, while keeping runtimes at practical levels. Experiments over controlled benchmarks demonstrated that the algorithms' performances are problem dependent [50, 51]. Clearly, these experiments indicated that LSGO problems are best solved using adaptive systems capable of leveraging population diversity to steer the search, which is a key BNGA characteristic. Therefore, since BNGA also assures efficient runtimes and requires only one hyper-parameter, future works will evaluate the scalability of the BNGA and EEB schemes to LSGO problems. The procedure developed in this study will be tested over the latest LSGO benchmark (CEC'2013) [52].

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Whitacre JM (2011) Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. *Computing* 93:121–133
- Whitacre JM (2011) Survival of the flexible: explaining the recent popularity of nature-inspired optimization within a rapidly evolving world. *Computing* 93:135–146
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- Bäck T (1992) The interaction of mutation rate, selection, and self-adaptation within genetic algorithm. In: Proceedings of the 2nd international conference on parallel problem solving from nature, PPSN 2, Brussels, pp 85–92
- Rojas I, Gonzalez J, Pomares H, Merelo JJ, Castillo PA, Romero G (2002) Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Trans Syst Man Cybern Part C Appl Rev* 32(1):31–37
- De Jong KA (1975) An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation, Univ. of Michigan, Ann Arbor
- Grefenstette JJ (1986) Optimization of control parameters for genetic algorithms. *IEEE Trans Syst Man Cybern SMC-16(1)*:122–8
- Schaffer JD, Caruana RA, Eshelman LJ, Das R (1989) A study of control parameters affecting online performance of genetic algorithms for function optimization. In: Proceedings of the 3rd international conference on genetic algorithms, ICGA, Fairfax, pp 51–60
- Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. *IEEE Trans Evol Comput* 3(2):124–141
- Karafotias G, Hoogendoorn M, Eiben AE (2015) Parameter control in evolutionary algorithms: trends and challenges. *IEEE Trans Evol Comput* 19(2):167–187
- Eiben AE, Schippers CA (1998) On the evolutionary exploration and exploitation. *Fundam Inf* 35:1–16
- Corriveau G, Guilbault R, Tahar A, Sabourin R (2012) Review and study of genotypic diversity measures for real-coded representations. *IEEE Trans Evol Comput* 16(5):695–710
- Pandey HM, Chaudhary A, Mehrotra D (2014) A comparative review of approaches to prevent premature convergence in GA. *Appl Soft Comput* 24:1047–1077
- Smith JE, Fogarty TC (1997) Operator and parameter adaptation in genetic algorithms. *Soft Comput* 1:81–87
- Tuson A, Ross P (1998) Adapting operator settings in genetic algorithms. *Evol Comput* 6(2):161–184
- Davis L (1989) Adapting operator probabilities in genetic algorithms. In: Proceedings of the 3rd international conference on genetic algorithms, ICGA, Fairfax, pp 61–69
- Julstrom BA (1995) What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In: Proceedings of the 6th international conference on genetic algorithms, ICGA, Pittsburgh, pp 81–87
- Barbosa HJC, Sá AM (2000) On adaptive operator probabilities in real coded genetic algorithms. In: Workshop on advances and trends in artificial intelligence for problem solving, Santiago, pp 1–6
- Herrera F, Lozano M (1996) Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In: Genetic algorithms and soft computing, Physica-Verlag, pp 95–125
- Motoki T (2002) Calculating the expected loss of diversity of selection schemes. *Evol Comput* 10(4):397–422
- Maturana J, Saubion F (2008) A compass to guide genetic algorithms. In: Proceedings of the 10th international conference on parallel problem solving from nature, PPSN X, Dortmund, pp 256–265
- Fialho ARS (2011) Adaptive operator selection for optimization, Ph.D. Dissertation, Univ. Paris-Sud 11, France
- Whitacre JM, Pham TQ, Sarker RA (2006) Use of statistical outlier detection method in adaptive evolutionary algorithms. In: Proceeding of the genetic and evolutionary computation conference, GECCO 06, Seattle, pp 1345–1352
- Rechenberg I (1973) Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog, Stuttgart
- Goldberg DE (1990) Probability matching, the magnitude of reinforcement, and classifier system bidding. *Mach Learn* 5:407–425
- Thierens D (2005) An adaptive pursuit strategy for allocating operator probabilities. In: Proceeding of the genetic and evolutionary computation conference, GECCO 05, Washington, DC, pp 1539–1546
- DaCosta L, Fialho A, Schoenauer M, Sebag M (2008) Adaptive operator selection with dynamic multi-armed bandits. In: Proceeding of the genetic and evolutionary computation conference, GECCO 08, Atlanta, pp 913–920
- Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of the 1996 IEEE international conference on evolutionary computation, ICEC'96, Nagoya, pp 312–317
- Hansen N, Kern S (2004) Evaluating the CMA evolution strategy on multimodal test functions. In: Proceedings of the 8th international conference on parallel problem solving from nature, PPSN VIII, Birmingham, pp. 282–291
- Auger A, Hansen N (2005) A restart CMA evolution strategy with increasing population size. In: Proceedings of the 2005 IEEE congress on evolutionary computation, CEC 05, Edinburgh, pp 1769–1776
- Eiben AE, Smith J (2015) From evolutionary computation to the evolution of things. *Nature* 521:476–82
- Larranaga P, Lozano JA (eds) (2002) Estimation of distribution algorithms: a new tool for evolutionary computation. Kluwer Academic Publishers, Norwell
- Pavón R, Díaz F, Luzón V (2008) A model for parameter setting based on Bayesian networks. *Eng Appl Artif Intell* 21:14–25
- Pavón R, Díaz F, Laza R, Luzón V (2009) Automatic parameter tuning with a Bayesian case-based reasoning system. A case study. *Expert Syst Appl* 36:3407–3420
- Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. 2nd print. Morgan Kaufmann Publishers, San Francisco
- Pearl J (1986) Fusion, propagation, and structuring in belief networks. *Artif Intell* 29(3):241–288
- Henrion M (1989) An introduction to algorithms for inference in belief nets. In: Proceedings of the 5th annual conference on uncertainty in artificial intelligence, UAI'89, Windsor, pp 129–138
- Cooper GF (1990) The computational complexity of probabilistic inference using Bayesian belief networks. *Artif Intell* 42(2–3):393–405
- Druzdzel MJ, van der Gaag LC (2000) Building probabilistic networks: “where do the numbers come from?”—Guest editors’ introduction. *IEEE Trans Knowl Data Eng* 12(4):481–486
- Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the 2nd

- international conference on genetic algorithms, ICGA, Cambridge, pp 41–49
41. Burke EK, Gustafson S, Kendall G (2004) Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Trans Evol Comput* 8:47–62
 42. Suganthan PN et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization. Nanyang Tech. Univ. Singapore, Tech. Rep. #2005005
 43. Sinha A, Tiwari S, Deb K (2005) A population-based, steady-state procedure for real-parameter optimization. In: Proceedings of the 2005 IEEE congress on evolutionary computation, CEC 05, Edinburgh, pp 514–521
 44. Hutter M, Legg S (2006) Fitness uniform optimization. *IEEE Trans Evol Comput* 10:568–589
 45. Deb K, Anand A, Joshi D (2002) A computationally efficient evolutionary algorithm for real-parameter optimization. *Evol Comput* 10(4):371–395
 46. Ono I, Kobayashi S (1997) A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In: Proceedings of the 7th international conference on genetic algorithms, ICGA 1997, East Lansing, pp 246–253
 47. Ballester PJ, Carter JN (2004) An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimization. In: Proceeding of the genetic and evolutionary computation conference, GECCO 04, Seattle, pp 901–913
 48. García S, Molina D, Lozano M, Herrera F (2009) 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 Heuristics* 15(6):617–644
 49. Lunacek M, Whitley D (2006) The dispersion metric and the CMA evolution strategy. In: Proceeding of the genetic and evolutionary computation conference, GECCO 06, Seattle, pp 477–484
 50. Mahdavi S, Shiri ME, Rahnamayan S (2015) Metaheuristics in large-scale global continues optimization: a survey. *Inf Sci* 295:407–428
 51. Cheng R, Jin Y (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
 52. Li X et al (2013) Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization. Tech. Rep. Evolutionary Computation and Machine Learning Group, RMIT University, Australia

