



HAL
open science

From Adaptive to More Dynamic Control in Evolutionary Algorithms

Giacomo Di Tollo, Frédéric Lardeux, Jorge Maturana, Frédéric Saubion

► **To cite this version:**

Giacomo Di Tollo, Frédéric Lardeux, Jorge Maturana, Frédéric Saubion. From Adaptive to More Dynamic Control in Evolutionary Algorithms. *Evolutionary Computation in Combinatorial Optimization*, 2011, Turin, Italy. pp.130 - 141, 10.1007/978-3-642-20364-0_12 . hal-03255423

HAL Id: hal-03255423

<https://univ-angers.hal.science/hal-03255423>

Submitted on 18 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Adaptive to More Dynamic Control in Evolutionary Algorithms

Giacomo di Tollo¹, Frédéric Lardeux¹, Jorge Maturana², and Frédéric Saubion¹

¹ LERIA, University of Angers (France)
name@info.univ-angers.fr

² Instituto de Informática, Universidad Austral de Chile (Chile)
jorge.maturana@inf.uach.cl

Abstract. Adaptive evolutionary algorithms have been widely developed to improve the management of the balance between intensification and diversification during the search. Nevertheless, this balance may need to be dynamically adjusted over time. Based on previous works on adaptive operator selection, we investigate in this paper how an adaptive controller can be used to achieve more dynamic search scenarios and what is the real impact of possible combinations of control components. This study may be helpful for the development of more autonomous and efficient evolutionary algorithms.

1 Introduction

From a high level point of view, Evolutionary Algorithms (EAs) [5] manage a set (population) of possible configurations of the problem solutions (individuals), which are progressively modified by variation operators, in order to converge to an optimal solution or, at least, to a sub-optimum of good quality. EAs have been successfully applied to various combinatorial and continuous optimization problems. It is now clearly assessed that the successful use of an EA mainly relies on the suitable combination of several components: the choice of an appropriate encoding of the problem, the definition of efficient operators and the adjustment of the behaviour of the algorithm by means of parameters. One may indeed identify two general classes of parameters: *behavioural*, mainly operator application rates or population size and *structural*, that define the main features of the algorithm and could eventually transform it radically, e.g., those related with the encoding and the choice of operators.

Even if significant progresses have been achieved in parameter setting [10], this setting often relies on empirical rules and/or problem-domain knowledge and often involve time-consuming experiences. Following the classification proposed in [4], we usually distinguish between *tuning* techniques that aim at adjusting the algorithm's parameters before the run and *control* techniques that modify the algorithm's behaviour along the search process. Efficient tuning methods are now available using either statistical tools[2] or meta-algorithms[8][14].

Back to the performance of the EAs, the management of the balance between the exploration and the exploitation of the search space (also known as

diversification and *intensification*) is largely recognized as a key feature for the overall performance of the search process. Therefore, there is a need for finding a suitable criteria that can be used for this management.

Several control techniques have been proposed, especially to tackle the selection problem, i.e., which variation operator should be applied at a given step of the search. Recent works on adaptive pursuit [17] or Adaptive Operator Selection (AOS) [7, 6] have provided promising results on the adaptive management of the search process but focused on a single criterion: the quality of the population. [12] proposed a control method for EAs, whose purpose is to manage simultaneously the mean quality and the diversity of the population, which are clearly related to intensification and diversification. The main idea is to consider the impact of an operator over the search and to keep a trace of its successes or failures. This knowledge is then used to choose the next operator to apply.

In [11] combinations of various components of the controller have been tested in order to achieve the best possible results by combining ideas from [6, 3, 12]. In [13], the adaptive management of the operators themselves is addressed. In these works, the search is guided by a fixed scheme to control the trade-off between intensification and diversification. The described controller is *adaptive* since it adapts its behaviour to the current state of the search and the algorithm's environment (including its components and some fixed parameters). However, this control is not really *dynamic* in the sense that the balance between intensification and diversification is keep fixed. Since the search may need different emphasis on these trends in different moments of the search in order to obtain better results [9], we want to explore ways to adjust this trade-off dynamically. In this work, our purpose is twofold:

- to provide a fair a clear experimental analysis of the controller behaviour: different level of control were interleaved in [13] and we need to assess the control ability of the controller itself.
- to provide an analysis of the dynamic control ability of the controller and of its potential ability to manage more dynamic strategies.

Such a study will help algorithm designers to better understand the actual effects of such adaptive control techniques and thus to select the suitable components for achieving more autonomous algorithms. This paper is organized as follows. The controller is presented in Sec.2, an experimental analysis is offered in Sec. 3 and 4, before drawing conclusions and outlining future work in Sec. 5.

2 Adaptive Operator Selection

Parameter tuning implies a high cost, reason why we will focus on *parameter control* and more specially on the adaptive control of the applications rates of operators that can be used in an EA. Considering a basic steady state EA, these parameters are used to select for each generation which operator will be applied. Actually, the concept of AOS has been widely studied for EAs and we may decompose it in the basic actions depicted in Fig. 1 and described below.

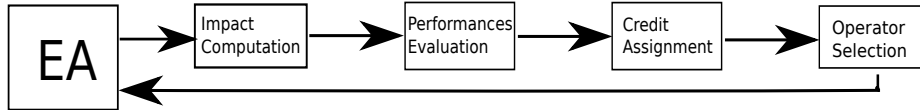


Fig. 1. AOS General Scheme

Impact Computation The first task consists in assessing the performances of the operator that has been applied. Following the ideas developed in [12], we evaluate an operator according to its impact on both quality and diversity, which can be seen as criteria that reflect the balance between intensification and diversification. Since considering a single value could not reflect the properties of an operator, it is more representative to record several values over a sliding window [11] of size T . Different functions can be used to compute the performance of the operator such as mean value or, as suggested in [6], the extreme value (i.e., max). We will show that the choice of this function may fully change the operator's behaviour. Once the impacts in terms of quality and diversity have been computed, they can be used to evaluate the operator.

Performance Evaluation The method presented in [12] plots ΔD (diversity variation) vs ΔQ (quality variation). In this framework, an user-defined angle θ is introduced to set the desired compromise between quality and diversity. Its value varies between 0 (maximum reachable diversity) and $\frac{\pi}{2}$ (maximum reachable quality). The similarity between the direction induced by the operator and the desired search direction (defined by θ) is computed as $(\sin(\theta) \cdot \delta D + \cos(\theta) \cdot \delta Q) / \sqrt{\sin(\theta)^2 + \cos(\theta)^2}$. Being this a multi-criteria measure, other evaluation functions could be used (e.g., Pareto Dominance or Pareto Rank [13]), however, they will not be considered in this study.

Credit Assignment Since we may have a wide variety of operator profiles, we are interested in comparing –and rewarding– them in a fair way. Performance is normalized according to the highest value and to the execution time of the operator (to set the trade-off between obtaining good results and the time invested in doing it). The credit register is then used by the the operator selection process. Note that, as for impact computation, the rewards can be computed over a given period of time either using the *maximum* reward or the *mean* reward. Nevertheless, it has been shown that this choice level has a negligible effect on the overall behaviour of the controller[11].

Operator Selection The operator selection has been considered from two basic points of view. On the one hand, Probability Matching (PM) is the most common method, based on a roulette-wheel routine: the application rate is proportional to the reward. On the other hand, operator selection is related to reinforcement learning problems since it basically consists in discovering the optimal application policy, i.e., applying the best possible operators, without neglecting

the potential offered by formerly-bad and not-used ones. Following ideas stemming from Multi-Armed Bandit methods (MAB), we used an Operator Selection scheme that has been firstly investigated in [3]. In MAB, the next operator is chosen according to the following formula: $MAB_{o,t} = r_{o,t} + C \sqrt{\frac{\log \sum_k n_{k,t}}{n_{o,t}}}$ where $r_{o,t}$ is the cumulated reward obtained by operator o at time t , and n is the number of times operator o has been applied so far. The scaling factor C^1 is used to properly balance the trade-off between both terms. Furthermore, $r_{o,t}$ and $n_{o,t}$ are reset when a change in operators' behaviour is detected (according to a Page-Hinkley test). This formula relies on the fact that all operators are available from the beginning of the search.

3 Quality and Diversity Management

Experiments have been performed on the satisfiability problem (SAT) [1] for two reasons: first, because SAT can encode a variety of problems with different fitness landscapes; then, because we use a SAT-specialized algorithm [18] that has several crossover operators whose performance is already known due to previous studies [13]. Population size has been set to 30 for every different experiments and a set of 20 crossover operators was used (to be detailed in Sec. 4). To show that the controller's behaviour is not dependent on the category of instances, we use three SAT instances coming from both random (F500) and handmade categories (SIMON, 3BIT). These instances have been chosen because they are representative of different kind of instances and show different landscapes². In the following experiments we will focus on the behaviour of the population along the execution time in terms of mean quality and diversity rather than the quality of the solution. Please notice that the execution time is constant for each operator, so it will be assessed as the number of crossovers performed during the search. We have observed that, with regards to the issues we want to highlight, the behaviour of the controller is relatively equivalent for the considered instances, reason why we will focus on the most representative figures. About 100 full experiments were executed for all different combinations of algorithm's components and strategies to change the angle's value.

3.1 The Trade-off Hypothesis

Previous works claimed that the angle value $\frac{\pi}{4}$ would lead to a good compromise between intensification and diversification. Our purpose is first to check this hypothesis. We will present experiments made to compare PM and MAB, considering how the *mean* and *max* impact computation criteria influence their behaviour (see Sec. 2, § 'Impact Computation').

¹ Let us note that some parameters are involved in MAB, but as studied in [11], their influence is much less significant than initial operator application rates and they have relative stable values over wide set of benchmarks.

² For more details, we forward the interested reader to the SAT competition's website <http://www.satcompetition.org/>.

To start, we fixed $\theta = \frac{\pi}{4}$ on the three considered instances. The following pictures show the normalized population’s entropy and the cost of all individuals as they appear in each step (i.e., at each crossover, shown on the x-axis).

When using PM, we can see that the search behaviour shows great differences between the *mean* and *max* criteria. As shown in the pictures, the *mean* approach can converge, depending on the instance, to both intensification (Fig. 2a) or diversification behaviours (high entropy, Fig. 2b).

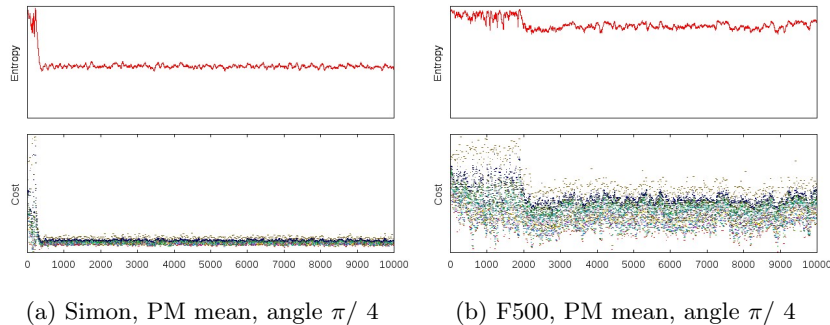


Fig. 2. Experiments with fixed angle $\pi/4$

The same uncertainty holds when introducing the MAB approach. This can lead either to a strong instability (Fig. 3a), without evident differences between the *mean* and *max* approach, or to a behaviour favouring diversification (Fig. 3b)³.

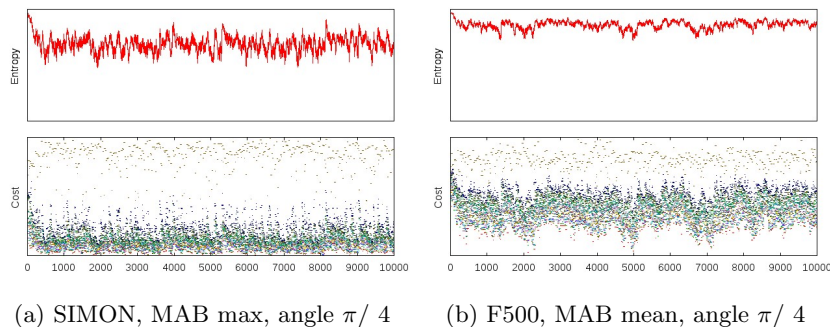


Fig. 3. Experiments with fixed angle $\pi/4$

We have seen that setting the value of θ to $\pi/4$ produces a variety of different behaviours (and not necessarily an intermediate compromise between diversification and intensification). Apparently, the instance’s features (fitness landscape) or other settings of the controller (the operator selection scheme used) are stronger factors than a fixed “universal“ value for θ . In order to elucidate the real effect of the angle over the search, we will observe how the search behaviour changes when the value of θ is modified during the search.

³ In this case, the controller shows a strong use of operator 6011, namely the only one able to diversify, see Sec. 4.

3.2 Towards a More Dynamic Control

We have defined different strategies in order to dynamically change the angle value during search:

- To split the execution time in several epochs and alternates the angle value between 0 and $\frac{\pi}{2}$ (*ALWAYSMOVING*).
- To split the execution time in several epochs and angle values in equally distributed levels in $[0, \frac{\pi}{2}]$ for each one. Angle can either increase (*ANGLEINCREASE*) or decrease (*ANGLEDECREASE*);
- To change the angle linearly over the execution time in $[0, \frac{\pi}{2}]$, either increasing (*LINEARINCREASE*) or decreasing (*LINEARDECREASE*).

In the following pictures we plot, along with the population’s Cost and Entropy, the value of angle θ during the execution, in the range $[0, \frac{\pi}{2}]$. It has been remarked that the EA behaviour does not change progressively product of small angle variations, either when using the *max* or the *mean* criteria.

In some cases it seems to exist a threshold from where the behaviour changes radically, supporting the hypothesis of the “positive feedback” proposed in [11], except that this threshold is not always found at $\theta = \pi/4$. Notice, for instance, how MAB with the *max* criterion and *ANGLEINCREASE* strategy switches near $\frac{\pi}{2}$ (Fig. 4b), while the switch occurs near $\frac{\pi}{4}$ for *ANGLEINCREASE-mean* (Fig. 4a) and *ANGLEDECREASE* both *mean* and *max* (Fig. 4d). In some other cases, particularly those mixing PM with decreasing strategies (with *mean* and *max*), the behaviour seems to be unaffected by the value of θ (Fig. 4c). When increasing the angle, the *mean* criterion shows itself more reactive than *max*. We can say that defining a dynamic strategy which accounts for progressive angle variations does not represent an advantage in terms of control, since the search behaviour does not react to progressive angle changes, but rather to trespassing a threshold. This is not satisfactory from a dynamic behaviour point of view.

The hypothesis of a threshold value for θ that depends on the instance being solved is supported by the observed behaviour of *LINEARDECREASE* and *LINEARINCREASE* in different instances (Fig. 5): When using MAB, there clearly exists an angle value that triggers a different search behaviour. Notice that the threshold value is different depending on whether we are increasing (Figs. 5 c and d) or decreasing (Figs. 5 a and b) θ . This could make difficult to find a compromise value, since it would depend on initial state of diversity.

Given that θ produces basically two extreme behaviours, we could devise a control strategy that simply alternates between minimum and maximum values in order to control both quality and diversity. This can be seen in experiments carried using the *ALWAYSMOVING* strategy, in which the MAB clearly reacts to angle changes in both *mean* and *max* impact computation criteria (Fig. 6).

4 Operators Management

Another relevant goal of this study is to investigate and understand how the control affects the Operator Selection during the search process. Operators lead

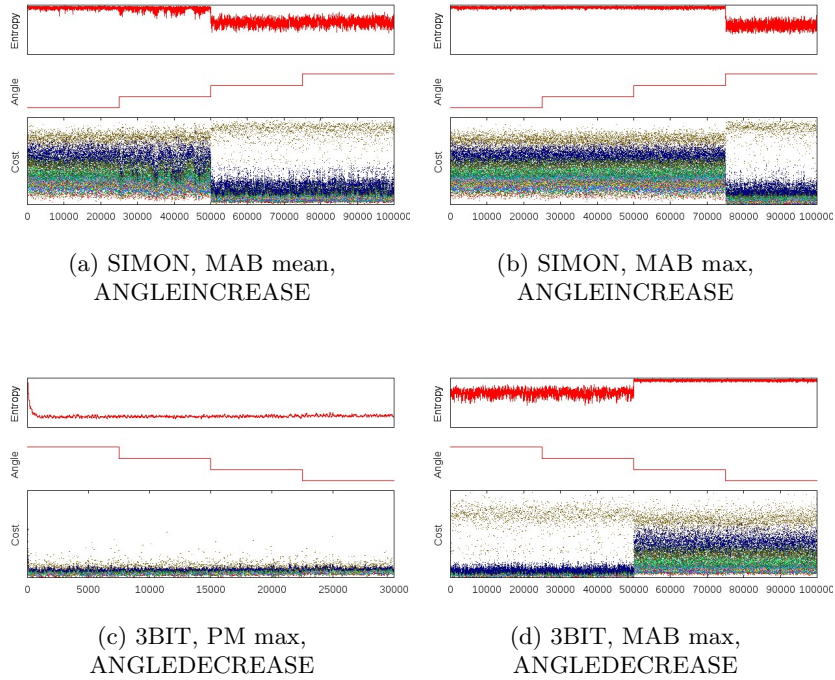


Fig. 4. Experiments with ANGLEINCREASE and ANGLEDECREASE

to the concentration of the population into specific search space areas (thus favouring intensification), others have it spread, thus favouring diversification.

To this goal, we have used a selection of 20 crossover operators from the set of more than 300 crossover operators defined in [13], combining four basic features.

We have grouped them according to their expected effect over population: **diversification:** 0035, 0015, 4455, 6011; **intensification:** 0011, 1111, 1122, 5011, 3332, 1134, 0022, 2352, 4454, 1224, 0013; **neutral:** 2455, 4335, 1125, 5035, 1335. Preliminary experiments have shown that, despite the above-mentioned expected effects, the behaviour induced by operators may be rather unclear at some point, except for 6011, which has shown excellent diversification capabilities in every condition. We will focus on the behaviour of the operator selection module w.r.t. its skill on dealing with those operators during diverse search epochs.

As a starting point, we show the operators frequency application when using a fixed angle. When favouring diversification ($\theta = 0$), both PM and MAB are able to identify the operator 6011 as the one able to insure the utmost diversification, using both *max* and *mean*. Notice that the probabilistic nature of PM can incur on the risk of applying a “wrong” operator if this has been applied in the early stage of the search. This is the case of the operator 1111, which is an operator favouring intensification, but that has been used by PM mean (Fig. 7a). MAB instead (Fig. 7b) does not show this shortcoming.

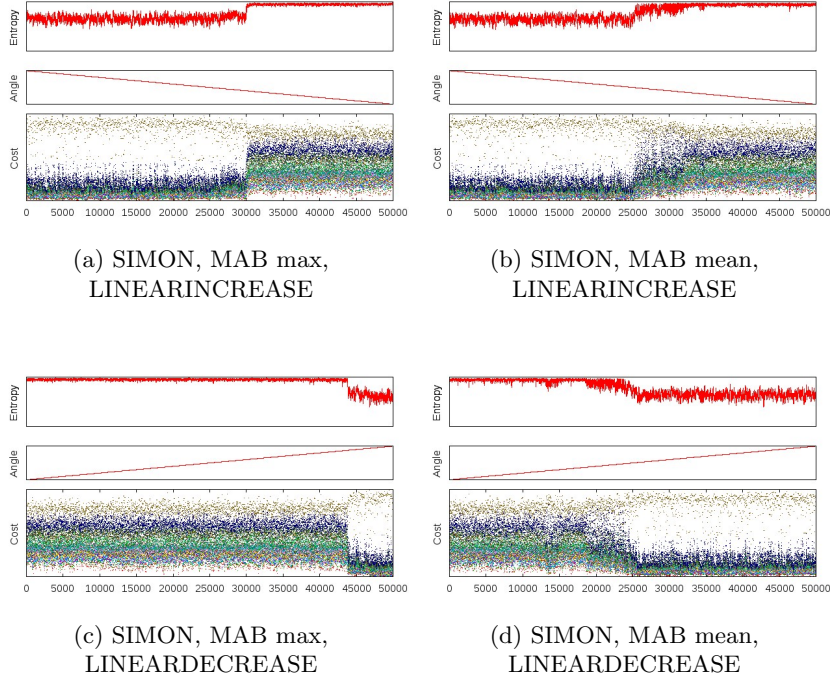


Fig. 5. Experiments with LINEARINCREASING and LINEARDECREASING

At this point, we have also to remark that it is obviously difficult to intensify the search when the population has reached a good mean quality, therefore intensification operators may then turn to another behaviour when being applied. In the following pictures, we are displaying, along with Cost, Entropy and Angle, the operator's relative application rate.

When imposing the maximum desired quality ($\theta = \frac{\pi}{2}$), MAB *mean* is able to identify the intensifier operator 1111 (Fig. 9a). The same happens with the MAB *max*, even with higher application magnitude (Fig. 9b). Note that when applying MAB, the operator selection process makes all operators to be selected at least a minimum number of times through the search: for this reason, operators which are known to not provide any intensification are deemed to worth a try. This is the case of the diversifying operator 6011, that has been detected and used in both *mean* and *max*. This is different with PM: When 6011 is not randomly selected at the early search stage, its not likely to be selected, neither using *mean* (Fig. 9c) nor *max* (Fig. 9d). Notice that also with PM the operator 1111 has been detected and used, but its magnitude is smaller than in the MAB case.

When dynamically changing the angle, the advantages of using MAB rather than PM become evident by analysing the operator frequency: in *ANGLEDECREASE*, PM turns not to be utterly unable to detect the operator 6011 as diversifying agent, in both *mean* and *max*. This is given by the fact that in the first search stages, when intensification is advocated, high rewards are assigned

to other operators. These high rewards make them to be applied also in further search stages (Fig 8e, only the *mean* case is reported). For each operator, several bars are plotted, each of which corresponds to a different search epoch, identified by a diverse angle value. MAB does not show this shortcoming, being able to correctly identify the operator 6011 as the best diversificator, in both *ANGLEINCREASE* and *ANGLEDECREASE* (Fig. 8f).

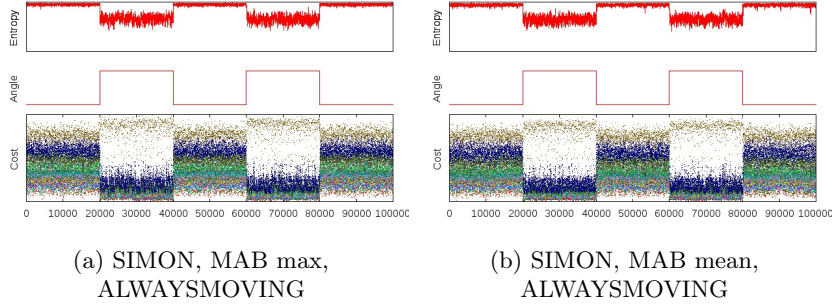


Fig. 6. SIMON, ALWAYSMOVING

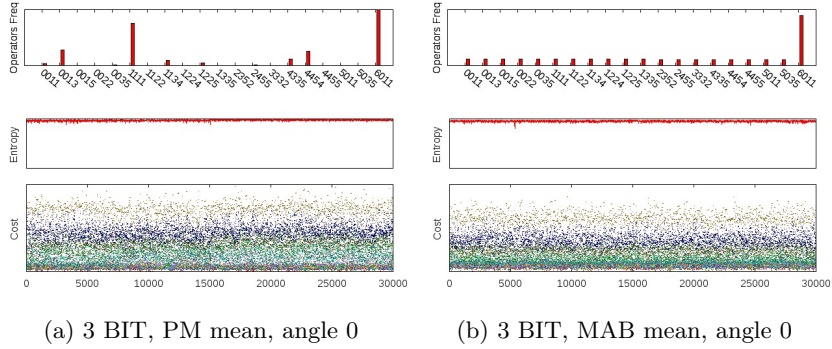


Fig. 7. Operator Frequency: angle 0

When applying *ANGLEINCREASE*, PM *max* identifies soon the operator 6011 as the one apt to diversify, applying it less and less as the angle value decreases (Fig. 8g). PM *mean* also discriminates among search epochs, but the operator 6011 is uniformly applied over the first two epochs (Fig. 8h).

Similar conclusions can be drawn when considering extreme and consecutive angle changes (*ALWAYSMOVING*): PM *max* is able to understand when to use operator 6011, when starting from both $\theta = \frac{\pi}{2}$ (Fig. 8a) and $\theta = 0$ (Fig. 8b). PM *mean* is instead incapable to self-adapt to these angle changes, and shows furthermore a huge application of operator 2455, which is a neutral one (Fig. 8c). MAB has instead no problem in self-adapting to extreme angle changes: using

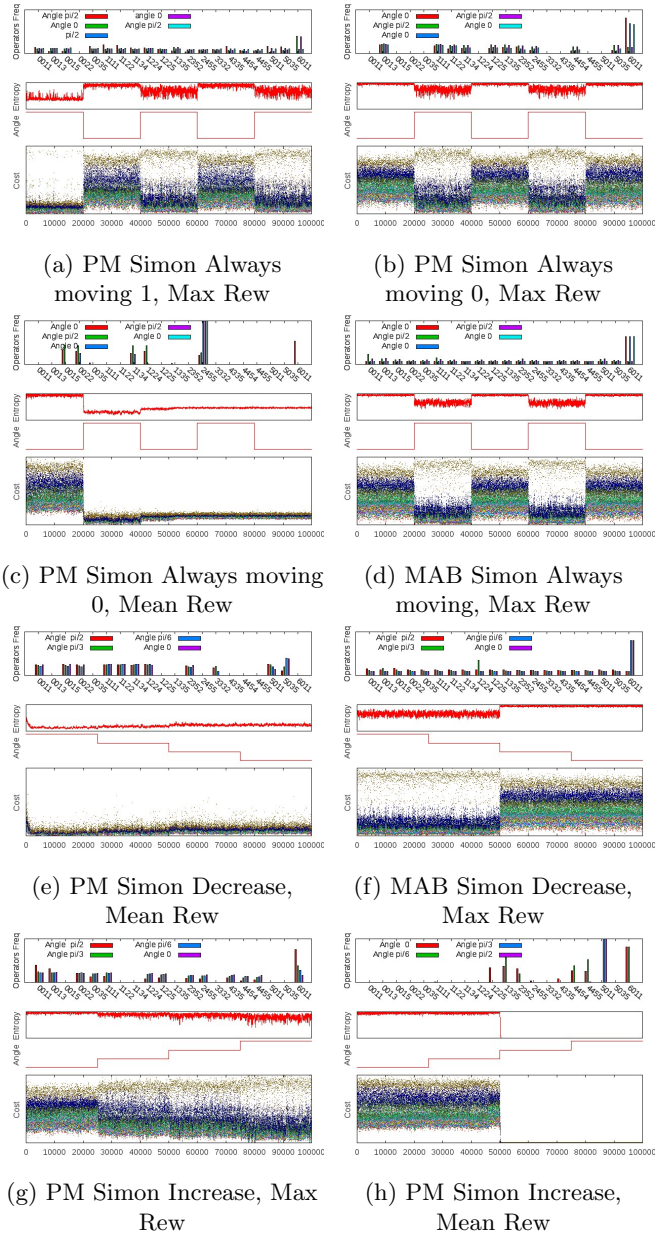


Fig. 8. Operator Frequency

both *mean* and *max*, and in both *ANGLEINCREASE* and *ANGLEDECREASE*, the behaviour can be summarised by Fig. 8d.

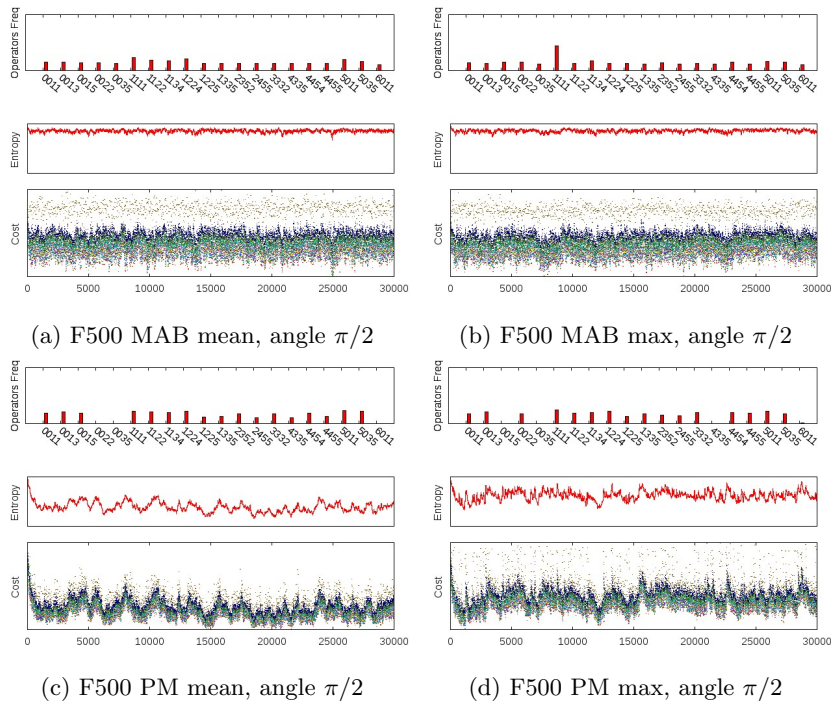


Fig. 9. Operator Frequency: Angle $\pi/2$

5 Conclusion

The general parameter control paradigm is related to the algorithm selection problem [16], which consists in selecting the most efficient algorithm for solving a problem at hand. Here, this selection is adaptive or dynamic and this general paradigm may open a new perspective for finding Free Lunch Theorems [15]. At least, there is a need of more formalized tools and criteria to compare algorithm's performances in term of reliability, adaptability and autonomy.

We have shown that, changing the angle value during search allows the controller to control the desired features. Since diversity depends not only on the value of θ but also in its previous state, extreme changes in θ seem to be the more straightforward way to control diversity. A finer diversity control could be done using two approaches: either to define how long these extrema values should be maintained, or to closely monitor the obtained diversity values and adjust θ according to the strategy schedule. Additional experiments are being carried out, which embed a mechanism to insert new operators on the fly. Further work will address the dynamic adjustment of the angle value based on the difference between desired and actual levels of intensification–diversification trade-off.

This work was supported by the French Chilean ECOS program C10E07

References

1. A. Biere, M. Heule, H.V. Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
2. M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proc. GECCO'02*: pages 11–18, M. Kaufmann, 2002.
3. L. Da Costa, Á. Fialho, M. Schoenauer, and M. Sebag. Adaptive operator selection with dynamic multi-armed bandits. In *Proc. GECCO'08*: pages 913–920. ACM Press, 2008.
4. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 3(2): pages 124–141, 1999.
5. A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, 2003.
6. A. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag. Extreme value based adaptive operator selection. In *PPSN X*, vol. 5199 of *LNCS*: pages 175–184. Springer, 2008.
7. W. Gong, Á. Fialho, and Z. Cai. Adaptive strategy selection in differential evolution. In *Proc. GECCO'10*: pages 409–416, ACM Press, 2010.
8. F. Hutter, H. H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In *Proc. AAAI '07*: pages 1152–1157, 2007.
9. A. Linhares and H. Yanasse. Search intensity versus search diversity: a false trade off? *Applied Intelligence*, 32(3): pages 279–291, 2010.
10. F. Lobo, C. Lima, and Z. Michalewicz, editors. *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*. Springer, 2007.
11. J. Maturana, A. Fialho, F. Saubion, M. Schoenauer, and M. Sebag. Compass and dynamic multi-armed bandits for adaptive operator selection. In *Proc. CEC'09, IEEE*: pages 365–372, 2009.
12. J. Maturana and F. Saubion. A compass to guide genetic algorithms. In *Proc. PPSN X*, vol. 5199 of *LNCS*: pages 256–265, Springer, 2008.
13. J. Maturana, F. Lardeux, and F. Saubion. Autonomous operator management for evolutionary algorithms. *Journal of Heuristics*, 16(6): pages 881–909, 2010.
14. V. Nannen, S. K. Smit, and A. E. Eiben. Costs and benefits of tuning parameters of evolutionary algorithms. In *Proc. PPSN X*, vol. 5199 of *LNCS*: pages 528–538, Springer, 2008.
15. R. Poli and M. Graff. There is a free lunch for hyper-heuristics, genetic programming and computer scientists. In *Genetic Programming*, vol. 5481 of *LNCS*: pages 195–207, 2009.
16. J. R. Rice. The algorithm selection problem. *Advances in Computers*, 15: pages 65–118, 1976.
17. D. Thierens. Adaptive Strategies for Operator Allocation. In *Parameter Setting in Evolutionary Algorithms*: pages 77–90. Springer Verlag, 2007.
18. F. Lardeux and F. Saubion and J.K. Hao. GASAT: A Genetic Local Search Algorithm for the Satisfiability Problem. In *Evolutionary Computation*, 14(2): pages 223–253. MIT Press, 2006.