Evolution Control Ensemble Models for Surrogate-Assisted Evolutionary Algorithms

Guillaume Briffoteaux*†§, Romain Ragonnet‡, Mohand Mezmaz*, Nouredine Melab†, Daniel Tuyttens*
*Mathematics and Operational Research Department (MARO), University of Mons, Belgium
Email: {guillaume.briffoteaux,mohand.mezmaz,daniel.tuyttens}@umons.ac.be
†Université de Lille, CNRS CRIStAL, Inria Lille, France
Email: nouredine.melab@univ-lille.fr
‡School of Public Health and Preventive Medicine, Monash University, Australia
Email: romain.ragonnet@monash.edu
§ Corresponding author

***Abstract*—Finding the trade-off between exploitation and exploration in a Surrogate-Assisted Evolutionary Algorithm is challenging as the focus on the landscape being optimized moves during the search. The balancing is mainly guided by Evolution Controls, that decide to simulate, predict or discard newly generated candidate solutions. Combining Evolution Controls in ensembles allows to regulate the degree of exploitation and exploration during the search. In this study, we propose ensemble strategies between multiple Evolution Controls in order to adapt the trade-off for each region scrutinized during the search. Experiments led on benchmark problems and on a real-world application of SARS-CoV-2 Transmission Control reveal that favoring exploration at the beginning of the search and favoring exploitation at the end of the search is beneficial in many cases.**

***Index Terms*—Surrogate-assisted Optimization, Evolution Control, Evolutionary Algorithm, Bayesian Optimization, Simulation, Massively Parallel Computing**

I. Introduction

Optimizing a function that depends on computationally expensive simulations is a tedious task. The primary difficulty is to gather information about the landscape to optimize. Evolutionary Algorithms (EAs) have shown to be effective when no hypothesis can be drawn about the function to be optimized [1]. EAs are nature-inspired meta-heuristics that maintain and evolve a population of candidate solutions according to the laws of living species evolution. Nevertheless, EAs require an elevated number of function evaluations, which is not convenient when a limited computational budget is imposed. To circumvent this limitation, surrogate models can be adopted. Surrogate models are built on historical data and aim to predict the expensive simulator outcome in timely fashion [2]. The surrogate model is used to replace the original function during the search.

In Surrogate-Assisted Evolutionary Algorithms (SAEAs), the balancing between Exploitation and Exploration (E&E) is realized at two levels. First, the calibration of Selection, Reproduction and Replacement operators of the EA [3], [4], [5] influences the trade-off. The second impact is due to a mechanism called Evolution Control (EC) [6] [7]. The EC chooses among the set of new solutions, which candidates to simulate, predict or discard. The most promising candidates are simulated, the less promising candidates are discarded while moderately promising candidates are only predicted.

On the one hand, the promise of a new candidate solution can be defined by the degree of uncertainty around the candidate predicted cost, as more uncertain candidates are more promising. In this case, exploration is enhanced as the unknown areas of the search space are visited. On the other hand, the promise can also be defined by the candidate predicted cost itself. Assuming a minimization problem, candidates that have a lower predicted cost are more promising. Exploitation is thus enhanced as the most favorable known area of the search space is scrutinized. The problem of balancing E&E can be seen as a bi-objective problem that can be treated using scalarization and Pareto-based methods.

The surrogate model is updated after each batch of simulations, hence exploration favors global prediction accuracy and exploitation favors local prediction accuracy.

The challenge addressed in this paper consists in balancing exploitation and exploration in a SAEA by designing ensembles of ECs that determine the convenient trade-off at each step of the search.

To take up this challenge, we propose three ensemble strategies between multiple existing ECs. In order to quantify the uncertainty of a candidate solution,

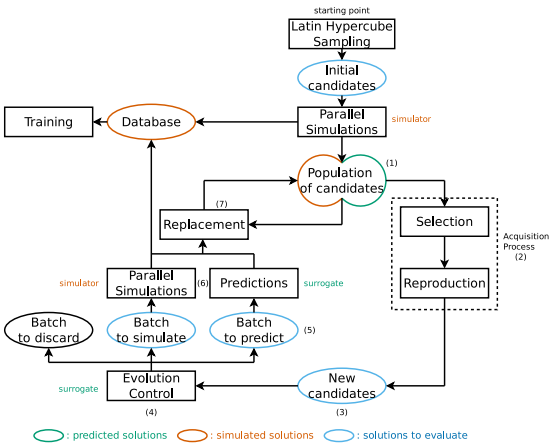

Fig. 1. Framework of Surrogate-Assisted Evolutionary Algorithm.

two measures are considered. The first measure is the distance from the candidate to the set of already known solutions lying in the search space. The second measure is the variance around the predicted cost provided by the surrogate. Since not all surrogates are able to provide such a quantity, Bayesian surrogate models are considered.

The next section gives a description of the framework of the SAEA used as well as the ECs and the surrogate models appearing recurrently in the literature. In Section III, the proposed ensemble strategies are presented. Section IV presents the experimental protocol and the associated results on benchmark problems producing different nuances of roughness in the search landscape. A problem of SARS-CoV-2 transmission control is explained and treated in Section V. Finally, conclusions and suggestions for future works are drawn in Section VI.

II. Background

A. Framework

The SAEA framework we propose in this study is presented in Figure 1.

New candidate solutions are generated by the Acquisition Process (2), made of a tournament selection of size 2, a SBX crossover and a polynomial mutation (reproduction operators) [1]. The crossover operator combines two parent solutions from the population (1) with the aim of creating a better offspring solution. The mutation operator is applied to the offspring in order to hinder its similarity with its parents. Crossover impacts exploitation while mutation impacts exploration. Subsequently, the set of newly generated solutions (3) is split into three batches (5) according to the EC-defined promising degree of each candidate. As stated in Section I, the EC (4) balances E&E by defining the promise. Finally, simulated and predicted candidates (6) apply for replacement (7) into the new population. It is ensured that both the best simulated and the best predicted candidates are inserted into the new population. The cycle of Selection (2), Reproduction (2), Evaluation (simulation and prediction) (6), and Replacement (7) is repeated until the computational budget is consumed.

The particularity of this framework is the possibility for the population to be composed of predicted solutions. In this configuration, a particular attention has to be paid to the surrogate prediction accuracy so that the search is not misled by poor predictions [8].

According to the classification proposed in [7], this framework is categorized as hybrid Direct/Indirect Fitness Replacement. Indeed, it can be classified as Direct Fitness Replacement because the surrogate plays the same role as the simulator. It could also be qualified as Indirect Fitness Replacement as the surrogate also helps the EC to discard least promising candidates.

B. Surrogate models

Three Bayesian surrogate models are integrated into the framework presented in the previous subsection in order to demonstrate the robustness of the ensemble-based ECs.

Kriging model: The Kriging surrogate model [9] [10] is a Gaussian Process widespread in surrogate-assisted optimization [11] [12] [13]. It is generally used in Efficient Global Optimization [14] [15] for its ability to estimate the variance around predictions.

Training the Kriging model consists in inverting a covariance matrix that represents the spatial correlation between all the already-simulated solutions. The inversion is cubic to the number of training samples and consequently makes the training expensive. Besides, incremental training is impossible.

MCDropout Bayesian Neural Network (MCD-BNN): A Bayesian Neural Network is an Neural Network for which a probability distribution is learnt for each weight of the network. Learning distributions allows to produce a variance around the prediction but training is a very time-consuming task [16]. To make the training easier, it has been proposed to sample multiple sub-networks from a network trained with Dropout [17], so as to compute the mean prediction and the variance around it. This technique is called MCDropout [18] [19].

Updating a MCD-BNN is faster than updating a Kriging model thanks to the incremental training capability. Moreover, the training is linear in the number of training samples. MCD-BNN has been used as surrogate model in SAEA in [8] [11].

Bayesian Linear Regressor Neural Network (BLR-NN): In [20], it is proposed to add a Bayesian Linear Regressor to the output layer of a Neural Network in order to decrease the computational cost of training a Gaussian Process and preserve the variance around the prediction. After the traditional training, each output neuron serves as a basis function to train the Bayesian Linear Regressor.

Training a BLR-NN is linear in the number of training samples and cubic to the number of basis functions. Incremental learning is possible for the underlying Neural Network but not for the Bayesian Linear Regressor that requires the inversion of a matrix made of the evaluations of the basis functions at the training points.

C. Evolution Controls

In this subsection, we present ECs already employed in SAEAs. These ECs are to be combined according to the ensemble strategies presented in the next section in order to meet the challenge of balancing E&E in SAEA.

Elementary ECs:

- ***rand*** randomly splits the set of newly generated candidates.
- ***bp*** considers as promising the best predicted candidates.
- ***dist*** considers as promising the candidates with highest distance from the set of already simulated solutions.
- ***var*** considers as promising the candidates with highest variance around the surrogate prediction.

The aforementioned elementary ECs have been compared in a MCD-BNN-assisted EA in [8]. As mentioned in Section I, *bp*-EC favors exploitation while *dist*-EC and *var*-EC favor exploration. To obtain a basic trade-off between E&E, the surrogate prediction and variance are often combined into scalarized or Pareto-based ECs.

Scalarized ECs:

- ***ei*** considers as promising the candidates with high Expected Improvement [14]:

$$EI(\boldsymbol{x}) = (y_{min} - \hat{y}(\boldsymbol{x}))\Phi\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x})\phi\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right)$$

- ***pi*** considers as promising the candidates with high Probability of Improvement [21]:

$$PI(\boldsymbol{x}) = \Phi\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right)$$

- ***lcb*** considers as promising the candidates with Low Confidence Bound:

$$LCB(\boldsymbol{x}) = \hat{y}(\boldsymbol{x}) - \lambda\hat{s}(\boldsymbol{x})$$

It is worth noting that EI and PI have been conceived for surrogates providing Gaussian predictions. While it is the case for the Kriging model, it is not the case for MCD-BNN and BLR-NN.

Pareto-based ECs:

- ***bo-max*** considers as promising the candidates with low non-dominated rank regarding the minimization of the predicted cost and the maximization of uncertainty [8].
- ***bo-min*** considers as promising the candidates with low non-dominated rank regarding the minimization of the predicted cost and the minimization of uncertainty.

In Pareto-based ECs, uncertainty could either be the distance to the set of already simulated solutions or the variance around the surrogate prediction.

Scalarized and Pareto-based ECs have been compared mainly in the Efficient Global Optimization framework [22] [23]. They also have proven to perform well on numerous real-world applications such as infectious disease transmission control [8] [11] and optimization of fused magnesium furnaces [24].

D. Related works

Few studies have been dedicated to balance E&E by designing ensembles of ECs. In [25], multiple rules are used to select multiple new candidate solutions to be simulated. Although multiple ECs are considered, the trade-off between exploitation and exploration is constant during the search. In [26], a scalarized EC adapts the balancing during the search according to the budget consumption. Contrary to our point of view, the strategy proposed in [26] does not consider the event of poor prediction accuracy. Indeed, the search begins by relying only on the surrogate predicted cost.

III. Proposed approaches

A. Dynamic ECs

Let

- $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be n ECs from the categories defined in the previous section ($n \in \mathbb{N}^* \setminus \{1\}$).
- $0 < p_1 < \cdots < p_{n-1} < 1$ be user-defined parameters.

The dynamic ensemble strategy, denoted ***dyn***, consists in switching from $\mathcal{E}_i$ to $\mathcal{E}_{i+1}$ at the end of the first cycle for which $100p_i\%$ of the budget has been consumed. ECs numbering is relevant as it defines the order in which the ECs will handle the search.

B. Adaptive ECs

The adaptive ensemble strategy, denoted ***ada***, is a reward-based mechanism proposed to automatically switch from one EC to another during the search. It advantageously removes the need for the user to define the parameters $p_1, \ldots, p_{n-1}$.

Let

- $\boldsymbol{x}$ be a candidate solution.
- $y(\boldsymbol{x})$ be the simulated cost.
- $\hat{y}(\boldsymbol{x})$ be the predicted cost.
- $e(\boldsymbol{x}) = (y(\boldsymbol{x}) - \hat{y}(\boldsymbol{x}))^2$ be the surrogate prediction error.
- $t_i = \frac{1}{n_{sim}} \sum_{j=1}^{n_{sim}} e(\boldsymbol{x}^{(i)(j)})$ be the average error committed during cycle i.
- $n_{sim} \in \mathbb{N}^*$ be the number of simulated solutions per cycle.
- $r(e)$ be a reward function.
- $\mathcal{E}^*$ be the current active EC.

When $\mathcal{E}^*$ decides to simulate $\boldsymbol{x}$ and $e(\boldsymbol{x}) < t_i$ holds then ECs that would have simulated $\boldsymbol{x}$ are rewarded by $r(e(\boldsymbol{x}))$ and ECs that would have predicted or discarded $\boldsymbol{x}$ are penalized by the same amount. Reversely, when $\mathcal{E}^*$ decides to simulate $\boldsymbol{x}$ and $e(\boldsymbol{x}) \geqslant t_i$ holds then ECs that would have simulated $\boldsymbol{x}$ are penalized by $r(e(\boldsymbol{x}))$ and ECs that would have predicted or discarded $\boldsymbol{x}$ are rewarded by the same amount. At the end of the cycle, the most-rewarded EC becomes active. Three reward functions are considered in the experiments reported in the next section:

$$r_{sig}(e) = \begin{cases} sign(t_i - e) \text{ if } e \neq 0 \\ 1 \text{ otherwise} \end{cases}$$

$$r_{lin}(e) = t_i - e$$

$$r_{tanh}(e) = tanh(t_i - e)$$

C. Committee of ECs

Inspired by active learning [27], the committee ensemble strategy, denoted ***com***, assumes that each of the available ECs $\mathcal{E}_1, \ldots, \mathcal{E}_n$ votes for each new candidate. Promising candidates are the ones collecting the greater number of votes.

IV. Experimentation on benchmark problems

A. Protocol and algorithms configuration

We propose to compare 31 ensemble-based and non-ensemble-based ECs on three benchmark functions [8] defined on $[0, 1]^6$:

- 6-D multi-modal Schwefel
- 6-D noisy Rastrigin
- 6-D valley-shaped Rosenbrock

TABLE I
Experimental parameters for benchmark problems

Budget	512+32 real evaluations
Population size	32
Crossover probability	0.9
Mutation probability	0.001
Number of children per cycle	64
Number of simulation per cycle	$n_{sim} = 8$
Number of prediction per cycle	24
Number of discarding per cycle	32

The experimental parameters are listed in Table I. The parameters of the EA have been determined by a grid search considering 36 configurations. For each configuration, the EA is run 10 independent times. The parameters related to the number of evaluations per cycle have been fixed arbitrarily.

Architecture and hyper-parameters of MCD-BNN have been determined thanks to a specific EA similar to the EA presented in [28]. In addition to the modifiers presented in [28] we added operators to modify the dropout probability, the weight decay coefficient and the weights initializer. A polynomial mutation operator is also considered to modify the learning rate. The specific EA is run for 30 minutes on 1 node made of 16 AMD EPYC 7301 CPU cores where each core generates and trains a new MCD-BNN. The population size is set to 32 and the generated MCD-BNNs are trained during 100 epochs on a training set made of 288 evaluations and validated on a validation set made of 2048 evaluations. The best retained MCD-BNNs are presented in Figure 2.

No specific efforts are engaged into Kriging and BLR-NN calibration. The default parameters of the pyKriging library [29] and the pybnn library [20] are considered for Kriging and BLR-NN respectively.

B. Results on benchmark functions

The 31 ECs considered in the experiments are listed in Table II. For dynamic ECs, $p_1 = 0.5$ when 2 ECs are involved except for *dyn-sy-75* and *dyn-dy-75* where $p_1 = 0.75$. When 3 ECs are involved $p_1 = 0.25$ and $p_2 = 0.75$.

The results obtained thanks to the MCD-BNN, BLR-NN and Kriging surrogate models are reported in Table III. According to these tables, it can be concluded that dynamic ensemble-based ECs perform better in the top 10 ECs lists. In particular, *dyn-dy-75* is the best EC in 5 of the 9 cases treated and *dyn-dby* is the best EC when optimizing the Schwefel function with an MCD-BNN surrogate model. Pareto-based ECs also perform consistently as they appear in all the top 10. Besides, *bio-yd-max* is the best EC in 2 cases. Adaptive and committee ECs are less successful than dynamic or Pareto-based ECs

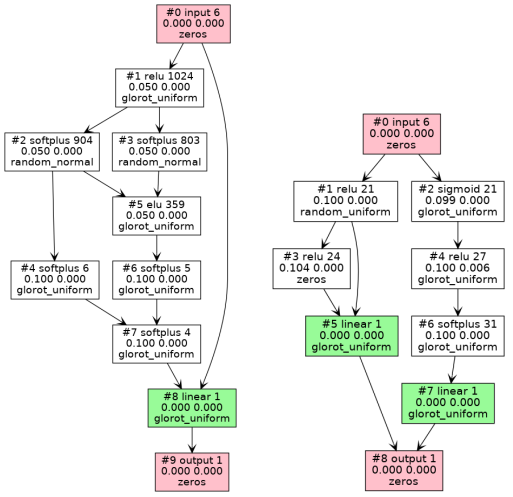

Fig. 2. MCD-BNN architecture and hyper-parameters for Schwefel and Rastrigin (left) and Rosenbrock (right) benchmark problems. Each box represents a layer. **First line** gives the layer identifier, the activation function and the number of neurons. **Second line** gives the dropout probability and the weight decay coefficient. **Third line** gives the weights initializer.

TABLE II
31 Evolution Controls considered in the benchmark experiments. For Pareto-based and ensemble-based ECs, **y** refers to *bp*, **s** refers to *var*, **d** refers to *dist* and **b** refers to *bio-max*. For Dynamic and Adaptive ECs, the first letter indicates the starting EC.

Elementary ECs *rand*, *bp*, *dist*, *var*
Scalarized ECs *pi*, *lcb*, *ei*
Pareto-based ECs *bio-ys-max*, *bio-ys-min*, *bio-yd-max*, *bio-yd-min*
Dynamic ECs *dyn-sy*, *dyn-ys*, *dyn-sy-75*, *dyn-sby*, *dyn-ybs*, *dyn-dy*, *dyn-yd*, *dyn-dy-75*, *dyn-dby*, *dyn-ybd*
Adaptive ECs *ada-ys-tanh*, *ada-ys-lin*, *ada-ys-sig*, *ada-syb-lin*, *ada-yd-tanh*, *ada-yd-lin*, *ada-yd-sig*, *ada-dyb-lin*
Committee ECs *com-ysb*, *com-ydb*

but outperform the scalarized ECs in the majority of cases.

Good results provided by *dyn-dy-75* and *dyn-dby* in Schwefel and Rastrigin indicate that exploration should be favored at the beginning of the search and exploitation should be favored at the end of the search, at least for rough landscapes. Nevertheless, when the surrogate provides very good prediction, the *bp* EC should be used as indicated by the results displayed in the last row of Table III on the Rosenbrock function.

In almost all cases, the distance measure seems to provide better uncertainty information about the surrogate prediction than the variance. This can be explained by either the calibration of the surrogate or the composition of the training set. Indeed for MCD-BNN, the parameter controlling the number of sub-networks used to compute the prediction variance has not been considered during calibration. For BLR-NN, no calibration is considered on the neural network architecture. For Kriging, the training set size is restricted and does not include all the available simulations.

V. Application to SARS-CoV-2 transmission control

A. Problem description

Awaiting for a vaccine against SARS-CoV-2, governments are resorting to social mixing reduction measures in order to limit the impact of COVID-19. The SARS-CoV-2 transmission control exercise proposed in this study consists in minimizing the number of deaths by determining the vector of 16 mitigation factors to apply to the 16 age-categories of the Spanish population. The mitigation vector is constrained by herd immunity realization after a social mixing mitigation period of 12 months.

The optimization problem can be re-formalized as a single real objective minimization problem by applying a penalty of $8,000,000,000$ to the number of deaths when herd immunity is not reached. Consequently, the problem treated consists in finding $\boldsymbol{x} \in [0,1]^{16}$ such that

$$\boldsymbol{x} = argmin \tilde{f}(\boldsymbol{x})$$

where

$$\tilde{f}(\boldsymbol{x}) = \begin{cases} f_1(\boldsymbol{x}) \text{ if } f_2(\boldsymbol{x}) = 1 \\ f_1(\boldsymbol{x}) + 8,000,000,000 \text{ if } f_2(\boldsymbol{x}) = 0 \end{cases}$$

f_1 and f_2 are the number of deaths and the boolean variable indicating whether herd immunity has been reached respectively. Both quantities are provided by a simulator modelling epidemic transmission and built using data from the World Health Organization. The simulator implements a deterministic compartmental model and solves the differential equations governing the pathogenic agent transmission [30]. The code used to implement this model is publicly available on Github [31]. The simulator has been used in numerous studies related to Tuberculosis transmission control [11] [8] [32] [33]. The social mixing data are integrated into the model through the age-specific contact matrices previously published in [34].

TABLE III
Top 10 ECs according to the mean best cost (50 independent replications) considering MCD-BNN (top), BLR-NN (middle) and Kriging (bottom) surrogate models.
The best EC is displayed at the top of the column for each surrogate and each benchmark function.
Pareto-based ECs appear in green, Dynamic ECs in orange, Adaptive ECs in blue and Committee ECs in purple.

	Schwefel-6D		Rastrigin-6D		Rosenbrock-6D	
	EC	mean[stdev]	EC	mean[stdev]	EC	mean[stdev]
MCD-BNN	***dyn-dby***	**94.6**[92.7]	***dist***	**7.7**[2.6]	***bio-yd-max***	**549.1**[**590.7**]
	bio-yd-max	116.0[116.4]	*dyn-dy-75*	8.3[**2.5**]	*dyn-dby*	640.8[999.2]
	dyn-dy-75	117.3[98.7]	*var*	9.6[5.1]	*ada-dyb*	686.8[844.8]
	dyn-dy	128.3[**84.5**]	*rand*	10.2[6.3]	*dist*	759.3[1406.5]
	dist	134.9[113.5]	*com-ydb*	10.5[4.3]	*com-ydb*	793.2[1131.1]
	ada-dyb	186.4[138.4]	*dyn-sy-75*	10.8[7.3]	*ada-yd-tanh*	855.5[1025.0]
	com-ydb	189.4[176.2]	*dyn-sy*	11.3[5.5]	*ada-yd-sig*	883.1[1498.2]
	ada-yd-lin	203.5[177.5]	*bio-yd-max*	11.4[3.4]	*dyn-yd*	1000.3[1642.5]
	ada-yd-tanh	234.4[172.2]	*dyn-dby*	11.6[4.3]	*dyn-ybd*	1036.5[1623.8]
	ada-yd-sig	240.1[156.4]	*dyn-dy*	12.3[5.2]	*dyn-dy-75*	1036.6[1760.8]
BLR-NN	***dyn-dy-75***	**168.2**[137.9]	***dyn-dy-75***	**7.16**[4.2]	***bio-yd-max***	**627.2**[**399.1**]
	dyn-dy	168.6[**126.3**]	*dyn-sy-75*	7.70[4.3]	*dist*	696.7[670.2]
	dyn-dby	177.1[141.8]	*dyn-dby*	7.72[**3.8**]	*ada-yd-tanh*	782.0[600.8]
	dist	203.8[148.8]	*dyn-dy*	7.83[4.7]	*dyn-dy-75*	827.4[869.2]
	dyn-sby	336.4[219.9]	*dyn-sy*	8.84[4.1]	*ada-yd-lin*	971.4[893.9]
	var	344.1[168.3]	*dyn-sby*	9.47[5.2]	*dyn-dy*	1012.3[1086.3]
	dyn-sy	360.7[221.0]	*var*	10.23[4.5]	*ada-dyb*	1017.0[909.3]
	dyn-sy-75	362.5[187.8]	*dist*	10.28[4.5]	*com-ydb*	1033.9[1350.3]
	bio-yd-max	366.9[206.7]	*bio-ys-max*	11.01[5.7]	*dyn-dby*	1056.8[1262.2]
	ada-yd-lin	413.9[201.6]	*com-ysb*	11.53[5.7]	*dyn-sby*	1060.3[1000.3]
Kriging	***dyn-dy-75***	**137.0**[148.9]	***dyn-dy-75***	**5.31**[**2.8**]	***bp***	**90.3**[**111.8**]
	dyn-dy	150.4[144.8]	*dyn-dy*	5.67[3.5]	*dyn-yd*	105.5[150.4]
	dist	166.7[160.9]	*dyn-sy-75*	6.86[3.4]	*dyn-ybd*	114.4[176.8]
	dyn-dby	180.2[150.2]	*dist*	7.92[3.1]	*ada-yd-tanh*	140.5[220.6]
	var	180.7[**138.5**]	*dyn-dby*	8.07[4.5]	*ada-ys-sig*	148.7[143.0]
	dyn-sy	201.5[185.8]	*var*	8.13[3.7]	*dyn-ys*	158.5[224.8]
	dyn-sy-75	215.3[160.8]	*dyn-sy*	8.82[4.6]	*ada-yd-sig*	162.9[186.4]
	dyn-sby	219.8[178.2]	*dyn-sby*	8.85[4.3]	*bio-yd-min*	163.2[213.6]
	com-ydb	385.6[245.2]	*bio-yd-max*	10.58[5.2]	*lcb*	170.1[213.1]
	bio-yd-max	387.4[267.4]	*com-ydb*	12.05[5.8]	*dyn-ybs*	193.6[201.5]

B. Protocol and algorithms configuration

The experimental parameters are listed in Table IV. The use of parallelism allows to increase the number of simulations in a limited period of time. The EA is calibrated with parameters similar to those used in [11] and [8]. The parameters related to the number of evaluations per cycle have been determined by multiple trials. For the moment, only MCD-BNN is considered as surrogate model for this experiment. Kriging and BLR-NN will be considered in future experiments.

TABLE IV
Experimental parameters for SARS-CoV-2 transmission control

Budget	30 minutes
Number of node	1
Cores per node	18 Intel Xeon Gold 5220
Simulation duration	12 seconds on 1 core
Population size	126
Number of children per cycle	252
Number of simulation per cycle	$n_{sim} = 96$
Number of prediction per cycle	30
Number of discarding per cycle	126
Number of layers	2
Activation function	tanh
Number of neurons per layer	256
Dropout probability	0.05
Training	incremental
Training set	252 last simulations

C. Results

For this experiment, 9 ECs are considered as well as the EA without surrogate, denoted *no_surrogate* in the following. For each EC, the EA is run 25 times and the mean over the best numbers of deaths found at the end of the searches is computed and reported in Table V.

By the results reported in Table V, only the dynamic EC *dyn-dby* successfully outperforms the results provided by the EA without surrogate. In this particular problem, favoring exploration at the beginning, exploration at the end and a trade-off between both at the middle of the search is beneficial.

The distance measure seems to produce better uncertainty information than prediction variance. Similarly to the benchmark experiments, no surrogate calibration is dedicated to obtain a reliable prediction variance.

TABLE V
ECs classification according to the mean best number of deaths (25 independent replications) on the SARS-CoV-2 transmission control problem.

EC	mean number of deaths
dyn-dby	13936
no_surrogate	15025
bio-yd-max	16073
dyn-dy-25	16474
dyn-sby	16586
dist	17229
dyn-sy-25	17579
var	19262
bio-ys-max	20373
bp	23265

VI. Conclusions and Perspectives

In this paper, we tackle the challenge of balancing exploitation and exploration in a Surrogate-Assisted Evolutionary Algorithm by designing ensemble strategies of Evolution Controls. Evolution Controls are mechanisms dedicated to determine whether to simulate, predict or discard new candidate solutions. Each of them consequently carries its own degree of exploitation and exploration.

According to the experimental outcomes, the dynamic ensemble strategy that favors exploration at the beginning of the search and exploitation at the end of the search appears to be the best among the set of Evolution Controls considered. In particular, dynamic ensemble strategies outperform famous scalarized Evolution Controls from the literature such as the so-called Expected Improvement. Moreover, this strategy successfully improves the results of a traditional Evolutionary Algorithm on a problem of SARS-CoV-2 Transmission Control.

During the search, the Evolutionary Algorithm dynamically identifies a favorable region that tends to become narrower and narrower. From the surrogate prediction accuracy's point of view, it seems convenient to promote exploration at the beginning of the search and, on the contrary, to promote exploitation at the end of the search. By the results, the dynamic ensemble strategy we proposed successfully switches from exploration-promotion to exploitation-promotion during the search.

The main drawback of the dynamic ensemble strategy is the need for the user to define the switching frequency between the Evolution Controls involved in the ensemble. To alleviate this drawback, an adaptive ensemble strategy based on rewards and a committee ensemble strategy were proposed. Development of such parameter-free strategies remains challenging as their versions proposed in this study do not improve the experimental results generated by the dynamic ensemble strategy. Our future efforts will be dedicated to improve the adaptive ensemble by incorporating the uncertainty information into the surrogate prediction error. The committee ensemble strategy will be reviewed by integrating weights to mitigate or amplify the importance of each Evolution Control. An automatic mechanism will be designed to update the weights during the search.

Acknowledgment

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

References

[1] E. G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.

[2] M. Asher, B. Croke, A.J. Jakeman, and L. Peeters. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51, 07 2015.

[3] M. Epitropakis, V. Plagianakos, and M. Vrahatis. Balancing the exploration and exploitation capabilities of the differential evolution algorithm. pages 2686–2693, 06 2008.

[4] H. Zhang, J. Sun, T. Liu, K. Zhang, and Q. Zhang. Balancing exploration and exploitation in multiobjective evolutionary optimization. *Information Sciences*, 497:129 – 148, 2019.

[5] M. Črepinšek, S.H. Liu, and M. Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.*, 45(3), July 2013.

[6] A. Diaz-Manriquez, G. Toscano Pulido, J. Barron-Zambrano, and E. Tello-Leal. A review of surrogate assisted multiobjective evolutionary algorithms. *Computational Intelligence and Neuroscience*, 2016:1–14, 06 2016.

[7] L. Shi and K. Rasheed. *A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms*, pages 3–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[8] G. Briffoteaux, R. Ragonnet, M. Mezmaz, N. Melab, and D. Tuyttens. Evolution control for parallel ann-assisted simulation-based optimization application to tuberculosis transmission control. *Future Generation Computer Systems*, 113:454 – 467, 2020.

[9] M. Georges. Principles of geostatistics. econ geol (lancaster). *Economic Geology*, 1963.

[10] D.G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.

[11] G. Briffoteaux, M. Gobert, R. Ragonnet, J. Gmys, M. Mezmaz, N. Melab, and D. Tuyttens. Parallel surrogate-assisted optimization: Batched bayesian neural network-assisted ga versus q-ego. *Swarm and Evolutionary Computation*, 57:100717, 2020.

[12] D. Ginsbourger, R. Le Riche, and L. Carraro. A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes. Technical report, March 2008.

[13] D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, Springer series in Evolutionary Learning and Optimization, pages 131–162. springer, 2010.

[14] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Dec 1998.

[15] J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[16] A. Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[18] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[19] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[20] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M.M.A. Patwary, M. Prabhat, and R. Adams. Scalable bayesian optimization using deep neural networks. *Statistics*, 02 2015.

[21] H.J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Fluids Engineering*, 86(1):97–106, 03 1964.

[22] B. Bischl, S. Wessing, N. Bauer, K. Friedrichs, and C. Weihs. Moi-mbo: Multiobjective infill for parallel model-based optimization. In Panos M. Pardalos, Mauricio G.C. Resende, Chrysafis Vogiatzis, and Jose L. Walteros, editors, *Learning and Intelligent Optimization*, pages 173–186, Cham, 2014. Springer International Publishing.

[23] J. Liu, Z.H. Han, and W. Song. Comparison of infill sampling criteria in kriging-based aerodynamic optimization. *28th Congress of the International Council of the Aeronautical Sciences 2012, ICAS 2012*, 2:1625–1634, 01 2012.

[24] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 2018.

[25] T. Akhtar and C. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, 64, 01 2016.

[26] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer. An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Information Sciences*, 519:317 – 331, 2020.

[27] J. O'Neill, S. Delany, and B. MacNamee. *Model-Free and Model-Based Active Learning for Regression*, volume 513, pages 375–386. 01 2017.

[28] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E.P. Xing. Neural architecture search with bayesian optimisation and optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2016–2025. Curran Associates, Inc., 2018.

[29] C. Paulson and G. Ragkousis. pykriging: A python kriging toolkit, July 2015.

[30] J.M.C Trauer, R. Ragonnet, T.N. Doan, and E.S. McBryde. Modular programming for tuberculosis control, the "autumn" platform. *BMC Infectious Diseases*, 17(1):546, Aug 2017.

[31] J.M. Trauer, R. Ragonnet, M. Segal, and E. McBryde M. Abayawardana. Autumn github repository. https://github.com/monash-emu/AuTuMN/, 2019.

[32] J.M.C. Trauer, J.T. Denholm, S. Waseem, R. Ragonnet, and E.S. McBryde. Scenario Analysis for Programmatic Tuberculosis Control in Western Province, Papua New Guinea. *American Journal of Epidemiology*, 183(12):1138–1148, 05 2016.

[33] R. Ragonnet, F. Underwood, T. Doan, E. Rafai, J.M.C. Trauer, and E.S. McBryde. Strategic planning for tuberculosis control in the republic of fiji. *Tropical Medicine and Infectious Disease*, 4(2), 2019.

[34] K. Prem, A. Cook, and M. Jit. Projecting social contact matrices in 152 countries using contact surveys and demographic data. *PLOS Computational Biology*, 13:e1005697, 09 2017.