Hybrid Acquisition Processes in Surrogate-based Optimization. Application to Covid-19 Contact Reduction

Guillaume Briffoteaux, Nouredine Melab, Mohand Mezmaz, Daniel Tuyttens

Hybrid Acquisition Processes in Surrogate-based Optimization. Application to Covid-19 Contact Reduction

Guillaume Briffoteaux[1,2,3], Nouredine Melab[2], Mohand Mezmaz[1] and Daniel Tuyttens[1]

[1] Mathematics and Operational Research Department, University of Mons, Belgium
{guillaume.briffoteaux,mohand.mezmaz,daniel.tuyttens}@umons.ac.be
[2] University of Lille, Inria, UMR 9189 - CRIStAL, France
nouredine.melab@univ-lille.fr
[3] corresponding author

Abstract. Parallel Surrogate-Assisted Evolutionary Algorithms (P-SAEAs) are based on surrogate-informed reproduction operators to propose new candidates to solve computationally expensive optimization problems. Differently, Parallel Surrogate-Driven Algorithms (P-SDAs) rely on the optimization of a surrogate-informed metric of promisingness to acquire new solutions. The former are promoted to deal with moderately computationally expensive problems while the latter are put forward on very costly problems. This paper investigates the design of hybrid strategies combining the acquisition processes of both P-SAEAs and P-SDAs to retain the best of both categories of methods. The objective is to reach robustness with respect to the computational budgets and parallel scalability.

1 Introduction

To solve black-box expensive optimization problems where the objective function is computationally costly to evaluate, Parallel Surrogate-Based Optimization Algorithms (P-SBOAs) are built by leveraging parallel computing and machine learning. Two categories of P-SBOAs arise: Parallel Surrogate-Assisted Evolutionary Algorithms (P-SAEAs) and Parallel Surrogate-Driven Algorithms (P-SDAs). Both families of algorithms differ by their Acquisition Process (AP), the mechanism in charge of suggesting new promising candidate solutions. On the one hand, the AP from P-SDAs aims at quickly reaching good solutions, consequently providing a strong improvement that faints after few cycles. On the other hand, the AP from P-SAEAs is more exploratory, thus making the improvement slighter but more durable. In a previous study, we observed that P-SAEAs are generally recommended in the context of moderately expensive problems and that P-SDAs are usually preferred to deal with very expensive problems [1]. Moderately expensive problems are characterized by a budget greater than 1000 simulations or a simulation lasting less than 5 minutes. In this article, we present

a hybrid method retaining the best of both P-SAEAs and P-SDAs. A strategy previously proposed in [2] already relies on hybridization of APs but shows a serious limitation regarding parallel scalability. The challenge we address in this study is to come up with a strategy that is robust with respect to the computational budget allocated to the search and that scales with multiple computing cores.

For a moderately expensive objective function, the computational budget may allow a meaningful number of expensive evaluations. Consequently, the database of exactly evaluated solutions may grow significantly enough for the surrogate training to become non-negligible in terms of computations. In this situation, it is not convenient to express the computational budget only as a limited number of objective function evaluations as it is commonly done in the field of surrogate-based optimization [3–7]. Indeed, the cost related to surrogate training would be hidden. Instead, we chose to define the budget as a limited duration on a limited number of computing cores.

Benefiting from numerous computing cores raises concerns to the performance of P-SBOAs. The AP emphasized in [8] outputs $q = 4$ new candidates per iteration, consequently triggering $q = 4$ parallel evaluations. The low value attributed to q points the difficulty of conserving a relevant degree of diversity when numerous new candidates are sampled at once, thus preventing to efficiently leverage more computing cores. In [2], the proposed Surrogate Model Based Optimization + Evolutionary Algorithm (SMBO+EA) demonstrates a superiority compared to state-of-the-art P-SDAs for a number of computing cores $n_{cores} < 10$. However, this hybrid method performs similarly to a surrogate-free parallel evolutionary algorithm for $n_{cores} \geqslant 10$.

The main contribution of this paper is the HSAP strategy (Hybrid Successive Acquisition Processes) that employs successively two APs during the search, thus providing robustness with respect to the computational budgets and efficient use of multiple computing cores. The numerical experiments consider an objective function based on a black-box simulator of Covid-19 transmission. The related moderately expensive real-world optimization problem consists in finding the best contact reduction strategy to minimize the number of deaths while attaining herd immunity.

The paper is organized as follows. In Section 2 a background on surrogate-based optimization is proposed and the Covid-19-related problem is presented in Section 3. The new algorithms based on hybrid APs are dissected in Section 4 and are compared with state-of-the-art methods through numerical experiments whose outcomes are reported in Section 5. Finally, conclusions and future research directions are pointed out in Section 6.

2 Background on Surrogate-based optimization

The surrogate-model is based on a machine learning algorithm for interpolation or regression in order to imitate the expensive objective function. The two models

considered in this study are a Gaussian Process with a Radial Basis Function kernel (GP_RBF) and a Bayesian Neural Network approximated by Monte-Carlo Dropout (BNN_MCD).

The general idea of the GP_RBF is to model the influence of one point $\boldsymbol{x}$ on the prediction at another point $\boldsymbol{x'}$ by the kernel function defined by:

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x'}||^2}{2s^2}\right) \tag{1}$$

where σ and s are hyper-parameters called the scale and the length scale respectively. By considering the observations as random variables and by applying the Bayes theorem, the GP_RBF provides a prediction $\hat{f}(\boldsymbol{x'})$ and a predictive standard deviation $\hat{s}(\boldsymbol{x'})$ at an unknown point $\boldsymbol{x'}$. The operation of training a GP is cubic to the number of training samples. More thorough details about GPs are given in [9].

The main principle behind BNN_MCD is to sample n_{sub} sub-networks $\hat{f}_i$ from a global artificial neural network and to use the n_{sub} predictions to compute an average prediction and a standard deviation:

$$\hat{f}(\boldsymbol{x'}) = \frac{1}{n_{sub}} \sum_{i=1}^{n_{sub}} \hat{f}_i(\boldsymbol{x'}) \qquad \hat{s}(\boldsymbol{x'}) = \sqrt{\frac{1}{n_{sub}} \sum_{i=1}^{n_{sub}} (\hat{f}_i(\boldsymbol{x'}) - \hat{f}(\boldsymbol{x'}))^2} \tag{2}$$

The sub-networks are sampled by randomly deactivating neurons in the global network. It has been proven in [10] that this technique amounts to perform an approximated Bayesian training. The operation of training a BNN_MCD is linear to the number of training samples.

The two categories of P-SBOAs, namely P-SAEAs and P-SDAs, differ by the coupling between the surrogate-model and the optimizer [1]. In P-SAEAs, the surrogate is attached to the Evolutionary Algorithm (EA) by means of an Evolution Control (EC) that defines the promisingness of new candidate solutions. The EA carries out the search by evolving a population of candidates through the stages of selection, reproduction and replacement. The surrogate is introduced at any stage to replace the expensive objective function [11]. In P-SDAs, a metric of promisingness called the Infill Criterion (IC) is optimized to locate new potential candidates [12]. The IC and the EC are based on the predictive objective value ($\hat{f}$) and/or predictive standard deviation ($\hat{s}$) delivered by the surrogate. The difference between the two concepts of IC and EC is thin. The EC is defined as a comparison operator while the IC is a real-valued metric. It is straightforward to convert an IC into an EC and a dedicated EA can be set up to optimize an IC corresponding to a given EC (by basing the selection and replacement on the EC) [13].

To build the new hybrid acquisition processes, we rely on two actual surrogate-optimizer couplings. The first one is denoted SaaF (Surrogate as a Filter) and comes from a P-SAEA [11]. The corresponding AP consists in generating multiple new solutions by reproduction and to filter them through the EC to retain

the q most promising candidates. The second coupling is derived from a P-SDA and is denoted *cl-mean* (Constant Liar with Mean) [14]. In *cl-mean*, to generate q new candidates, the IC is optimized q times and the surrogate model is updated between each optimization. The surrogate update is based on the new proposed candidates associated to the mean of the objective values observed in the database of already simulated points. As soon as the q new candidates are available, they are simulated in parallel.

The question of what is a promising solution is answered by defining the promisingness. In this work, we focus on two ensembles of ECs: the voting committee of ECs and the dynamic inclusive ensemble of ECs. The voting committee already presented in [15], consists in making the ECs vote for the candidates. A solution receives one vote if one EC considers it as promising and the candidates gathering the more votes are the most promising ones. The actual committee highlighted here is denoted *com-spf* and embeds three ECs, the minimization of $\hat{f}$ (favoring exploitation), the maximization of $\hat{s}$ (promoting exploration) and a third EC based on the Pareto dominance between exploitation and exploration similar to the one exhibited in [16]. The dynamic inclusive ensemble of EC denoted *dyn-df-incl* comprises two ECs, the maximization of the distance d to the database of known solutions (favoring exploration) and the minimization of $\hat{f}$. The contribution of each EC to constitute the batch of the q new candidates varies during the search. Indeed, the search is decomposed into 5 equal periods and the proportions of the contribution for (max d, min $\hat{f}$) are consecutively (100%,0%), (75%,25%), (50%,50%), (25%,75%) and (0%,100%). In other terms, exploration is favored at the beginning of the search and exploitation is reinforced at latter stages.

3 COVID-19 contact reduction problem

At the beginning of the Covid-19 crisis, when no vaccines were available, governments of the affected countries adopted different strategies to contain the spread of the virus. While some countries imposed lockdown and physical distancing, others, bet on reaching herd immunity by natural transmission. This approach has not proven to be effective during the first two years of the epidemic [17]. However, at the time, studying the possible consequences of this strategy was of importance. Recently, the new Omicron variant of the coronavirus and the deployment of vaccines revive the debate about herd immunity [18].

The problem consists in optimizing the contact reduction strategy to minimize the number of Covid-19-related deaths in Spain while reaching herd immunity. The Spanish population is divided into 16 age-categories and the decision variables represent the contact mitigation factors to apply to each category. For a decision vector $\boldsymbol{x} \in [0,1]^{16}$, $f_1(\boldsymbol{x})$ represents the simulated number of deaths after the considered period and $f_2(\boldsymbol{x}) \in \{0,1\}$ is a simulated boolean variable indicating whether herd immunity has been reached. The optimization problem

consists in finding $\boldsymbol{x}^*$ such that:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in[0,1]^{16} \text{ s.t. } f_2(\boldsymbol{x})=1}{\arg\min} f_1(\boldsymbol{x}) \tag{3}$$

According to [19], handling constrained problems with EAs can be realized by different means. For our problem, it is not known how to generate feasible candidates so designing repairing operators or specific reproduction operators is impossible. Rejecting infeasible individuals would prevent to keep knowledge about the infeasible region location, besides, this technique works only if the search space is convex, that is probably not the case. Adding the amount of infeasibility as an additional objective would increase the complexity of the problem because the new objective would be boolean. Finally, we opt for the penalization of the infeasible candidates to handle the constraint of the Covid-19 contact reduction problem. The penalty value is set to the approximate Spanish population size (46,000,000) as it is the only *a priori* known upper bound for f_1. A higher value would more likely prevent the search to visit the boundary region between the feasible and infeasible search spaces.

Therefore, the problem is re-formulated as an unconstrained optimization problem by applying a penalty to the objective f_1 when herd immunity is not reached. The re-formulated problem consists thus in finding $\boldsymbol{x}^*$ such that:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in[0,1]^{16}}{\arg\min}\, \tilde{f}(\boldsymbol{x}) \tag{4}$$

where:

$$\tilde{f}(\boldsymbol{x}) = \begin{cases} f_1(\boldsymbol{x}) \text{ if } f_2(\boldsymbol{x}) = 1 \\ f_1(\boldsymbol{x}) + 46,000,000 \text{ if } f_2(\boldsymbol{x}) = 0 \end{cases} \tag{5}$$

The impact of the contact reduction strategy is simulated thanks to the AuTuMN simulator available at `https://github.com/monash-emu/AuTuMN/` [20]. This simulator is developed by the Department of Public Health and Preventive Medicine at Monash University in Melbourne, Australia, to study epidemic transmission. Both quantities f_1 and f_2 are obtained *via* resolution of differential equations governing the flow of individuals in a compartmental model where the population is divided according to the disease state (Susceptible, Exposed, Infectious, Recovered) [21]. The graph of f_1 is expected to be multi-modal with flat regions according to the prior knowledge issued by the developers of AuTuMN. The simulation takes place in three phases. First, the past dynamic of the epidemic is analysed by calibrating uncertain parameters according to past information. Second, the contact reduction strategy is applied during a period of 12 months. After the 12-month period, mobility restrictions are lifted and herd immunity is recognized if incidence still decreases after two weeks while assuming persistent immunity for recovered individuals [22]. The degrees of contact between individuals are integrated into the model through the contact matrix C provided by [23] where the populations are divided into 16 age-categories. $C_{i,j}$ is the average number of contacts per day that an individual of age-group j makes

with individuals of age-group i. The decision variables representing the mitigation factors are applied to matrix C such that $C_{i,j}$ is replaced by $x_i.x_j.C_{i,j}$. A decision variable $x_i = 0$ impedes any contact to individuals from age-category i while setting $x_i = 1$ lets the contact rates unchanged compared to the pre-Covid-19 era.

4 Hybrid Acquisition Processes

The two categories of P-SBOAs, namely P-SAEAs and P-SDAs, are attractive for different budgets or landscapes as shown in [1]. In this section, we attempt to retain the best of both categories by investigating the design of hybrid APs. The generation of new candidates is envisioned *via* both IC optimization and reproduction operators.

Two APs are combined into two novel optimization algorithms. The first AP is a *cl-mean* with the voting committee EC *com-spf* and the GP_RBF surrogate model. The second AP is inspired by P-SAEA, where a BNN_MCD surrogate is only used as a filter to discard unpromising candidates (SaaF). Both APs are the most adequate for each framework on the Covid-19 problem as identified by a preliminary grid-search considering multiple surrogate models (Kriging, Bayesian Linear Regressor, GP_RBF, BNN_MCD), definitions of promisingness (Expected Improvement, Lower Confidence Bound *etc.*) and surrogate-optimizer couplings (notably Kriging Believer, *cl-max* and *cl-min*) [13].

The first new hybrid method is named HCAP for "Hybrid Concurrent Acquisition Process" and is presented in Algorithm 1. The two aforementioned APs are executed concurrently at each cycle to propose new candidates that are subsequently simulated in parallel. The algorithm starts by a search space sampling *via* LHS and the evaluation of the initial candidates (line 1). The surrogates are created and the population is initialized (lines 2 to 4). At the beginning of a cycle, the first AP generates $q_1 = 9$ new promising candidates (line 6). Thence, parents are selected from the population and reproduced to create a batch $\mathcal{P}_c$ of $n_{chld} = 288$ children (lines 7 and 8). From $\mathcal{P}_c$, the $q_2 = 63$ more promising candidates are retained and the remaining $n_{disc} = 225$ candidates are discarded (line 9). A total of $q_1 + q_2 = 72$ new candidates are simulated in parallel at each cycle (lines 10 and 11). Thereafter, the surrogates are updated (lines 13 and 14) and a new population is formed by elitist replacement (line 15).

The analysis led in [1] indicates that P-SDAs are relevant for few objective function evaluations and P-SAEAs to deal with moderately expensive problems. This conclusion appeals to design another hybrid method that would execute successively an AP based on IC optimization and an AP relying on evolutionary computations. The novel method is referred to as HSAP for "Hybrid Successive Acquisition Processes" and is detailed in Algorithm 2. The first stage consists of running 6 cycles of q-EGO *cl-mean* with GP_RBF and *com-spf* for $q = 18$ thus corresponding to 108 simulations (lines 2 to 11). Afterwards, P-SAEA is run with

Algorithm 1 Framework of HCAP.

Input

simulator: real objective function
budget: computational budget for the search
GP_RBF: surrogate model for AP1
com-spf: evolution control for AP1
$q_1 = 9$: number of candidates to simulate per cycle for AP1
$n_{pop1} = 50$: population size for AP1
$n_{gen} = 100$: number of generations for AP1
BNN_MCD: surrogate model for AP2
dyn-df-incl: evolution control for AP2
$n_{pop2} = 72$: population size for AP2
$n_{chld} = 288$: number of new candidates issued per cycle for AP2
$q_2 = 63$: number of candidates to simulate per cycle for AP2
$n_{disc} = 225$: number of discarding per cycle for AP2

1: *database* ← LHS+parallel_simulations(*simulator*, n_{pop2})
2: GP_RBF ← training(*database*)
3: BNN_MCD ← training(*database*)
4: $\mathcal{P}$ ← *database* ▷ initial population
5: **while** *budget*$\neq 0$ **do**
6: $\mathcal{B}_{sim1}$ ← Constant_Liar_AP(*database*, *com-spf*, GP_RBF, q_1, n_{pop1}, n_{gen})
7: $\mathcal{P}_p$ ← selection($\mathcal{P}$, n_{chld}) ▷ population of parents
8: $\mathcal{P}_c$ ← reproduction($\mathcal{P}_p$, n_{chld}) ▷ population of children
9: $\mathcal{B}_{sim2}$ ← filtering($\mathcal{P}_c$, *dyn-df-incl*, BNN_MCD, q_2, n_{disc})
10: $\mathcal{B}_{sim} \leftarrow \mathcal{B}_{sim1} \cup \mathcal{B}_{sim2}$
11: parallel_simulation(*simulator*, $\mathcal{B}_{sim}$)
12: *database* ← *database* $\cup$ $\mathcal{B}_{sim}$
13: GP_RBF ← training(*database*, 72)
14: BNN_MCD ← training(*database*, all)
15: $\mathcal{P}$ ← elitist_replacement($\mathcal{P}$, $\mathcal{B}_{sim}$, n_{pop2})
16: *budget* ← get_remaining_budget(*budget*, elapsed_time)
17: **end while**
18: ($\boldsymbol{x}_{min}$, y_{min}) ← get_best_cost(*database*)
19: **return** $\boldsymbol{x}_{min}$, y_{min}

reproduction operators informed by BNN_MCD through the *dyn-df-incl* EC until the budget is totally consumed (lines 12 to 24). The population is initialized by taking a special care of balancing between exploration and exploitation. To foster exploitation, the 10 best candidates identified so far are included in the initial population (line 12). To boost exploration, a K-Means algorithm [24, 25] partitions the set of decision vectors from the database into 62 groups and one randomly-selected solution per cluster is added to the initial population (line 13).

To test the new HCAP and HSAP, the SMBO+EA from [2] is reproduced by considering $n_{cores} = 18$ as allowed by the computational budget further described in the next section. Moreover, the GP_RBF surrogate model replaces the Kriging model originally employed in [2] as this latter has not been relevant in the preliminary grid search. In SMBO+EA, a cycle consists in running three APs in parallel. The first AP, executed on one computing core, maximizes the Expected Improvement IC [12] to produce a new candidate. The second AP, also running on one computing core, minimizes $\hat{f}$ to output one new solution. The third AP generates $q = 16$ new candidates *via* reproduction of 16 parents extracted from the current population. The 18 new candidates are simulated in parallel on the 18 cores. After the simulation step, the database, the surrogate and the population are updated and the cycle is repeated until the computational budget is wasted.

In SMBO+EA, no EC is used in the AP based on the reproduction operators whereas a dynamic ensemble of ECs helps to discard unpromising candidates in HCAP and HSAP. Relying on an EC at this step gives more opportunity to the reproduction operators to generate good candidates. The objective pointed out in [2] for future works is to improve the performance of the method when n_{cores} increases. Indeed, in the experiments reported in [2], SMBO+EA performs similarly to P-EA (without surrogate) for $n_{cores} \geqslant 15$. In HCAP and HSAP, the use of two surrogates from different types aims at enhancing diversification in the batch of new samples and improving the overall performance of the hybrid methods. In SMBO+EA, the three APs are performed in parallel while the two APs from HCAP are performed sequentially thus giving a slight advantage to SMBO+EA regarding idleness of computing cores.

5 Experiments

The experimental protocol consists in repeating the execution of the algorithms ten independent times to compute the statistics reflecting the performance of the stochastic methods. The ten initial databases are constituted *via* LHS. The experiments are supported by a parallel machine made of 18 computing cores provided in an Intel Xeon Gold 5220 CPU. The parallel machine is part of the Grid5000, a French infrastructure dedicated to parallel and distributed computing and enabled by several universities [26]. A computational budget of 30 minutes on 18 computing cores is granted for each search.

Algorithm 2 Framework of HSAP.

Input

simulator: real objective function
budget: computational budget for the search
GP_RBF: surrogate model for AP1
com-spf: evolution control for AP1
$q_1 = 18$: number of candidates to simulate per cycle for AP1
$n_{pop1} = 50$: population size for AP1
$n_{gen} = 100$: number of generations for AP1
BNN_MCD: surrogate model for AP2
dyn-df-incl: evolution control for AP2
$n_{pop2} = 72$: population size for AP2
$n_{chld} = 288$: number of new candidates issued per cycle for AP2
$q_2 = 72$: number of candidates to simulate per cycle for AP2
$n_{disc} = 216$: number of discarding per cycle for AP2

database ← LHS+parallel_simulations(*simulator*, n_{pop2})
GP_RBF ← training(*database*)
counter=0
while *counter*< 6 AND *budget*≠ 0 **do**
  $\mathcal{B}_{sim}$ ← Constant_Liar_AP(*database*, *com-spf*, GP_RBF, q_1, n_{pop1}, n_{gen})
  parallel_simulation(*simulator*, $\mathcal{B}_{sim}$)
  database ← *database* ∪ $\mathcal{B}_{sim}$
  GP_RBF ← training(*database*)
  budget ← get_remaining_budget(*budget*, elapsed_time)
  counter=*counter*+1
end while
$\mathcal{P}$ ← get_best(*database*, 10) ▷ initial population
$\mathcal{P}$ ← $\mathcal{P}$∪ K-Means_sampling(*database*, 62)
BNN_MCD ← training(*database*)
while *budget*≠ 0 **do**
  $\mathcal{P}_p$ ← selection($\mathcal{P}$, n_{chld}) ▷ population of parents
  $\mathcal{P}_c$ ← reproduction($\mathcal{P}_p$, n_{chld}) ▷ population of children
  $\mathcal{B}_{sim}$ ← filtering($\mathcal{P}_c$, *dyn-df-incl*, BNN_MCD, q_2, n_{disc})
  parallel_simulation(*simulator*, $\mathcal{B}_{sim}$)
  database ← *database* ∪ $\mathcal{B}_{sim}$
  BNN_MCD ← training(*database*, all)
  $\mathcal{P}$ ← elitist_replacement($\mathcal{P}$, $\mathcal{B}_{sim}$, n_{pop2})
  budget ← get_remaining_budget(*budget*, elapsed_time)
end while
($\boldsymbol{x}_{min}$, y_{min}) ← get_best(*database*, 1)
return $\boldsymbol{x}_{min}$, y_{min}

The GP_RBF, implemented through GPyTorch [27], is trained on a controlled-size set in HCAP whereas the whole database is used in HSAP and SMBO+EA. The BNN_MCD is built using the Keras library [28] and is always updated thanks to all the simulations performed so far. Training BNN_MCD lasts around 7 seconds on sets of 72 or 256 samples while the GP_RBF training varies from 40 to 100 seconds on non-normalized data. Normalizing the data limits the training of GP_RBF to 1 second due to an early stopping mechanism implemented in GPyTorch. In addition to the three hybrid methods presented in the previous section, the parallel evolutionary algorithm (P-EA) without surrogate is considered. Besides, SaaF with BNN_MCD as surrogate and *dyn-df-incl* as EC, and *cl-mean* with GP_RBF as surrogate and *com-spf* as IC are also included into the comparison. The pySBO platform is used as the software framework for implementation and experimentation [29]. The calibration of the algorithms is given in Table 1. Two versions of *cl-mean* are considered: the one where the surrogate is trained on the complete training set (CTS) made of all the solutions already simulated during the search, and the other one on the restricted training set (RTS) of the last 72 simulations. In *cl-mean*, $q = 18$ simulations are performed per cycle and the optimizer is an EA where both the selection and replacement are based on the criterion defined by *com-spf*. For this specific EA, the population size and the number of generations are set by grid-search to 50 and 100 respectively and the remaining parameters are set as in Table 1 for P-EA.

Figure 1 shows the distribution of the 10 best objective values obtained at the end of the search for each strategies. The corresponding ranking according to the average final objective value is displayed in Table 2. It can be observed that the new hybrid method HSAP significantly outperforms all its competitors. The average, median and variance of the results are all improved when employing HSAP as shown in Figure 1. The concurrent combination of APs proposed by HCAP is also a reliable strategy as, it outperforms all the non-hybrid methods and SMBO+EA as displayed in Figure 1 and Table 2. It can be noticed that SMBO+EA behaves as expected as it produces results similar to the P-EA without surrogate.

The convergence profiles are displayed in Figure 2. Expectedly, HSAP and the P-SDAs exhibit a similar very steep curve for less than 108 simulations. After the AP switch in HSAP, the improvement is slowed down but a continuous progress is noted until around 600 simulations where the convergence is almost reached. Figure 3 displays a zoom that highlights the benefit from using HSAP over *cl-mean* with GP_RBF trained on a reduced training set (RTS) from 300 simulations. Firstly, HSAP allows one to perform more simulations than *cl-mean* as indicates the length of the curves in Figure 2. Secondly, the use of the surrogate to inform the reproduction operators enables a continuous improvement as soon as the IC-based AP has reached steady state. An attractive enhancement of HSAP would be to automatically detect the flatness in the convergence curve and trigger the AP switch. Such a mechanism is not trivial to design, par-

Table 1. Calibration of the algorithms.

Symbol	Name	Value	Calibration method
	Calibration of BNN_MCD		
n_{sub}	number of sub-networks	5	grid search
n_{hl}	number of fully-connected hidden layers	1	grid search
m_u	number of units per layer	1024	grid search
λ_{decay}	weight decay coefficient	10^{-1}	grid search
l	Normal standard deviation for weights initialization	10^{-2}	grid search
p_{drop}	dropout probability	0.1	grid search
$h()$	activation function	Relu	[30]
ξ	Adam initial learning rate	0.001	[30]
	Calibration of P-EA		
n_{pop}	population size	72	grid search
p_c	cross-over probability	0.9	grid search
η_c	cross-over distribution index	10	set from [5]
p_m	mutation probability	$\frac{1}{d}$	set from [31]
η_m	mutation distribution index	50	set from [5]
n_t	tournament size	2	set from [32]
	Calibration of SaaF		
n_{chld}	children per cycle	288	grid search
q	simulations per cycle	$72 = 0.25 * n_{chld}$	[33, 34]
n_{disc}	discardings per cycle	216	$n_{chld} - q$
(δ_{ES}, n_{ES})	BNN_MCD early stopping	$(10^{-8}, 32)$	grid search
	2-fold cross-validation	yes	grid search

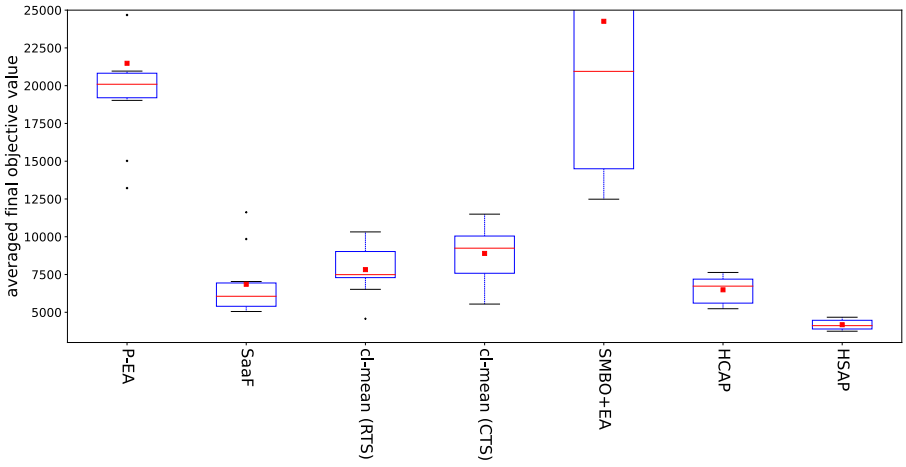

Fig. 1. Distribution of the best objective values from the 10 runs of the experiment. Averaged values are depicted by red squares, median values by red dashes and variance information is given by the length of the boxes.

Table 2. Ranking of the best strategies according to the final objective value averaged over 10 runs. Ordering according to ascending average from top to bottom.

Strategy	Average
HSAP	4,178
HCAP	6,487
SaaF	6,854
cl-mean (RTS)	7,824
cl-mean (CTS)	8,897
P-EA	21,483
SMBO+EA	24,262

ticularly because user-defined parameters must be avoided. However, exploiting the gradient of the curve is a potential lead that we plan to investigate in the future. HCAP outperforms SMBO+EA and SaaF in Figure 2 while SaaF overtakes SMBO+EA from 260 simulations. The bad performances of P-EA stressed by Figure 2 demonstrate again the profit brought by surrogate models for both moderately and very expensive problems.

The length of the curves in Figure 2 yields indications about the computational cost of the methods. Among the hybrid methods, SMBO+EA is the more computationally costly as the surrogate is trained on the entire database and IC optimizations are run at each cycle. By reducing the training set size as in HCAP, more simulations are enabled and by reducing the computational effort dedicated to IC optimization as in HSAP, the number of simulations gets closer to the one of SaaF. A possible way to relieve the computational cost of HCAP would be to execute both APs in parallel.

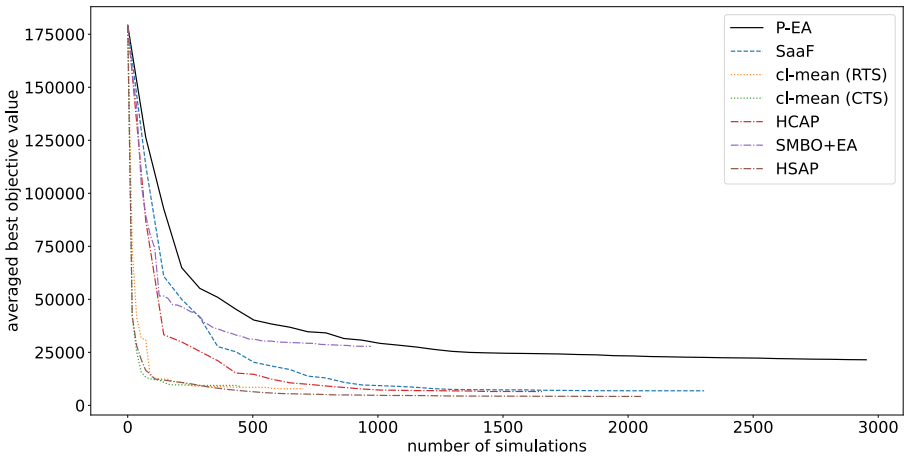

Fig. 2. Convergence profile in terms of best objective values averaged over the 10 repetitions of the experiment.

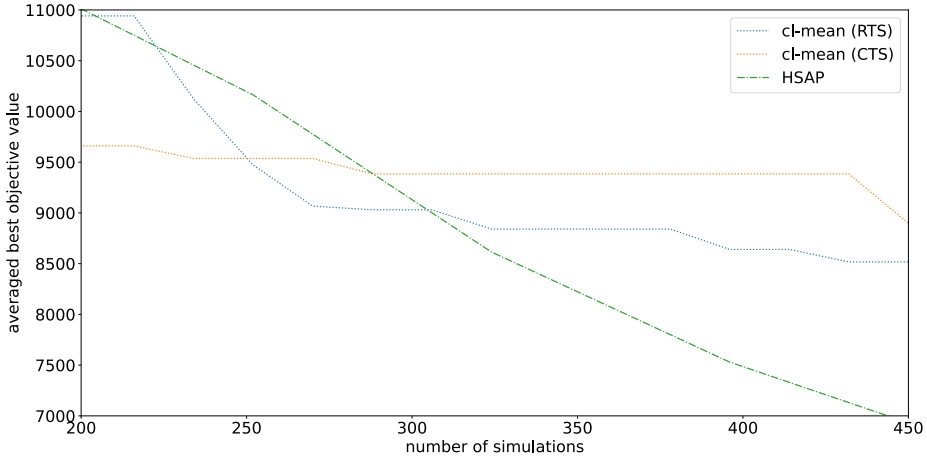

Fig. 3. Convergence profile in terms of best objective values averaged over the 10 runs of the experiment.

6 Conclusion

In this paper, the hybridization of IC optimization and informed reproduction operators is investigated to propose new candidate solutions in P-SBOAs with the aim of bringing robustness with respect to the computational budgets. The Hybrid Successive Acquisition Processes (HSAP) we propose outperforms state-of-the-art methods on a simulation-based problem of Covid-19 contact reduction with a significant number of computing cores. The new strategy consists in relying on IC optimizations during the early stages of the search and in employing informed reproduction operators at the latter stages. For tight computational budgets, only the AP inherited from P-SDAs is employed thus providing fast improvement. For larger budgets, the AP extracted from P-SAEAs is added, therefore further enhancing the search quality. The use of ensembles of ECs favors diversification in the set of newly proposed candidates consequently allowing the efficient use of multiple computing cores. Future works will consider to extend the numerical comparisons by further increasing the number of computing cores and by tackling a larger amount of benchmark problems. Moreover, the HSAP will be improved by designing a mechanism to automatically switch from one AP to another.

Acknowledgment

We thank Romain Ragonnet and the Department of Public Health and Preventive Medicine at Monash University in Melbourne, Australia, for helping us to set the AuTuMN simulator.

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. 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.
2. F. Rehback, M. Zaefferer, J. Stork, and T. Bartz-Beielstein. Comparison of parallel surrogate-assisted optimization approaches. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, page 1348–1355, New York, NY, USA, 2018. Association for Computing Machinery.
3. H. Wang, Y. Jin, and J. Doherty. Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Transactions on Cybernetics*, 47(9):2664–2677, 09 2017.
4. Y. Jin and B. Sendhoff. Reducing fitness evaluations using clustering techniques and neural network ensembles. pages 688–699, 06 2004.
5. K. Deb and P. Nain. An Evolutionary Multi-objective Adaptive Meta-modeling Procedure Using Artificial Neural Networks. In *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51, pages 297–322. 2007. doi: `http://dx.doi.org/10.1007/978-3-540-49774-5\_13`.
6. R. Regis and C. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19:497–509, 11 2007.
7. M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
8. J. Liu, W. Song, Z. Han, and Y. Zhang. Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models. *Structural and Multidisciplinary Optimization*, 55, 03 2017.
9. C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.
10. Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.
11. Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011. doi: `https://doi.org/10.1016/j.swevo.2011.05.001`.
12. 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, 12 1998.
13. G. Briffoteaux. *Parallel surrogate-based algorithms for solving expensive optimization problems*. PhD thesis, Université de Mons, Université de Lille, 2022.
14. D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, Springer series in Evolutionary Learning and Optimization, pages 131–162. springer, 2010.
15. G. Briffoteaux, R. Ragonnet, M. Mezmaz, N. Melab, and D. Tuyttens. Evolution Control Ensemble Models for Surrogate-Assisted Evolutionary Algorithms. In *High Performance Computing and Simulation 2020*, Barcelona, Spain, March 2021.

16. J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin. Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(3):459–472, 06 2019.
17. M. Claeson and S. Hanson. Covid-19 and the swedish enigma. *The Lancet*, 397(10271):259–261, 2021.
18. Medicalxpress. Weaker virus? herd immunity? omicron sparks cautious hopes. `https://medicalxpress.com/news/2022-01-weaker-virus-herd-immunity-omicron.html`, 2022.
19. Z. Michalewicz, D. Dasgupta, R. G. Le Riche, and M. Schoenauer. Evolutionary algorithms for constrained engineering problems. *Computers and Industrial Engineering*, 30(4):851–870, 1996.
20. 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.
21. J. M. Caldwell *et al.* Modelling covid-19 in the philippines: technical description of the model. Technical report, Monash University, 2020.
22. R. Ragonnet *et al.* Optimising social mixing strategies achieving covid-19 herd immunity while minimising mortality in six european countries. *medRxiv*, 2020.
23. K. Prem, A. R. Cook, and M. Jit. Projecting social contact matrices in 152 countries using contact surveys and demographic data. *PLOS Computational Biology*, 13(9):1–21, 09 2017.
24. D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 1177–1178, New York, NY, USA, 2010. Association for Computing Machinery.
25. D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the symposium on Discrete algorithms*, pages 1027–1035, 2007.
26. F. Cappello *et al.* Grid'5000: a large scale and highly reconfigurable grid experimental testbed. In *The 6th IEEE/ACM International Workshop on Grid Computing, 2005.*, 2005.
27. J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.
28. F. Chollet. Keras. `https://keras.io`, 2015.
29. G. Briffoteaux. pysbo: Python framework for surrogate-based optimization. `https://pysbo.readthedocs.io/`, 2021.
30. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.
31. E. G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.
32. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
33. Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 592–599, 2001.
34. D. Buche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2):183–194, 2005.