Surrogate-Assisted Optimization for Multi-stage Optimal Scheduling of Virtual Power Plants

Maxime Gobert, Francois Vallee, Jan Gmys, Nouredine Melab, Jean-François Toubeau, Daniel Tuyttens

Surrogate-Assisted Optimization for Multi-stage Optimal Scheduling of Virtual Power Plants

Maxime Gobert
Mathematics and Operational Research
University of Mons
Mons, Belgium
maxime.gobert@umons.ac.be

Jan Gmys
Mathematics and Operational Research
University of Mons
Mons, Belgium
jan.gmys@umons.ac.be

Jean-François Toubeau
Electrical Engineering
University of Mons
Mons, Belgium
jean-francois.toubeau@umons.ac.be

François Vallée
Electrical Engineering
University of Mons
Mons, Belgium
francois.vallee@umons.ac.be

Nouredine Melab
INRIA Lille - Nord Europe
CNRS/CRIStAL, Université de Lille
Lille, France
nouredine.melab@univ-lille.fr

Daniel Tuyttens
Mathematics and Operational Research
University of Mons
Mons, Belgium
daniel.tuyttens@umons.ac.be

***Abstract*—This paper presents a comparison between two surrogate-assisted optimization methods dealing with two-stage stochastic programming. The Efficient Global Optimization (EGO) framework is challenging a method coupling Genetic Algorithm (GA) and offline-learnt kriging model for the lower stage optimization. The objective is to prove the good behavior of bayesian optimization (and in particular EGO) applied to a real-world two-stage problem with strong dependencies between the stages. The problem consists in determining the optimal strategy of an electricity market player participating in reserve (first stage) as well as day-ahead energy and real-time markets (second stage). The decisions optimized at the first stage induce constraints on the second stage so that both stages can not be dissociated. One additional difficulty is the stochastic aspect due to uncertainties of several parameters (e.g. renewable energy-based generation) that requires more computational power to be handled. Surrogate models are introduced to deal with that additional computational burden. Experiments show that the EGO-based approach gives better results than GA with offline kriging model using smaller budget.**

***Index Terms*—Efficient Global Optimization, Two-stage optimization, power market, profit maximization, surrogate models.**

INTRODUCTION

As industrial applications require more and more computational resources to achieve good levels of accuracy, we now face problems impossible to solve in a reasonable time. To handle this time constraint, many *black-box-based* optimization methods have been developed in the last decades [1]–[4]. *Black-box* optimization is a global optimization process relying on surrogate models to substitute the so-called *black-box* function which is a costly simulation that we do not need to know precisely. One of the state-of-the-art algorithms is *Efficient Global Optimization* (EGO), presented by Jones *et al.* in [5], on which is based this study. EGO is coupled with Gaussian Processes (GP), or kriging, as a surrogate model. Kriging seems to be a natural choice because of its particularity to provide an estimation of the prediction error, which is important to compute the Expected Improvement (EI). EGO with EI as infill criterion has proven to be a good choice for highly time-constrained optimization. It has been used in industrial applications such as the optimization of the shape of horn-loaded loudspeakers in [6]. It also has been successfully applied to other research field like combinatorial problems in [7]. Literature points out the wide range of applications of bayesian optimization methods like EGO. Nevertheless, the application to multi-stage optimization is a very recent contribution. Multi-stage optimization has been rarely treated with surrogate models. In Sabater *et al.* [8], a two-stage optimization is used for uncertainty quantification on the lower level. Uncertainty quantification is replaced by a surrogate model prediction and global optimization follows the maximization of EI scheme. In Islam *et al.* [9], both stages are based on differential evolution (DE) and the inner stage is an optimization problem solved with surrogate model assistance for objective and constraint functions.

The proposed work builds on [10] where the principal objective is to provide an efficient decision tool to maximize the total profit of an electricity market player participating in medium-term operating reserves as well as day-ahead and real-time energy markets. These platforms involve different embedded time horizons going from week-ahead (reserve market) to real-time. Consequently, the problem formulation takes the form of a two-stage algorithm where the first stage (reserve market decisions) is called Medium Term Optimization (MTO) and the second stage (decisions undertaken on the day-ahead and real-time markets) is called Short Term Optimization (STO). The STO is considered as a black-box taking as inputs the first stage decisions (reserve procurement) and as output a single real value: a variable profit, in euros. The distinctive characteristic here is that we do not solve the lower problem with a traditional optimization method. We only predict its optimal value using a metamodel trained with realizations of STO of which some first-stage variables (basically, the three existing electricity reserve products) are fixed at different (optimally-chosen) values at each iteration. Furthermore, the optimization tool must take into account the prediction uncertainty due to unexpected load or renewable production to contract energy reserves and be able to contribute to the grid safe state restoration in case of unexpected dizziness on the electricity transmission system. This is the reason why it is treated using a stochastic algorithm where several scenarios of possible short-term realizations are generated [11].

This stochastic programming approach has two main challenges. The first one consists in modeling the uncertainty through a set of time-varying predictive scenarios that represent time trajectories of all uncertain variables. The second is to overcome the computational burden associated with the resulting formulation dealing with uncertainties. Both are treated in [10]. This is why the use of a metamodel is mandatory to complete the optimization in the dedicated time (2 hours maximum due to electricity market rules).

To that end, in [10], a kriging model is constructed as a preliminary task to be inserted in place of the STO. This type of substitution is often called offline learning and does not allow to improve the metamodel from experience (new simulations) and therefore, to control the evolution. In this paper, we investigate another type of surrogate-assisted optimization method to optimize the portfolio management. The approach is based on Bayesian optimization and more precisely on EGO [5] and qEGO variant from [12]. We treat STO as a black-box function and maximize the profit with a method able to improve the metamodel at each iteration. This kind of behavior is called online learning in opposition to offline learning. Experimentations are performed using a small but realistic case-study (Section III) in order to compare both approaches. The first one is constituted of an offline learnt kriging model from the `SUMO` Matlab toolbox [13] coupled to a Genetic Algorithm (GA) from `Matlab Global Optimization` Toolbox [14]. The second approach is the EGO-based one, relying on `DiceOptim` and `DiceKriging` R packages [15]. Experimental results show that the EGO-based approach finds improved solutions while requiring less calls to the simulator (namely: STO).

The paper is organized as follows. First, a short presentation of the two-stage stochastic optimization tool is provided in Section I. Theoretical aspects of the used methods are presented in Section II. Afterwards, in Section III we present the comparison of both approaches and display experimental results. Finally, some conclusions and perspectives for future works are drawn.

I. Problem description

A. Context and problem presentation

1) Context: Following the liberalization of the electricity sector, generation and transmission activities are fully decoupled. This context resulted in the development of aggregators, also referred to as Virtual Power Plants (VPPs) whose initial goal is to make some profit by helping the transmission system operator to maintain balance between generation and consumption (ensuring consequently a constant 50 Hz frequency within the grid). Practically, VPPs can be seen as single actors combining different generation, storage and load management technologies that are jointly operated within the objective to maximize their expected profit. In order to do so, VPPs can either participate in energy markets (selling/buying energy to other actors), or offer services to the transmission system operator to help him maintaining a safe and efficient system operation. Such services are typically contracted in mid-term (i.e. week or month ahead).

2) Multistage optimization: One can distinguish three levels in the optimization process: (1) the mid-term, (2) the day-ahead and (3) the real-time, each one is a market where decisions are taken. The combination of the day-ahead and real-time markets is also named short-term and is an optimization process itself. The decision tool is designed from the VPP perspective so that it maximizes its expected profit. The aggregator decision procedure involves two different (time-dependent) stages. First, the optimization is done at the mid-term level where decisions reserve procurement have to be taken. Practically, three products do exist depending on the required dynamics and are named R_1 (to be activated within 30 seconds to stabilize the frequency shift around 50 Hz), R_2 (to be fully activated within 7.5 and 15 minutes to help restoring the frequency to 50 Hz) and R_3 (longer-term reserve if frequency has not been set back to 50 Hz after use of R_1 and R_2). Let us call X_u the decision vector of first stage variables. Second, based on that mid-term commitment, the contribution to day-ahead and real-time markets must be decided, it sets the second stage variables X_l (e.g. activation of the reserves in real-time, unit commitment of generators). Since it is designed to couple two time horizons, it takes the form of a two-stage optimization problem where optimization is made at each stage.

One could write the problem as follows:

$$max\ P_t(X_u, X_l) = max\ P_f(X_u) + \mathbb{E}\left[P_v(X_u, X_l)\right] \quad (1)$$

$$s.t\ X_l = argmax\left(\mathbb{E}\left[P_v(X_u, X_l)\right]\right) \quad (2)$$

The X_u variable is a real value triplet containing the volumes (R_1, R_2, R_3), and X_l is a mixed-integer vector. P_f and P_v are the two contributions to the total profit P_t, respectively corresponding to the fixed profit (linear combination of X_u) and the variable profit coming from the Mixed Integer Linear short-term Optimization problem (second stage) and related to X_u.

B. Two-stage problem formulation

As pointed out above, the optimization of the two temporal horizons (mid-term and short-term) are intrinsically linked. Indeed, the resources allocated in mid-term decisions must remain available if they are to be activated in real-time. Fig. 1 illustrates both stages of the algorithm and their outputs P_v and P_f.

Furthermore, the STO has to be carried out on a daily basis and some moderate risk attitude can thus be envisaged

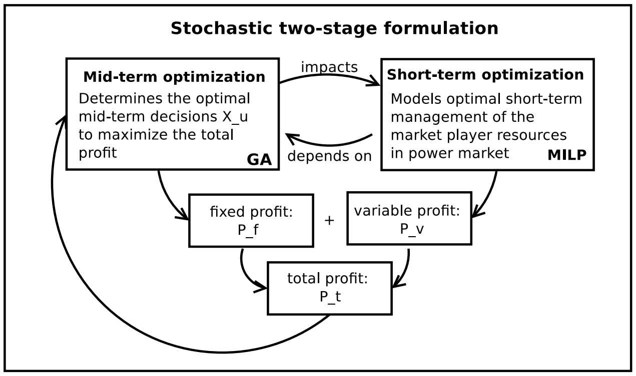

Fig. 1. Summary of the two-stage problem description

here. Two-stage scenario-based optimization is therefore favored over techniques such as robust or interval optimization techniques that are known to yield conservative (and thus suboptimal) solutions [16].

1) Mid-term Optimization: One of the major constraints is to maintain the balance on the power grid: at any time, the consumption must be equal to the production. Any failure to respect this balance can have disastrous effects such as congestion or blackout of the electricity system [17]. The transmission system operator is responsible for this constraint and has to contract energy reserves to be triggered when needed. This situation instores an intricate relation between the electricity market and the reserve market so that it is highly preferable to tackle both problems simultaneously as a two-stage optimization problem.

The first stage optimization (MTO) tries to determine the optimal participation to the services (reserves offered to the system operator). In practice, MTO decides the volume of reserves to be contracted. As already mentioned, the energy reserve is composed of three different types: primary, secondary and tertiary reserves($X_u = (R_1, R_2, R_3)$). They are organized with respect to their temporal dynamics (response speed), with the primary product being the most reactive one. The total profit, the one we want to maximize, has two contributions: the first one called fixed profit depends linearly on the triplet fixed at mid-term $P_f(X_u)$, and the second is the result of the the non-linear second-stage optimization (STO) $P_v(X_u, X_l)$.

2) Short Term optimization metamodeling: STO is an optimal scheduling problem and is itself a two-stage optimization problem (which makes the global problem a three-stage algorithm). It gathers the day-ahead and the real-time decisions. During the process, decisions concerning short-term market participation or reserve activations are taken regarding the mid-term decisions. The decisions must consider constraints such as the physical limits of the various means of production, or the necessity to be balanced in real-time (i.e. what has been procured in previous market floors has to be actually delivered). The STO constitutes the non-linear part of the objective function: it gives the variable profit P_v by optimizing the resource management in day-ahead and real-time markets. The short-term optimization is formulated as a Mixed Integer Linear Program (MILP) following the model from [18]. It is subject to the three volumes of reserves set in the mid-term decision process.

II. Surrogate-Based Optimization

Surrogate-based optimization (SBO) is used to tackle expensive-to-evaluate objective functions within an optimization process. The main idea is to momentarily sacrifice the model precision (using a surrogate model) in favour of execution speed. Using such a method allows to use global optimization methods that require a lot of evaluations - like population-based algorithms. Here, our attention is focused on a particular type of surrogate model, which is kriging.

A. Kriging

Kriging has been developed first in the field of geostatistics by the mining engineer Danie G. Krige in the early 1950s. The method has been formalized by Georges Matheron in the early 1960s [19]. The kriging method consists in finding the best linear unbiased predictor (BLUP). It relies on a set of observations $Y = (y^{(1)}, \ldots, y^{(n)})$ at locations $X = (x^{(1)}, \ldots, x^{(n)})$ and their spatial correlation: how sensible is the target according to the location variation. The target Y is assumed to be the realization of a gaussian process $\mathcal{N}(\mu(x), \sigma^2(x))$.

1) Kriging predictor: Let us consider the random variable $Y(x) = \mu(x) + Z(x)$ with $\mu(x)$ the trend, and $Z(x)$ the residual function. The trend can be written as a linear combination of L basis functions and $Z(x)$ as a centered gaussian process (GP) (as we removed the trend/mean). We end up with $Y(x) = \sum_{l=0}^{L} a_l \phi_l(x) + Z(x)$ and the BLUP at location x^* is $\hat{Y}(x^*) = \sum_{i=1}^{n} \omega_i Y(x^{(i)})$ where $Y(x^{(i)})$ are the n observations. One (strong) hypothesis is that the centered gaussian process has to be stationary: $\mathbb{E}[Z(x)] - \mathbb{E}[Z(x+h)] = 0, \forall x, h \in \mathbb{R}^n$ and $\mathbb{V}ar[Z(x) - Z(x+h)] = f(h) < \infty, \forall x, h \in \mathbb{R}^n$. This means that the mean is constant over the search space, and we have a spatial correlation between data. Intuitively, for two locations x_1 and x_2, $Z(x_1)$ must be close to $Z(x_2)$ and only depends on $h = u - v$. This is the problematic part when dealing with optimization processes. We cannot be sure that for two executions of the optimizer with close parameters, we end up with close optimized solutions.

In the application targeted by this paper, the MTO only requires the variable profit resulting from STO. One of the challenges of this work is to substitute an optimization process (the STO) by a surrogate model. The problem lies in the nature of the optimization: for the given objective function of the STO $P_v : E \subset \mathbb{R}^n \rightarrow F \subset \mathbb{R}$ we are interested in finding an optimal point $X_l \in E$ which gives the optimum value of P_v for a fixed X_u. As the algorithm in question is a MILP one can not assume the continuity of $\psi : X_u \rightarrow X_l$. But the fact is that we are searching for an optimal triplet $X_u^* = (R_1^*, R_2^*, R_3^*)$ which maximizes the total profit P_t. So that when considering the total profit computation P_t (instead of P_v) as a black-box, it is reasonable to consider that for a neighborhood of X_u, the profit remains in a neighborhood of $P_t(X_u, X_l)$. Indeed, for close mid-term decisions, the optimal management of the remaining resources should earn similar variable profits and thus, similar total profits. Furthermore, the first approach consisting in building a kriging model as preliminary task shows good results [10].

The BLUP must be unbiased and minimize the variance: min $\mathbb{V}ar[\hat{Y}(x^*) - Y(x^*)]$ such that $\mathbb{E}[\hat{Y}(x^*) - Y(x^*)] = 0$. The solution of the problem gives the universal kriging predictor which writes (from [15]):

$$m(x^*) = f(x^*)\omega^T + c(x^*)^T C^{-1}(Y - F\omega) \tag{3}$$

$$\begin{aligned}\sigma(x^*) = & C(x^*, x^*) - c(x^*)^T C^{-1} c(x^*) \\ & + (f(x^*)^T - c(x^*)^T C^{-1} F)^T (F^T C^{-1} F)^{-1} \\ & (f(x^*)^T - c(x^*)^T C^{-1} F)\end{aligned} \tag{4}$$

with $\omega = (F^T C^{-1} F)^{-1} F^T C^{-1} Y$, and $F = (f(x^{(1)}), \ldots, f(x^{(n)}))$, $f(x)$ being the vector of trend functions values at location x, C the covariance matrix and $c(x^*)$ correlation vector. We will detail C and $c(x^*)$ in the next section. Note that $f(x^*)\omega^T$ corresponds to the linear combination of basis functions: it is the trend value. The second part is related to the spatial correlation between data. More details on this formula can be found in [15].

2) Spatial correlation: Kriging is based on the spatial correlation between data which is represented by the covariance kernel. The covariance kernels used for this article are built upon the following expression from Roustant *et al.* [15]:

$$c(h) = C(u, v) = \sigma^2 \prod_{j=1}^{d} \gamma(h_j, \theta_j) \tag{5}$$

where d is the dimension, $h = u - v$ and $\sigma = C_{i,i} = C\left(x^{(i)}, x^{(i)}\right) = \mathbb{V}ar\left(x^{(i)}\right)$.

Hence, from the previous statement, we have for equations (3): $C = \left(C\left(x^{(i)}, x^{(j)}\right)\right)_{i,j}$, $i, j = 1, \ldots, n$ and $c(x^*) = \left(C\left(x^*, x^{(i)}\right)\right)_i$, $i = 1, \ldots, n$. And θ is a vector of hyperparameters that must be fitted. This is often done by numerical *Maximum Likelihood Estimation* (MLE), which is an optimization process.

We usually choose a covariance kernel among well known forms. Indeed, it must satisfy some constraints to get the resulting covariance matrix positive definite. One can find those conditions in N. Cressie's book [20]. Verifying such an assumption is not an easy task and this is why we usually choose a kernel among a few forms. Among often used kernel functions, we can find: exponential, gaussian or matérn kernels. Their analytical form can be found in [15].

This model is often presented as the DACE model - *Design and Analysis of Computer Experiments*, popularised by Sacks *et al.* in [1].

B. Efficient Global Optimization (EGO)

1) EGO algorithm: EGO is an *online* learning algorithm, meaning that the metamodel is improved with time. But even more, it actively searches for interesting areas to explore and selects the most promising point. This algorithm is based on a metric, called *acquisition function*, which selects the point that most likely improves the predictor. We can mention the Probability of Improvement (PI) and the Expected Improvement (EI) [21]. They both rely on the uncertainty of the predictor, which is available in the kriging model.

The EGO algorithm works through several stages repeated sequentially.

- First, let us consider a sample and fit the model (kriging model in our study) to the data. We need a sample that represents the search space fairly well. Jones [5] advises to start with $\approx 10d$ points from *Latin Hypercube Sampling* (LHS), where d is the problem dimension. Another recommandation comes from Sóbester *et al.* [22] who suggest to initiate the process with 35% of the budget.
- The second step is to maximize the chosen acquisition function: EI, defined by Equation (6).

$$\begin{aligned}EI(x) = & (y_{opt} - y_{pred}(x))\Phi\left(\frac{y_{opt} - y_{pred}(x)}{\sigma(x)}\right) \\ & + \sigma(x)\phi\left(\frac{y_{opt} - y_{pred}(x)}{\sigma(x)}\right)\end{aligned} \tag{6}$$

where: Φ is the probability density function (pdf) of $\mathcal{N}(0, 1)$ and ϕ is the corresponding cumulative density function (cdf). EI is high when the prediction is close to the current known optimum or when the uncertainty is large. Conversely, if the prediction is reliable or not attractive, EI will be low.

There exists a few methods for that purpose, we can mention the Branch-and-Bound algorithm from Jones *et al.* [5] and the `genoud` and `BFGS` algorithms from the R package `DiceOptim` [15]. If the maximization of EI is able to provide a new point (not too close to an already known area), this latter is evaluated and the model updated.

- Once the point maximizing EI is found, it is evaluated with the real simulator (STO). This new known point is added to the sample and the model is updated.

The process is repeated until the stopping criterion is met. Stopping criterion can be for example a given budget, a maximum time reached, or a condition on the improvement. Note that the update of the model as well as the maximization of EI can become costly when the sample becomes important. Algorithm 1 summarizes the process through the pseudo-code and Fig. 2 gives an illustration of one EGO cycle.

2) qEGO adaptation: Another version of the EGO algorithm has been developed by Ginsbourger *et al.* in [12]. The structure of the algorithm remains the same but the selection method of new points is modified. Several points may be selected (q points) to be added to the sample. We present here only the practical method to get a multi-point expected improvement. The interested reader can find the theoretical description of this method in Ginsbourger *et al.* [12].

Algorithm 1 EGO pseudo-code

Create initial design $X = [x^{(1)}, \dots, x^{(n)}]$,
Evaluate $Y = f(X)$,
Fit the kriging model,
while stoping criterion = false **do**
  $x^{(n+1)} = argmax_{\mathcal{D}}(EI(x))$
  Update the kriging model adding $(x^{(n+1)}, y^{(n+1)})$
end while

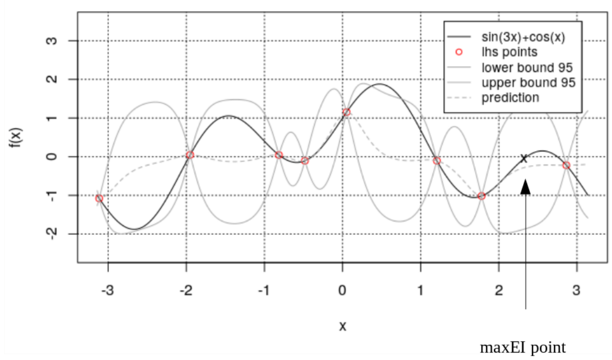

Fig. 2. One step of EGO illustrated

One convenient way to compute q-points EI is to run sequentially q EGO steps without the costly evaluation. This leads to the following heuristics to attribute a value to the selected point.

a) qEI heuristics:

- *Kriging Believer* (KB) method consists in trusting the metamodel prediction. We run EGO, select one point and attribute the predicted value as it was the answer from the simulator. We learn a new model based on the n known points to which we add the best candidate, we end up with $n+1$ points. This sequence is reapeated until we get the q points and only then the q elements are evaluated.

Believing in the kriging accuracy can misguide the optimization process because of the current quality of the model. Once again, the theoretical uncertainty on the kriging predictor can be used to counter that drawback, which leads to the *Constant Liar* heuristic.

- *Constant Liar* (CL) method is very similar to the previous one, the only difference is that we use the confidence interval given by the kriging surrogate model. Instead of using the prediction, we can decide if we use the upper or the lower prediction bound. Let us denote those two methods CL_{min} and CL_{max}.

Say we are searching for a minimum, CL_{min} has an attractive effect on the exploration process and then we favor intensification. Conversely CL_{max} has the opposite effect and then we favor exploration. One could imagine randomly use one or another version of CL heuristic to balance exploration and exploitation.

III. qEGO APPLIED TO OPTIMAL SCHEDULING OF VPP

A fair comparison requires to take into account both execution time and simulation budget. Indeed, it is not appropriate to account only for evaluation time, considering that the simulation time is strongly impacted by the aggregator portfolio and we are dealing with a small actor (around 30 seconds per evaluation) so that 2 hours is more than enough to deal with such a case. We also observed that most of the computation time is spent for the simulations and that the extra cost of learning a new kriging model at each cycle is not prohibitive to consider the simulation budget as a fair comparison metric. This is particularly true for longer simulations.

A. Settings of the testing environment

1) Hardware and software: Experimentations are run on a single core of an Intel(R) Core(TM) i7-7500U CPU clocked at 2.70GHz. We choose the R package `DiceOptim` [15] because it is user friendly and sufficiently complete. Indeed, `DiceOptim` is based on `DiceKriging` which contains several kernel functions, and various kriging forms for preliminary studies. Furthermore, in addition to classical EGO algorithm, `DiceOptim` contains functions related to qEGO. The use of qEGO, even in a sequential mode can be advantageous because of the CL heuristic presented above. It can be used as an additional parameter to balance exploitation and exploration. The simulator is written in Matlab so we interface R and Matlab using the `RMatlab` package from R [23]. The second compared method is implemented with Matlab and the Global Optimization toolbox [14] for the GA. The surrogate is constructed thanks to the SUMO toolbox [13].

2) Description of the VPP test case: The VPP is composed of 3 Conventional Power Plants (CPP), with a maximum output power of respectively 130, 80 and 55 MW as well as 2 Pump Storage Units (PSH), both characterized by an output power of 15MW (but with energy limitations of respectively 25 and 75 MWh). Then, there are also renewable generation, i.e. wind farms and domestic rooftop photovoltaic (PV) installations totalizing 120MW of power.

The mid-term uncertainty is addressed by defining statistically representative days of wind and solar generation as well as total consumption within the portfolio. Afterwards, the day-ahead scenarios are generated. The VPP is paid for both the availability of reserves (capacity) and their actual activation (energy). The prices for the reserve capacity are fixed to 16€/MW for R_1, 4€/MW for R_2, and 1€/MW for R_3. The activation prices reflect the technology-specific operation costs. The portfolio is arbitrarily created based on real data from the Belgian system in order to represent a typical actor. The generation and consumption patterns are realized regarding aggregated data for a typical month of July.

B. Technical aspects and initial settings of the methods

Both algorithms have parameters (or hyperparameters) that must be tuned for each algorithm.

1) Genetic algorithm with Gaussian Processes: The inner part (STO) is replaced by a kriging model trained with a classical Latin Hypercube Sampling to which the 8 corner points are added for a total of 48 points. The mid-term optimization is the part of the algorithm which determines the optimal triplet (R_1^*, R_2^*, R_3^*) that realizes the maximum total profit. MTO is performed by a GA from the Matlab Global Optimization toolbox [14]. Each individual of the GA is encoded by (R_1, R_2, R_3). The GA initializes the population with a uniform sampling. The selection operator is stochastic uniform. It attributes a section of a line of length proportional to the individual score and randomly picks a number to select the interval. The crossover function is a scatter function that generates a random binary operator and selects attributes from parent 1 or 2 according to the corresponding binary element, its probability is 0.8. Mutation is a small perturbation of each entry of the individual taken from a centered gaussian distribution. The gaussian standard deviation is set to $\sigma_k = \sigma_{k-1}(1 - \frac{k}{N})$, where k is the current generation, N the maximum number of generations, and $\sigma_0 = 1$. Its probability of appearance is 0.01. The number of generations is set to $100d$, which makes 300 generations maximum. Each generation is constituted of 50 individuals. The previous parameters constitute the default choices of the toolbox.

2) Efficient Global Optimization: According to preliminary experiments we decided to start with a relatively small sample containing 15 points. This roughly corresponds to 30% of the simulation budget (which is 47). 15 points are enough to represent fairly the landscape and to realize the first kriging model. The initial sampling is done by a LHS from the R package `DiceDesign` [24].

The kriging model uses a linear trend and a Matérn covariance kernel. Linear trend is chosen because of the linear part of the total profit, and the kernel choice is based on literature. Likelihood maximization is done with the `genoud` algorithm because it gives better results than the other algorithm proposed in the package, i.e. the `BFGS` algorithm. For larger problems which necessitate more sample points, `BFGS` might be favored because it is faster. Concerning the optimization process, we select the CL heuristic for the qEI points selection. This method presents interesting behavior for the exploration balance [12]. The maximization of EI is done by the *genoud* algorithm which is the defaut parameter in `DiceOptim` [15]. A brief analysis on benchmark functions revealed that even in sequential, qEGO can favor diversification in the search and avoid to remain stuck in local maxima. To confirm that, we investigate qEGO for the treated problem.

C. Experimental results

1) Investigating qEGO: The goal of this paragraph is to evaluate the impact of the parameter q on the solution quality. Fig. 3 presents the behavior of EGO and qEGO with $q = 2, 4$, or 8. The plots show the best known target (i.e. the total expected profit) relatively to the size of the sample, this means that for $q = 2$ we have 16 cycles, $q = 4$ allows 8 cycles and $q = 8$ only 4, against 32 for traditional EGO. Regarding this, it is not surprising that for equivalent sample size, the best known target is not as good for $q = 8$ which has 4 times less surrogate updates than for $q = 2$. But it is also a good point to get a wider exploration which is indicated by a higher standard deviation (for equivalent sample size).

The analysis presented in Fig. 3 has been realized with the same initial sample for every search. It shows that $q = 2$ seems to be a good compromise to explore wider without delaying the convergence. Indeed, a larger box in the boxplot reveals a larger variance and so a wider exploration. If we convert the *sample size* parameter in *number of cycles* we observe that for less cycles we also achieve good results. In particular, we reach a good expected profit with $q = 8$ with only 4 cycles. Let us remind that a cycle covers a single surrogate model update, and q simulations.

2) Comparison of the two approaches: qEGO against GA + kriging: All the executions of the SUMO approach took around 25 minutes each, and for the same simulation budget the qEGO algorithm took around 30 minutes. Even if the number of calls to the simulator is the same (47 in total, including initial training set), we observe a few minutes difference due to the update of the kriging model at each qEGO cycle and to the maximization of EI.

Fig. 4 presents a boxplot realized with 10 executions of the qEGO algorithm running with different initial samples. We first observe a fast improvement of the target (P_t) even though the standard deviation remains important. We also can see that after a few cycles ($q = 2$ points added per cycle), the variance shrinks and does not vary much after 10 cycles, which can be interpreted as a convergence to an optimum. The algorithm does not improve significantly the optimal target with more cycles. The qEGO-based approach reveals to be slightly more time consuming for equal budgets. Nevertheless, it is important to notice that the optimal solution area is found by qEGO within the 10 first cycles (cf. Fig. 4) and the algorithm would have given a satisfactory result (better than GA-based approach) even with truncated execution, this result is evidenced by Fig. 5.

Table I presents the expected optimal profit realized by the aggregator. The first column shows qEGO results, which are truly evaluated by the simulator. The second column displays GA results within two sub-columns. The GA-based method runs with the surrogate model that returns predicted solutions. Optimal predicted solutions are truly evaluated *a posteriori* and presented in the *real* column. One can notice a substantial difference between the real and predicted values.

If we compare the final target of the qEGO-based approach with the one from GA-based approach on Table I, we can see that on average the daily expected total profit obtained by the qEGO approach is 244€ greater than the one obtained by the GA-based approach. This corresponds to an average annual gain of 89000€. We must also highlight the very small standard deviation of the qEGO results that show the robustness of the algorithm. Indeed the GA-based approach gives very different results from an execution to another and often

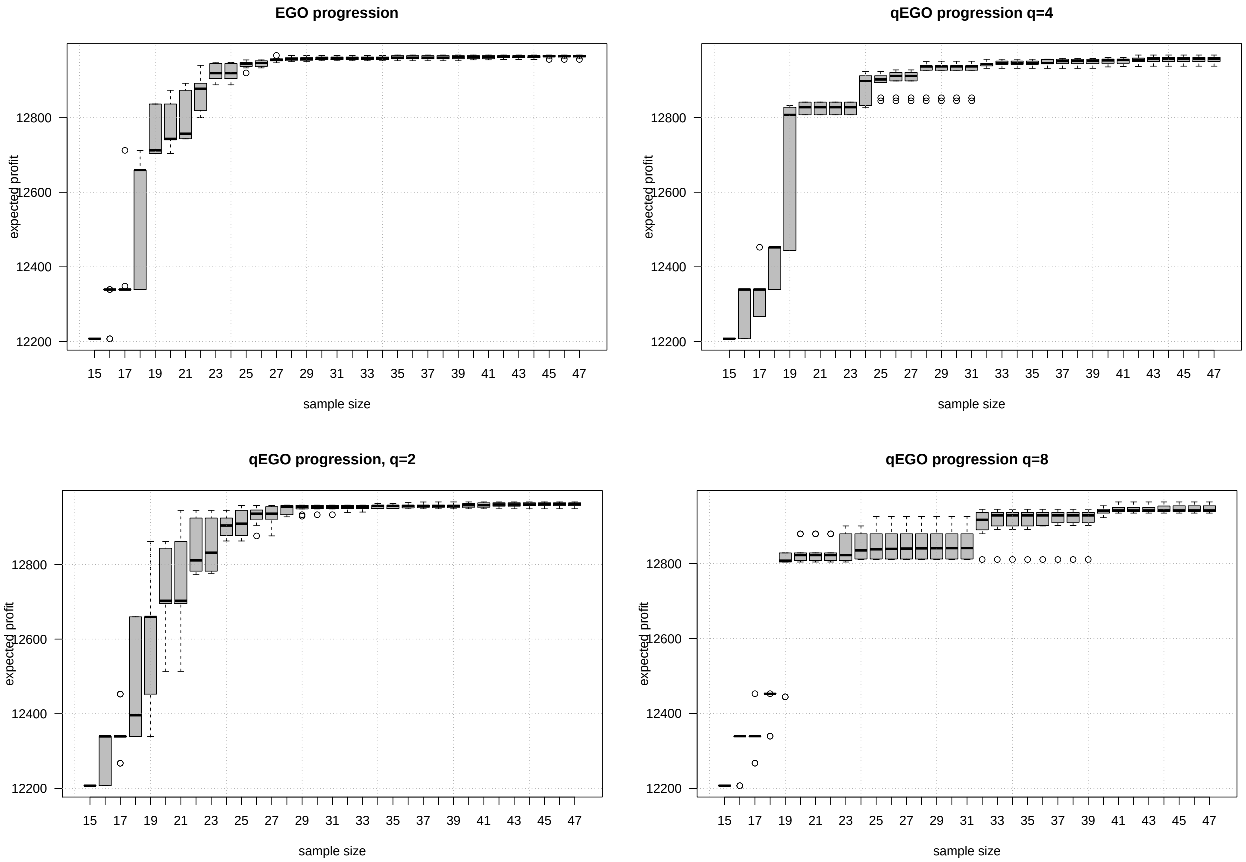

Fig. 3. Influence of the q parameter for qEGO algorithm

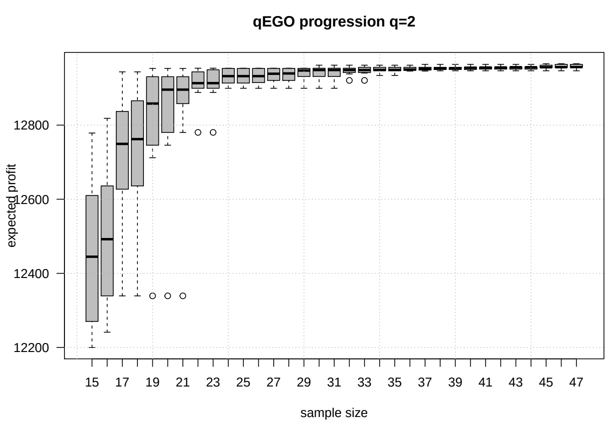

Fig. 4. Boxplot of 10 qEGO runs with different initial samples

TABLE I
Maximum profit from GA and qEGO at the end of optimization, mean over 10 realizations, in euros (€)

	qEGO	GA real	GA (prediction)
average	12 963	12 719	(12 973)
stdev	4.84	119	(85)
min	12 921	12 529	(13 092)
max	12 961	12 818	(12 923)

overestimates the optimal value that results in a suboptimal choice. When considering the extreme case realizations, we find an annual differences ranging from 50 000€ to 155 000€. Let us remind that this is a relatively small actor so that one can expect much bigger benefits with major actors.

Fig. 5 presents the results of both methods in the form of a boxplot. One can see that for any qEGO execution, we obtain a better final expected profit (qEGO18 box) than for any execution of the GA-based approach (GA_SUMO box). Furthermore, the small area of the box reveals a small standard deviation and so a stable result. Other presented boxes correspond to prematurely stopped executions of qEGO: qEGOa represents the expected total profit after a cycles. It confirms that the qEGO algorithm is more efficient in finding an optimal expected profit, and qEGO would have given good results (still better than GA+SUMO) with an early stop.

CONCLUSION AND FUTURE WORK

To overcome issues in the electrical transmission network, the system operator must contract reserves to one or more aggregators (also referred to as VPPs). Once it is settled, the VPP has to uphold the contract and reserves that are contracted in mid-term must remain available in case of need

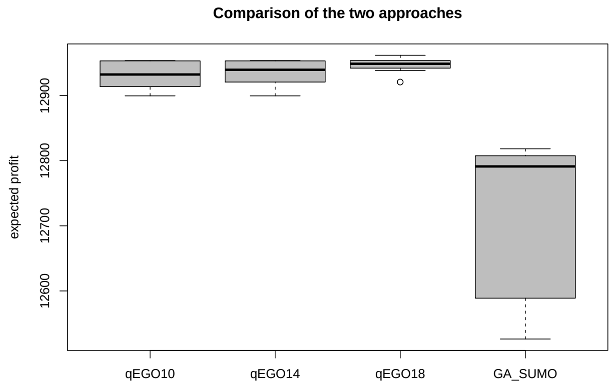

Fig. 5. Boxplot of 10 realizations for the two approaches, with truncated results of qEGO (q=2)

in real-time. Non-respect of the contract implies financial penalties for the VPP. Thus, omission of the dependencies between different time horizons can lead to non-feasible or non-tractable solutions. To deal with this issue, a two-stage stochastic optimization algorithm has been developed. The computational cost of the lower level requires the use of surrogate models to be able to treat the optimization within the time limit of 2 hours. The first approach presented here and previously developed in [10] consists in coupling a GA to an offline learnt kriging model which raplaces the STO. The second approach is based on the EGO framework. This is an online learning method that presents the advantage of having a population constituted only of simulated individuals. The metamodel starts with a restricted budget and improves with new simulations that are given by the maximization of the EI metric. Another advantage is that this algorithm is able to provide a solution even if the execution is interrupted (e.g. due to time constraints).

Considering a small, but realistic case study, the presented experimental results show that EGO and qEGO outperform the GA-based approach in terms of solution quality and execution time. Indeed, the VPP's expected profit is always larger with the qEGO-based approach. Furthermore, the optimization result is very stable between executions (very low variance). The expected extra profit for the VPP is evaluated up to 155 000€ (+3.4% compared to GA+kriging) for a year when using the qEGO-based approach for the studied test-case. Good results of the proposed optimizer coupled with surrogate models makes realistic the management of bigger problems. As mentioned, the aggregator for this study might be a real case, it remains a relatively small actor. Considering bigger actors (with bigger portfolios) and the multiplication of renewable energy sources will increase the computational burden. Therefore, combining multi-level massively parallel computing with surrogate-assisted optimization (qEGO) will be a viable and important perspective of this work.

REFERENCES

[1] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statist. Sci.*, vol. 4, no. 4, pp. 409–423, 11 1989.

[2] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, Dec 2001.

[3] G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *Journal of Mechanical Design*, vol. 129, April 2007.

[4] A. I. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1, pp. 50–79, 2009.

[5] D. Jones, M. Schonlau, and W. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 12 1998.

[6] R. Morgans, A. C. Zander, C. Hansen, and D. Murphy, "Ego shape optimization of horn-loaded loudspeakers," *Optimization and Engineering*, vol. 9, pp. 361–374, 12 2008.

[7] M. Zaefferer, J. Stork, M. Friese, A. Fischbach, B. Naujoks, and T. Bartz-Beielstein, "Efficient global optimization for combinatorial problems," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, July 2014.

[8] C. Sabater and S. Görtz, "An efficient bi-level surrogate approach for optimizing shock control bumps under uncertainty," January 2019.

[9] M. M. Islam, H. K. Singh, and T. Ray, "A surrogate assisted approach for single-objective bilevel optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 681–696, Oct 2017.

[10] J.-F. Toubeau, Z. De Grève, and F. Vallée, "Medium-term multi-market optimization for virtual power plants: a stochastic based decision environment," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1399–1410, 2018.

[11] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203–1215, March 2019.

[12] D. Ginsbourger, R. Le Riche, and L. Carraro, "A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes," Tech. Rep., Mar. 2008. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00260579

[13] D. Gorissen, K. Crombecq, I. Couckuyt, T. Dhaene, and P. Demeester, "A surrogate modeling and adaptive sampling toolbox for computer based design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, July 2010.

[14] "Matlab optimization toolbox," the MathWorks, Natick, MA, USA.

[15] O. Roustant, D. Ginsbourger, and Y. Deville, "Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization," *Journal of Statistical Software, Articles*, vol. 51, no. 1, December 2012.

[16] K. Bruninx, "Improved modeling of unit commitment decisions under uncertainty," Ph.D. dissertation, KU Leuven, 2016.

[17] J.-F. Toubeau, F. Vallée, Z. De Grève, and J. Lobry, "A new approach based on the experimental design method for the improvement of the operational efficiency in medium voltage distribution networks," *International Journal of Electrical Power and Energy Systems*, vol. 66, pp. 116 – 124, 2015.

[18] H. Pandžić, I. Kuzle, and T. Capuder, "Virtual power plant mid-term dispatch optimization," *Applied Energy*, vol. 101, pp. 134–141, 01 2013.

[19] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, pp. 1246–1266, 12 1963.

[20] N. Cressie, *Statistics for Spatial Data*. John Wiley and Sons, 1993.

[21] M. Schonlau, "Computer experiments and global optimization," Ph.D. dissertation, University of Waterloo, 1997.

[22] A. Sóbester, S. J. Leary, and A. Keane, "On the design of optimization strategies based on global response surface approximation models," *Journal of Global Optimization*, vol. 33, pp. 31–59, September 2005.

[23] B. Henrik, J. Andy, and R. Jason, "Read and write mat files and call matlab from within r."

[24] D. Dupuy, C. Helbert, and J. Franco, "DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments," *Journal of Statistical Software*, vol. 65, no. 11, pp. 1–38, 2015.