Fast computation of the multipoint Expected Improvement with applications in batch selection

Clément Chevalier, David Ginsbourger

Fast computation of the multipoint Expected Improvement with applications in batch selection

Clément Chevalier
Institute of Mathematical Statistics and Actuarial Science (IMSV)
University of Bern
Alpeneggstrasse 22, 3012 Bern, Switzerland
clement.chevalier@stat.unibe.ch

David Ginsbourger
IMSV, University of Bern
ginsbourger@stat.unibe.ch

Abstract

The Multipoint Expected Improvement criterion (or q-EI) has recently been studied in batch-sequential Bayesian Optimization. This paper deals with the new way of computing q-EI, without using Monte-Carlo simulations, through a new closed-form formula. The latter allows a very fast computation of q-EI for reasonably low values of q (typically, less than 10). New parallel kriging-based optimization strategies, tested on a 6-dimensional toy example, show promising results.

1 Introduction

In the last decades, metamodeling, or surrogate-modeling has been increasingly used for problems involving costly computer codes (or "black-box simulators"). The practitioners typically dispose of a very limited evaluation budget and aim at selecting evaluation points cautiously when attempting to solve a given problem. In global optimization, the code is often seen as a real-valued function f with d scalar inputs. In this settings, [1] proposed the now famous Efficient Global Optimization (EGO) algorithm, relying on a Kriging metamodel [2] and on the Expected Improvement (EI) criterion [3]. In EGO, f is optimized by sequentially evaluating points maximizing EI. A crucial advantage of this criterion is its fast computation (besides, the analytical gradient of EI is implemented in [4]), so that the hard optimization problem is replaced by series of much simpler ones.

Coming back to the decision-theoretic roots of EI [5], a multipoint EI for batch-sequential optimization was defined in [6] and further developed in [7, 8]. Even though an analytical formula was derived in [7] for $q = 2$, the only available method for computing q-EI when $q \geq 3$ seems to be the Monte Carlo (MC) approach of [8]. However, the latter makes the criterion itself expensive-to-evaluate, and quite tricky to optimize. A lot of effort has recently been paid to adress this problem. The pragmatic approach proposed by [8] consists in circumventing a direct q-EI maximization, and replacing it by simpler strategies where batches are obtained using on offline q-point EGO. In such strategies, the model updates are done using dummy response values such as the Kriging mean prediction (Kriging Believer) or a constant (Constant Liar). In [9] and [10], q-EI optimization strategies were proposed relying on the MC approach, where the number of samples is tuned online to discriminate between candidate designs. Finally, [11] proposed a q-EI optimization strategy involving stochastic gradient, with the crucial advantage of *not* requiring to evaluate q-EI itself.

Here we derive a formula allowing a fast deterministic evaluation of q-EI. This formula may considerably speed-up strategies relying on this criterion. Following the main result in Section 2, the usability of the proposed formula is illustrated in Section 3 through a benchmark experiment.

2 Computation of $q-$points Expected Improvement based on Tallis formula

In this section we give an explicit formula allowing a fast an accurate deterministic evaluation of q-EI. Let $\mathbf{Y} := (Y_1, \ldots, Y_q)$ be a Gaussian Vector with mean vector $\mathbf{m} \in \mathbb{R}^q$ and covariance matrix Σ. We consider a maximization problem and a threshold T; T is typically the maximum of the n available response values. Our goal is then to explicitly calculate:

$$\mathrm{EI}_q := \mathbb{E}_n[(\max_{i\in\{1,\ldots,q\}} Y_i - T)_+] \tag{1}$$

where $(.)_+ := \max(., 0)$. In an EGO algorithm, $\mathbf{Y}$ is the unknown response of f at a given batch of q points $(\mathbf{x}_1, \ldots, \mathbf{x}_q) \in \mathbb{X}^q$ where $\mathbb{X}$ is the input set of f (often, a compact subset of $\mathbb{R}^d$, $d \geq 1$).

To obtain a tractable analytical expression for Expression (1), let us first recall a useful formula given and proven in [12], and recently used in [13] in a Gaussian Process framework:

Proposition 1 (Tallis formulas)**.** *Let $\mathbf{Z} := (Z_1, \ldots, Z_q)$ be a centred, normalised Gaussian Vector (meaning that $\forall k \in \{1, \ldots, q\},\ \mathbb{E}_n(Z_k) = 0$ and $Var(Z_k) = 1$) with correlation matrix R. Let $\mathbf{b} = (b_1, \ldots, b_q) \in \mathbb{R}^q$. One can calculate the expectation of any coordinate Z_k under the linear constraint $\mathbf{Z} \leq \mathbf{b}$ with the following formula:*

$$\mathbb{E}_n(Z_k | \forall j \in \{1, \ldots, q\},\ Z_j \leq b_j) = -\frac{1}{p} \sum_{i=1}^{q} R_{ik}\ \varphi(b_i)\ \Phi_{q-1}\left(\mathbf{b}^{(i)}, R^{(i)}\right) \tag{2}$$

where:

- $p := P(\mathbf{Z} \leq \mathbf{b}) = \Phi_q(\mathbf{b}, R)$
- $\Phi_q(\mathbf{u}, \Sigma)$ $(\mathbf{u} \in \mathbb{R}^q, \Sigma \in \mathbb{R}^{q\times q}, q \geq 1)$ *is the c.d.f. of the centred multivariate Gaussian distribution with covariance matrix Σ.*
- $\varphi(.)$ *is the p.d.f. of the standard univariate Gaussian distribution*
- $\mathbf{b}^{(i)}$ *is the vector of $\mathbb{R}^{q-1}$ with general term $\frac{b_j - R_{ij}}{\sqrt{1-R_{ij}^2}}$, $j \neq i$*
- $R^{(i)}$ *is the $(q-1)\times(q-1)$ matrix of partial correlations knowing variable i. This matrix is obtained by computing $\frac{R_{uv} - R_{iu}R_{iv}}{\sqrt{1-R_{iu}^2}\sqrt{1-R_{iv}^2}}$ for $u \neq i$ and $v \neq i$.*

A crucial point for the practical use of this result is that there exist very fast procedures to compute the c.d.f. of the multivariate Gaussian distribution. For example, the work of [14], [15] have been used in many R packages (see, e.g., [16], [17]). The Formula (2) above is a crucial tool to efficiently compute Expression (1) as shown with the following Property:

Proposition 2. *The $q-$points Expected Improvement can be efficiently computed by applying Formula (2) q times, on q different "well chosen" Gaussian vectors.*

Proof. Using that $\mathbb{1}_{\{\max_{i\in\{1,\ldots,q\}} Y_i \geq T\}} = \sum_{k=1}^{q} \mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j \neq k\}}$, we get

$$\begin{aligned}
\mathrm{EI}_q &= \mathbb{E}_n\left[\left(\max_{i\in\{1,\ldots,q\}} Y_i - T\right)\sum_{k=1}^{q} \mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j\neq k\}}\right] \\
&= \sum_{k=1}^{q} \mathbb{E}_n\left((Y_k - T)\mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j\neq k\}}\right) \\
&= \sum_{k=1}^{q} \mathbb{E}_n\left(Y_k - T \middle| Y_k \geq T,\ Y_j \leq Y_k\ \forall j \neq k\right) P(Y_k \geq T, Y_j \leq Y_k\ \forall j \neq k)
\end{aligned}$$

The calculation of q-EI then amounts to calculate the q terms of this sum. Each term can be calculated using the same method. In particular, the k^{th} term may be rewritten as follows:

$$\mathbb{E}_n\left(Y_k - T \middle| -Y_k \leq -T,\ Y_j - Y_k \leq 0\ \forall j \neq k\right) P(-Y_k \leq -T, Y_j - Y_k \leq 0\ \forall j \neq k)$$

We then consider the following Gaussian Vector $\mathbf{Z}^{(k)} := (Z_1^{(k)}, \ldots, Z_q^{(k)})$:

$$Z_j^{(k)} := Y_j - Y_k \ , \ j \neq k$$
$$Z_k^{(k)} := \ - Y_k$$

The mean vector and covariance matrix of this random vector can be calculated straightforwardly from $\mathbf{m}$ and Σ. Now, the calculation of $P(Z_k^{(k)} \leq -T, Z_j^{(k)} \leq 0 \, \forall j \neq k)$ simply requires one call to the Φ_q function and the calculation of $\mathbb{E}_n(Z_k^{(k)} | Z_k^{(k)} \leq -T, Z_j^{(k)} \leq 0 \, \forall j \neq k)$ can be done by normalising the vector $\mathbf{Z}^{(k)}$ and applying Tallis formula (2). □

Remark. From Properties (1) and (2), it appears that computing q-EI requires a total of q calls to Φ_q and q^2 calls to Φ_{q-1}. The proposed approach performs thus well when q is moderate (typically: lower than 10). For higher values of q, estimating q-EI by Monte-Carlo might remain competitive.

3 Application: using q-EI to choose between different candidate batches

Let us first illustrate Proposition 2 and show that our q-EI calculation is actually consistent with an MC estimation. From a Kriging model based on 12 observations of the Branin-Hoo function [1], we generated a 4-point batch (Figure 2, left plot) and calculated its q-EI value (middle plot, dotted line). The MC estimate converges to a value close to the latter, and the relative error is less than 10^{-5}. Batches of four points generated from three strategies detailed below are drawn (right plot).

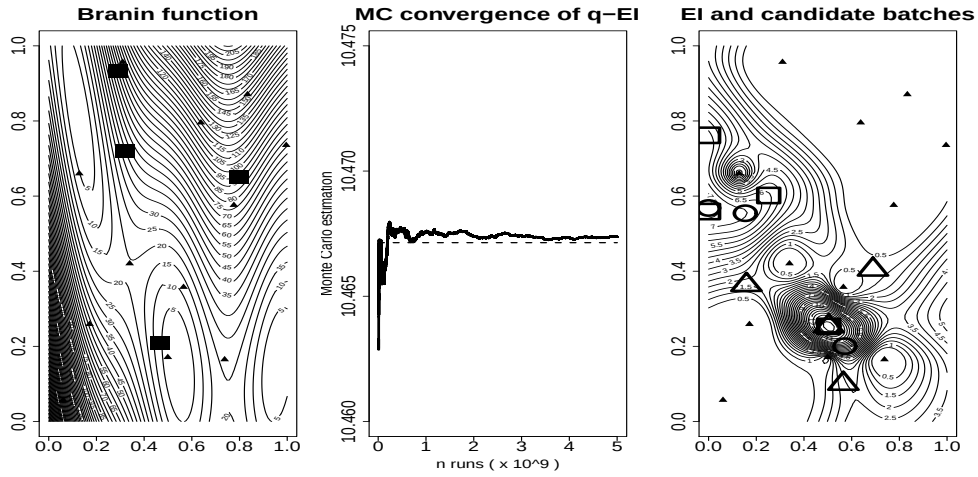

Figure 1: Convergence (middle) of an MC estimate to the q-EI value calculated with Proposition 2 in the case of a batch of four points (shown on the left plot). Right: candidate batches obtained by q-EI stepwise maximisation (squares), CL-min (circles) and CL-max (triangles) strategies.

We now compare a few kriging-based batch-sequential optimization methods on the function $x \mapsto -\log(-\text{Hartman6}(x))$ (see, e.g., [1]), defined on $[0,1]^6$. For each run of the benchmark, we start with a random initial Latin hypercube design (LHS) of $n_0 = 50$ points and estimate the covariance parameters by Maximum Likelihood (here a Matérn kernel with $\nu = 5/2$ is chosen). For all strategies, batches of $q = 4$ points are added at each iteration, and the covariance parameters are re-estimated with the q new observations. The results are analyzed in terms of evolution of the current observed minimum along the runs. Since the tests are done for several designs of experiments, we chose to represent quantiles (at levels 10%, 50%, and 90%) curves summing up the 30 curves obtained for the 30 considered LHS designs (See Figure 2). The tested strategies are:

- (1) q-EI stepwise maximization: q sequential d-dimensional optimizations are performed. We start with the maximization of the 1-point EI and add this point to the new batch. We then maximize the 2-point EI (keeping the first point obtained as first argument), add the maximizer to the batch, and iterate until q points are selected.

- (2) Constant Liar min (CL-min): We start with the maximization of the 1-point EI and add this point to the new batch. Then we assume a dummy response (a"lie") at this point, and update the Kriging metamodel with this point and the lie. We then maximize the 1-point EI obtained with updated Kriging metamodel, get a second point, and iterate the same process until a batch of q points is selected. The dummy response has the same value over the $q-1$ lies, and is here fixed to the minimum of the current observations.

- (3) Constant Liar max (CL-max): The lie in this Constant Liar strategy is fixed to the maximum of the current observations.

- (4) Constant Liar mix (CL-mix): At each iteration, two batches are generated with the CL-min and CL-max strategies. From these two "candidate" batches, we choose the batch with the best actual q-EI value, calculated based on Proposition 2.

Note that CL-min tends to explore the function near the current minimizer (as the lie is a low value) while CL-max is more exploratory. For all the tests we use the DiceKriging and DiceOptim packages [4]. The optimizations of the different criteria are done with a genetic algorithm, available in the rgenoud package [18]. Figure 2 represents the compared performances of these strategies.

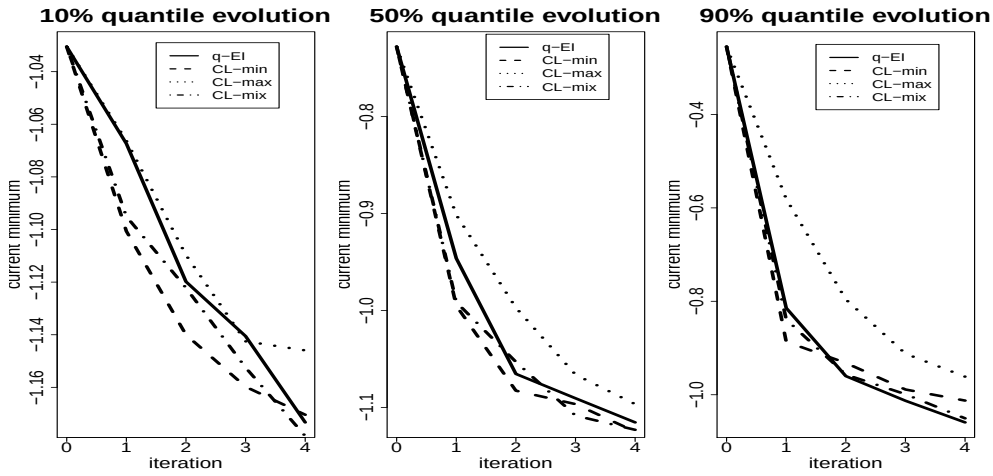

Figure 2: Compared performances of the four considered batch-sequential optimization strategies

From these plots we draw two main conclusions. First, the CL-min strategy has roughly the same performance as the q-EI stepwise maximization strategy. It points out that CL-min seems particularly well-adapted to this test case, but also potentially that the way q-EI is heuristically maximized is sub-optimal. Since running a CL is computationally much cheaper, it is tempting to recommend CL-min here. However, it is not straightforward to know in advance which of CL-min or CL-max will perform better on a given test case. A drawback considering the performances of CL-max here. Second, we can see that using q-EI in the CL-mix heuristic enables a performance close to CL-min without having to select one of the two lie values in advance. This suggest that a good heuristic might be to generate, at each iteration, candidate batches obtained with different strategies (e.g. CL with different lies) and to discriminate those batches using q-EI. Another perspective, currently under study, is the crucial need to improve the optimization method of q-EI, e.g. through a more adapted choice of the algorithm and/or a gradient calculation of q-EI as a function of $d \times q$ arguments.

Acknowledgments

This work has been conducted within the frame of the ReDice Consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (Ecole des Mines de Saint-Etienne, INRIA, and the University of Bern) partners around advanced methods for Computer Experiments. Clément Chevalier also gratefully acknowledges support from the French Nuclear Safety Institute (IRSN).

References

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

[2] T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer Verlag, 2003.

[3] J. Mockus. *Bayesian Approach to Global Optimization. Theory and Applications*. Kluwer Academic Publisher, Dordrecht, 1989.

[4] O. Roustant, D. Ginsbourger, and Y. Deville. Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodelling and optimization. conditionally accepted to the Journal of Statistical Software, 2012.

[5] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. Dixon and Eds G. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. Elsevier, 1978.

[6] M. Schonlau. *Computer Experiments and global optimization*. PhD thesis, University of Waterloo, 1997.

[7] D. Ginsbourger. *Métamodèles multiples pour l'approximation et l'optimisation de fonctions numériques multivariables*. PhD thesis, Ecole nationale supérieure des Mines de Saint-Etienne, 2009.

[8] D. Ginsbourger, R. Le Riche, and Carraro L. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, volume 2 of *Adaptation Learning and Optimization*, pages 131–162. Springer, 2010.

[9] J. Janusevskis, R. Le Riche, and D. Ginsbourger. Parallel expected improvements for global optimization: summary, bounds and speed-up. August 2011.

[10] J. Janusevskis, R. Le Riche, D. Ginsbourger, and R. Girdziusas. Expected improvements for the asynchronous parallel global optimization of expensive functions : Potentials and challenges. In *LION 6 Conference (Learning and Intelligent OptimizatioN), Paris : France*, 2012.

[11] P. I. Frazier. Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference, Miami FL*, 2012.

[12] G.M. Tallis. The moment generating function of the truncated multi-normal distribution. *J. Roy. Statist. Soc. Ser. B*, 23(1):223–229, 1961.

[13] S. Da Veiga and A. Marrel. Gaussian process modeling with inequality constraints. *Annales de la Faculté des Sciences de Toulouse*, 21 (3):529–555, 2012.

[14] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–149, 1992.

[15] A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities*. Springer-Verlag, 2009.

[16] A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, B. Bornkamp, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2012. R package version 0.9-9992.

[17] A. Azzalini. *mnormt: The multivariate normal and t distributions*, 2012. R package version 1.4-5.

[18] W. Mebane and J. Sekhon. Genetic optimization using derivatives: The rgenoud package for r. *Journal of Statistical Software*, Vol. 42, Issue 11:1–26, 2011.