Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm

R. Bardenet, Balázs Kégl

Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm

Rémi Bardenet BARDENET@LRI.FR
Balázs Kégl BALAZS.KEGL@GMAIL.COM
LAL/LRI, University of Paris-Sud, CNRS, 91898 Orsay, France

Abstract

In global optimization, when the evaluation of the target function is costly, the usual strategy is to learn a surrogate model for the target function and replace the initial optimization by the optimization of the model. Gaussian processes have been widely used since they provide an elegant way to model the fitness and to deal with the exploration-exploitation trade-off in a principled way. Several empirical criteria have been proposed to drive the model optimization, among which is the well-known Expected Improvement criterion. The major computational bottleneck of these algorithms is the exhaustive grid search used to optimize the highly multimodal merit function. In this paper, we propose a competitive "adaptive grid" approach, based on a properly derived cross-entropy optimization algorithm with mixture proposals. Experiments suggest that 1) we outperform the classical single-Gaussian cross-entropy method when the fitness function is highly multimodal, and 2) we improve on standard exhaustive search in GP-based surrogate optimization.

1. Introduction

There are numerous important global optimization problems in which the single evaluation of the target fitness function is very costly. Parameter optimization of large complex systems often requires running expensive simulations or carrying out real experiments that can take a long time. Hyperparameter optimization in Machine Learning is another example: evaluating one set of hyperparameters requires the full training that can take hours or days on today's large databases. A natural way to deal with such problems is to replace the fitness function by a cheap-to-evaluate estimator, and optimize this surrogate model to propose a small number of points where the expensive fitness function is evaluated in an iterative active learning setup. Gaussian processes (GPs) provide an elegant way to model the fitness and to deal with the exploration-exploitation trade-off in a principled way. The paradigm of global optimization based on GPs dates back to the 70's with (Mockus et al., 1978). Start with some initial training points spread over the input space, evaluate the fitness function f at those points, and repeat the following steps: 1) choose the next point to evaluate x^* by optimizing a cheap sampling criterion that measures some merit of an additional evaluation at any point of the input space, 2) evaluate f at x^*, and 3) add $(x^*, f(x^*))$ to the training set. GP regression intervenes in the sampling criterion evaluation which involves the GP posterior over the fitness function given the training set.

The design of efficient sampling criteria (the so-called *merit functions*) is a hot research topic. Recent advances in this domain include the conditional entropy of the minimizer (Villemonteix et al., 2006) dealing with noisy evaluations and a multi-armed bandit criterion (Srinivas et al., 2009) to derive regret bounds. Although our proposed technique is generic, we will concentrate throughout this paper on the classical Expected Improvement (EI) criterion of J. Mockus (see (Jones, 2001) for a recent extensive review), which measures the expected amount by which we can improve the best fitness value obtained so far by going to a new point. These criteria are usually highly multimodal, so optimization is typically done by a grid search or a Latin hypercube sampling approach that requires a large number of evaluations of the sampling criterion. This is a major draw-back of these methods especially when the evaluation involves Monte Carlo sampling from the GP (Villemonteix et al., 2006), so these methods are used mostly to optimize "expensive-

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

to-evaluate" functions when the computational time to evaluate the functions justifies the time to be spent on proposing the next evaluation point.

In this paper we propose to improve on the computational bottleneck of these methods by replacing the exhaustive evaluation of the surrogate and merit functions by a cross-entropy-based mixture method. The main idea is to replace the grid search by an adaptive evolutionary algorithm that iteratively samples in regions of higher merit value. The search distribution will be a mixture to take advantage of prior knowledge on the shape of the merit functions like EI. We have two contributions: a well-formulated mixture Cross-Entropy method and its application to surrogate optimization. Experiments suggest that 1) we outperform the classical single-Gaussian cross-entropy method when the fitness function is highly multimodal, and 2) we outperform standard exhaustive search in GP-based surrogate optimization.

The outline of the paper is as follows. In Section 2 we briefly recall GP basics, explain the details of the EI criterion, and describe the surrogate optimization problem. Then in Section 3 we formally derive a mixture cross-entropy optimization method and propose an initialization procedure using triangulation of the training data. Finally in Section 4 we benchmark our mixture algorithm as a generic global optimization method, and compare it experimentally to exhaustive search on particular EI optimization problems.

2. Surrogate optimization based on Gaussian processes

The GP (also known as *kriging*) is a popular model for surrogate optimization mainly due to its capacity to elegantly handle the uncertainty about the unknown fitness function. Several criteria have been proposed to handle the exploration-exploitation tradeoff in global optimization. The most well-known are the Probability of Improvement and the Expected Improvement (Jones, 2001). More recently (Villemonteix et al., 2006) proposed to use the conditional entropy of the global minimizer to improve on EI when the the evaluation of the fitness function is noisy. One of the most recent novelty in the field is (Srinivas et al., 2009)'s proposal of using multi-armed bandits based on a GP surrogate model. After recalling the basics of GPs in Section 2.1 (based on (Rasmussen & Williams, 2006)), we describe the most well-known criterion of Expected Improvement (Jones, 2001) in Section 2.2. We carried out all our experiments using this criterion; note, however, that the proposed technique is applicable with any GP-based merit function.

2.1. Gaussian processes

Gaussian processes (GP) provide a convenient way to put priors over functions. Let k be a positive definite kernel on the input space $\mathcal{X}$. Under a $\mathrm{GP}(0,k)$ prior, the distribution of any vector of function values $\mathbf{f} = (f(x_1), ..., f(x_n))^T$ is a multivariate Gaussian $\mathbf{f} \sim \mathcal{N}(0, K)$, where the matrix K is defined through $K_{ij} := k(x_i, x_j)$.

The most useful property of the GP prior is that it is closed under sampling: given a prior $p(f) \sim \mathrm{GP}(0,k)$ over functions and a set of samples $\mathcal{D} := \big\{(x_i, f(x_i)); 1 \leq i \leq n\big\}$, the posterior $p(f|\mathcal{D})$ is also a GP form with mean and covariance functions

$$\begin{aligned}\widetilde{m}(x) &= k(x, \mathbf{x})K^{-1}\mathbf{f},\\ \widetilde{k}(x, x') &= k(x, x') - k(x, \mathbf{x})K^{-1}k(\mathbf{x}, x').\end{aligned}$$

It is then straightforward to make predictions about the function value at a test point x^* since, according to the posterior, $f(x^*)$ has distribution $\mathcal{N}(\widetilde{m}(x^*), \widetilde{k}(x^*, x^*))$. Observe that the posterior variance at training points is zero, as the observations are noiseless. Additive Gaussian noise can also be handled easily (see (Rasmussen & Williams, 2006)).

2.2. The Expected Improvement criterion

Assume we want to minimize an unknown fitness function f, already evaluated at n points $\mathcal{D} := \{(x_i, f(x_i)); 1 \leq i \leq n\}$. The goal of EI is to find the next point x_{n+1} where the expected improvement over the currently best minimum $m_n := \min_i f(x_i)$ is the highest. We start by fitting a GP on $\mathcal{D}$ to obtain, at every test point x^*, a guess $\widetilde{m}(x^*)$ and a standard error $\widetilde{\sigma}(x^*) = \widetilde{k}(x^*, x^*)^{1/2}$. The EI merit function is then defined by

$$\mathrm{EI}(x) := \mathbb{E}\big(\max(m_n - f(x), 0)|\mathcal{F}_n\big),$$

where $\mathcal{F}_n$ is the σ-algebra generated by the previous fitness evaluations summarized in $\mathbf{f}$. An easy computation yields

$$\mathrm{EI}(x) = \widetilde{\sigma}(x)\big(u\Phi(u) + \phi(u)\big), \tag{1}$$

where $u = (m_n - \widetilde{m}(x))/\widetilde{\sigma}(x)$, and Φ and ϕ denote the cdf and pdf of the $\mathcal{N}(0,1)$ distribution, respectively. This alternative definition is easier to understand: EI represents a compromise between regions where the mean function is close to or better than the best value obtained so far and regions where the uncertainty is high. Notice that the EI merit function is always nonnegative and zero at every training point. It is generally smooth since it inherits the smoothness of the chosen kernel (which is in practice often at least once differentiable). The EI merit function is also likely to be highly multimodal, especially as the number of training points increases. The goal of this paper is to design an optimization algorithm which exploits this prior knowledge on the shape of the EI merit function to optimize it efficiently.

3. A mixture cross-entropy algorithm

Optimizing a GP-based merit function such as EI (1) is itself a difficult optimization problem. Due to its multimodality, the most common technique is grid search either on a full grid or, especially in higher dimensions, using a Latin hypercube sampling. In any case, the criterion itself has to be evaluated a lot of times in each of the outer iterations of the global optimization loop. This can be slow even if the evaluation is analytical, let alone the case when the evaluation itself requires a Monte Carlo simulation from the GP (Villemonteix et al., 2006). For this reason, GP-based global optimization is often "marketed" as a technique for optimizing expensive functions, where the high computational complexity of evaluating the original fitness function f justifies the work invested in predicting the next evaluation point. In this section we describe an approach that can improve the computational complexity of the surrogate optimization, bringing GP-based global optimization closer to the family of generic global optimizers.

Our approach uses importance sampling to adapt the search grid to the optimization problem. This is done by means of the cross-entropy method (CEM; see (Rubinstein & Kroese, 2004) for a detailed review) that we describe in Section 3.1. We make use of the fact that the multimodality of the merit functions suggests to model them with mixture distributions. Our main contribution is found in Section 3.2: we show how to use mixture distributions in CEM. Section 3.3 describes a specific initialization routine adapted to GP-based merit functions.

3.1. The cross-entropy method

The cross-entropy method is a technique originally designed for integration on rare events. It proved to apply quite naturally to optimization. CEM provides a solid mathematical justification to evolutionary sampling methods, rigorously introducing the selection step in the estimation procedure. Consider computing

$$I := \mathbb{P}_u(A) = \mathbb{E}_u 1_A = \int 1_A(x) g(x;u) dx \qquad (2)$$

where the expectation is taken with respect to a pdf $g(x;u)$ belonging to some parametric family $\mathcal{G}$ indexed by u and A is $\mathbb{P}_u$-rare. If one knows how to sample from $g(x;u)$, a crude Monte Carlo estimate of (2) is computable. But as A is rare for $\mathbb{P}_u$, few of the sampled points will happen to fall in A, so it is preferable to use importance sampling to reduce the variance of the MC estimator by sampling more points in the region of interest A. Importance sampling in this case consists in writing

$$I = \int 1_A(x) \frac{g(x;u)}{q(x)} q(x) dx \approx \sum_{i=1}^{N} 1_A(x_i) \frac{g(x_i;u)}{q(x_i)} \qquad (3)$$

for some distribution q we chose for easy sampling, whose support contains the support of $g(.;u)$ and $x_1, ..., x_N \sim q$ i.i.d. (Robert & Casella, 2004).

There is a theoretical answer to the question of the optimal choice of q, as if one takes $q = \tilde{q} \propto 1_A(.)g(.;u)$, the variance of the MC estimator will be 0. Of course, this is of no practical use, since to normalize q the integral of interest I is needed, but one can still try to approximate $\tilde{q}$ in some sense. In particular, minimizing over $g(.;v) \in \mathcal{G}$ the Kullback-Leibler divergence between $\tilde{q}$ and $g(.;v)$ is equivalent to solve

$$\max_v \int 1_A(.) g(.;u) \log g(.;v), \qquad (4)$$

or, taking the empirical counterpart of (4) with eventually a new importance sampling step:

$$\max_v \sum_{i=1}^{N} 1_A(x_i) \frac{g(x_i;u)}{g(x_i;w)} \log g(x_i;v) \qquad (5)$$

where the x_i's are drawn i.i.d. according to $g(.;w)$.

Let us now turn this remark into an evolutionary optimization algorithm adapted to our original problem. Let us denote by S the criterion to maximize over $\mathcal{X}$. The key idea is to think of estimating probabilities of level sets, i.e., integrals of the form $\mathbb{P}_u(S(X) \geq \gamma)$. Using the CEM principle to approximate the optimal importance distribution $1_{(S(.)\geq\gamma)} g(.;u)$, the importance sampling paradigm will help us to sample points in $(S(X) \geq \gamma)$. Iteratively repeating the procedure while cleverly adapting γ to keep enough samples in the region of interest should lead us to sample from close to the optima of S.

Since we do not care about the actual estimate of the integral, we can get rid of the importance weights in (5) and iteratively optimize our choice of the importance distribution $g(.;v)$ to estimate $\mathbb{P}_{v_{t-1}}(S(X) \geq \gamma_t)$. The core algorithm finally proposed by the authors of (Rubinstein & Kroese, 2004) is given in Figure 1.

Note that taking $\mathcal{G}$ to be the family of Gaussians in CEM leads to the Estimation of Multivariate Normal

```
CEM for Optimization$(S, N, \rho, d)$
  1  Initialize $v_0$, set $t \leftarrow 1$.
  2  Sample $x_1, ..., x_N \sim g(.; v^{t-1})$ i.i.d.
  3  Order $S(x_1), ..., S(x_N)$ decreasingly.
  4  Take $\gamma_t$ to be the $(1-\rho)$-quantile
     of the ordered performances.
  5  Solve
         $v^t := \arg\max_v \sum_{i=1}^{N} 1_{(S(x_i)\geq\gamma_t)} \log g(x_i; v)$     (6)
  6  If $t \geq d$ and $\gamma_t = \gamma_{t-1} = ... = \gamma_{t-d}$,
     then stop. Else set $t := t+1$ and go
     back to step 2.
```

Figure 1. The CEM algorithm: the goal is to iteratively sample in regions of higher criterion value.

Algorithm (EMNA; see (Larrañaga & Lozano, 2001) for a review on Estimation of Distribution Algorithms and their applications), a popular evolutionary algorithm used in neural network training.

3.2. Introducing mixtures into the CEM

(Rubinstein & Kroese, 2004) claim that (6) is analytically solvable when $\mathcal{G}$ is an exponential family. It turns out that there is a certain class of more generic distributions which would intuitively allow for a better fit of disconnected areas $(S \geq \gamma)$, performing better exploration of multimodal landscapes by clustering the data: the mixture distributions. Their simplest form is a weighted sum of distributions belonging to parametric family $\Phi = (\varphi(., v))_v$, i.e. $g(.; \mathbf{v}) := \sum_{d=1}^{D} \alpha_d \varphi(.; v_d)$ where $\sum_d \alpha_d = 1$. In the following subsection, we demonstrate with an EM-flavored technique that they also lead to analytical update formulae, whenever Φ is an exponential family.

At time t, denoting $g(.; v^{t-1})$ by π, the problem in its integral form is $\max_v \ell(\alpha, v)$, where

$$\ell(\alpha, v) := \int 1_{(S(x)\geq\gamma)} \times \log\left(\sum_{d=1}^{D} \alpha_d \varphi(x; v_d)\right) \times \pi(x) dx.$$

Defining the posterior probability of x belonging to the dth cluster by $\rho_d(x; \alpha, v) \propto \alpha_d \varphi(x; \mu_d, \Sigma_d)$, consider

$$\int \sum_{d=1}^{D} 1_{(S(x)\geq\gamma)} \rho_d(x; \alpha^t, v^t) \log(\alpha_d \varphi(x; v_d)) \pi(x) dx$$

denoted by $L^t(\alpha, v)$. By concavity of the log, we have

$$L^t(\alpha, v) - L^t(\alpha^{t-1}, v^{t-1}) \leq \ell(\alpha, v) - \ell(\alpha^{t-1}, v^{t-1}),$$

so any increase in L^t would mean a bigger-or-equal increase in ℓ. At the same time, maximization of $L^t(\alpha, v)$ leads to a closed formula whenever φ belongs to an exponential family, e.g. in the Gaussian case, writing $\omega^t_{d,\gamma}(x)$ for $\rho_d(x; \alpha^t, \mu^t, \Sigma^t) \times 1_{(S(x)\geq\gamma)}$, we derive

$$\begin{aligned}
\alpha_d^{t+1} &= \int \omega^t_{d,\gamma}(x) g(x; v^t) dx, \\
\mu_d^{t+1} &= \frac{1}{\alpha_d^{t+1}} \int \omega^t_{d,\gamma}(x) x g(x; v^t) dx, \\
\Sigma_d^{t+1} &= \frac{1}{\alpha_d^{t+1}} \int \omega^t_{d,\gamma}(x)(x - \mu_d^{t+1})(x - \mu_d^{t+1})^T g(x; v^t) dx
\end{aligned}$$

whose empirical versions can be directly used as updates in Line 5 of Figure 1. Remark that this new algorithm is very similar to Population Monte Carlo schemes for integration (Robert & Casella, 2004).

3.3. Initialization via triangulation

CEM with mixtures fits the shape of the merit function, but it does not specify how to initialize the mixture components, a crucial step in practice. Recall that merit functions are zero at every training point and nonnegative everywhere. Their local maxima are "in between" the training points, so we should try to initialize the mixture components into these areas. The main idea is to triangulate the set of training points, and look at the modes in the interior of the simplices. We propose to initialize a component mean at the center of mass of every simplex and its covariance to sI, where s is the distance from the center of mass to the nearest corner.

We chose to use the Delaunay triangulation[1] because it keeps the simplices as compact as possible (Preparata & Shamos, 1988), in the sense that the interior of the circumsphere of any simplex contains no training point. In the Euclidean plane, this is equivalent to saying that the minimum angle of Delaunay's triangles is maximum over all triangulations: every triangle is then "as equilateral as possible". However, efficient implementations of Delaunay's triangulation for large datasets exist only for dimensions up to 6 (Hornus & Boissonnat, 2008). In small dimensions, we could afford to build an initial training set containing the 2^d corners of a hypercubic search domain C, whereas in cases where $d \geq 6$, we replaced the complete set of corners by a small number of uniformly sampled corners.

4. Experiments

We present two sets of experiments. Since the mixture CE method is a novel proposal, we first benchmark it on three common test functions taken from the optimization community (Table 1). We then compare the

[1] When $d \geq 3$, triangulation means "simplexification".

Table 1. The three benchmark functions.

Function	Expression	Bound
Sphere	$\sum_{i=1}^{d} x_i^2$	600
Rastrigin	$10d + \sum_{i=1}^{d} \left(x_i^2 - 10\cos(2\pi x_i)\right)$	5
Ackley	$20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)$	10

mixture algorithm with Delaunay initialization against grid search on single steps of EI optimization with different training set sizes in two and ten dimensions.

4.1. Benchmarking the mixture CE algorithm

In this section we experimentally compare EMNA (Larrañaga & Lozano, 2001) to our mixture CE algorithm, using mixtures of Gaussians. EMNA is a popular evolutionary algorithm based on the on-line fit of a Gaussian surrogate model, and our algorithm can thus be seen as a generalization of EMNA, allowing to launch several EMNAs in different loci of the search space and adapt to the fitness landscape while favoring the best components. A mixture version of EMNA called EMDA (Estimation of Mixture of Distributions Algorithm) was already mentioned in (Larrañaga & Lozano, 2001). Alhough our algorithm can be seen as an instance of EMDA, it is derived differently and it has the advantage of theoretically justifying the selection step by a ghost integration goal which is at the core of the CE method.

We started the optimization with a relatively large number of mixture components (10 for dimension 10, 20 for dimension 50), and gradually killed them when their mixing proportions went below a certain threshold (10^{-5} in our experiments). We used two different killing strategies. In the first strategy (red curves in Figure 4.1), we simply continued the procedure with the remaining components without replacing the killed ones. In the second strategy (green curves in Figure 4.1), we added a new component at the old component with highest mixing proportion, and assigned the new one a large initial variance to favor exploration around the current detected modes. We performed seven independent runs for each of the three algorithms. We plot the mean fitness obtained at the mean of the component with the highest mixing proportion versus the number of function evaluations. Shaded areas represent one standard deviation.

We chose common benchmark test functions in the continuous optimization community. We tried to reproduce the conditions of (Larrañaga & Lozano, 2001), initializing means uniformly over the initial ranges $[-B, B]^d$ where B is the specified bound in Table 1, taking $N = 2000$ points at each iteration and selecting the best $\mu = 1000$ points to compute the updates. We initialized all variances to 1. The three columns of Figure 4.1 depict our results on the Sphere, Rastrigin, and Ackley functions, respectively, the latter two being highly multimodal. The two columns correspond to dimensions $d = 10$ and $d = 50$. All functions are normalized to present a unique minimum at the origin.

Fitness graphs and spatial and eigenvalue-based diagnoses suggest that EMNA – with the degeneracy-avoiding update – finds the optimal mode after a reasonable number of function evaluations. Our mixture CE method seems to reach more quickly the best mode with either killing strategy: it automatically selects the best area according to its global model of the surface by focusing on the best components. We observed that after this pre-selection phase, the component means quickly concentrated on the best mode. After this step, the behavior was of course very similar to EMNA.

Let us insist on the fact that we used the same N and μ for EMNA and the mixture CEM, so updating a mixture costs the same price in function evaluations as updating a single distribution. That is why we think our algorithm can be considered as an automatic way of tuning the initialization of EMNA. Looking back to our original problem, this is exactly what we need in the context of EI optimization, as the EI landscape is possibly multimodal, and the mixture approach will allow us to visit the different modes, progressively select the best one, and finally sample only in the interesting area. As both killing strategies performed equally, we will use the first – no replacement – in what follows, for the sake of simplicity.

4.2. A comparison on single steps: the setup

To verify experimentally whether the mixture method is more efficient than a grid search on the EI optimization problem, we propose the following setup. We considered the domain $C := [-5, 5]^d$ with $d = 2, 10$, and we started by uniformly sampling $n' = 5, 20, 40$ points that we take as our training set, along with the (full or sampled) domain corners. The different values of n' represent different epochs in the final algorithm: as n' grows, we steer from an exploration phase to more advanced (exploitation) stage.

For each n', we optimized the EI criterion using grids of different resolution. Then we ran our mixture CE method with an identical budget, meaning that it was only allowed to perform as many EI function evalu-

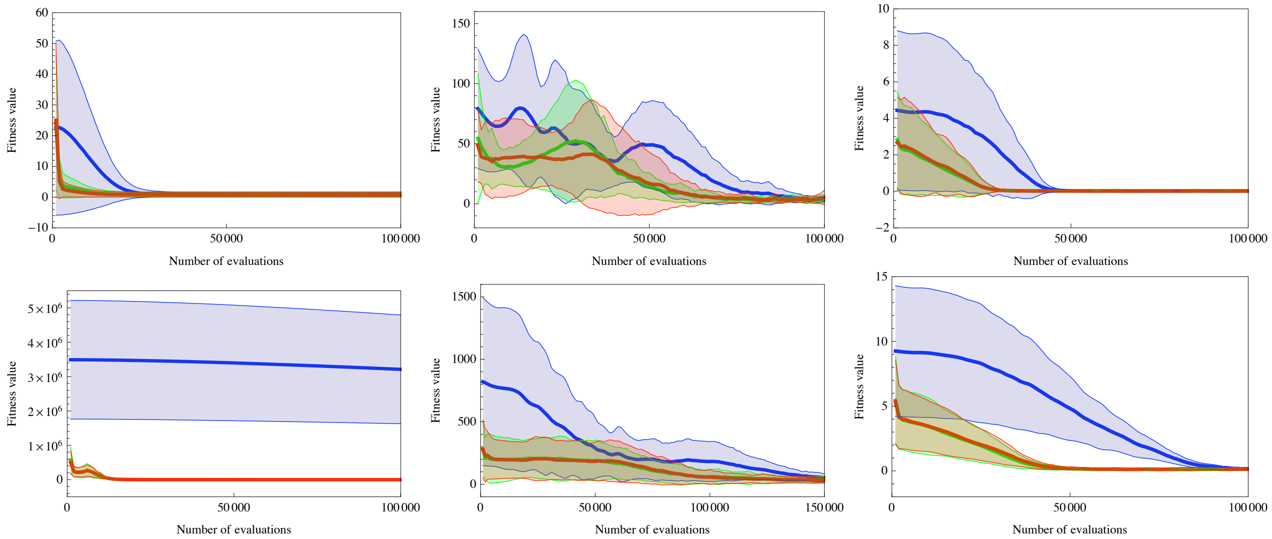

Figure 2. Empirical comparison of EMNA (blue) with two different killing strategies of our mixture CE algorithm (red and green, see the text for details) on the three benchmark functions (Sphere, Rastrigin and Ackley, from left to right) in dimensions 10 (top) and 50 (bottom). Thick lines represent the mean fitness values of the components with highest mixing proportion, while shaded areas represent one σ error bars.

ations as the number of points in the grid. For example, a 2D grid with a step size of 0.5 contains length$(-5 : 0.5 : 5)^2 = 441$ points, so the red point corresponding to 0.5 in Figure 4.2 is the value obtained by the mixture algorithm after 441 EI function evaluations. The algorithms were run several times (3 times for each grid of stepsize r, with a shift uniformly distributed in $[0, r]$, and 5 times for each budget); the mean values obtained are plotted in thick lines, with shaded areas representing one σ error bars.

The grid search becomes a real bottleneck in higher dimensions. The most popular solution is to replace the grid search by a Latin hypercube search, where the budget is a parameter. Figure 4.2 depicts a comparison of Latin hypercube search with our mixture search (in which we replaced the full set of corners by a subsample of 10), as detailed in Section 3.3.

The underlying fitness functions were the Sphere, Rastrigin and Ackley functions, respectively (Table 1). The covariance function used in the experiments was an isotropic squared exponential with noise, for which the hyperparameters where tuned by maximizing the marginal likelihood of the GP (Rasmussen & Williams, 2006).

4.3. A comparison on single steps: comments

The mixture search outperforms the grid on all test functions in 2D for small training set size, and shows outstanding robustness to budget reduction for every training set size on the most difficult task (Ackley's function). On the Sphere and Rastrigin's functions on medium and big size training sets, the mixture method remains competitive with the grid without outperforming it, while our method performs poorly on the Sphere function with the biggest training set.

Several empirical diagnoses can be invoked. The Sphere function is the easiest to interpolate, and very quickly, the EI landscape consists only of a thin peak near the best value obtained so far. For Ackley's function, even after 40 iterations, the structure of the function has not been explored enough, the support of the EI function is thus broader and allows for better estimation by the cross-entropy algorithm. Indeed, the elite sample of selected points is more representative of the EI function than in a case where almost all the samples fall in a very low fitness area of $C := [-5, 5]^d$.

In 10D, the comparison is even more favorable to the mixture method. It can clearly adapt to the landscape whereas the Latin hypercube sampling cannot. The Latin hypercube, and thus a grid, must contain a lot more points than the tested budgets to hope to find a decent mode. It is interesting to note that the only function that behaves differently is the Sphere, for which Latin hypercube sampling seems more efficient for very low budget. But the implied EI values are quite high and do not give evidence for a too peaky EI landscape as in 2D. We think the problem might be the number of points given to the mixture algorithm: there must be an optimal trade-off between the number of iterations and the number of points for a given budget. This is similar to CMA-ES (Hansen, 2006), another efficient search heuristics based on the

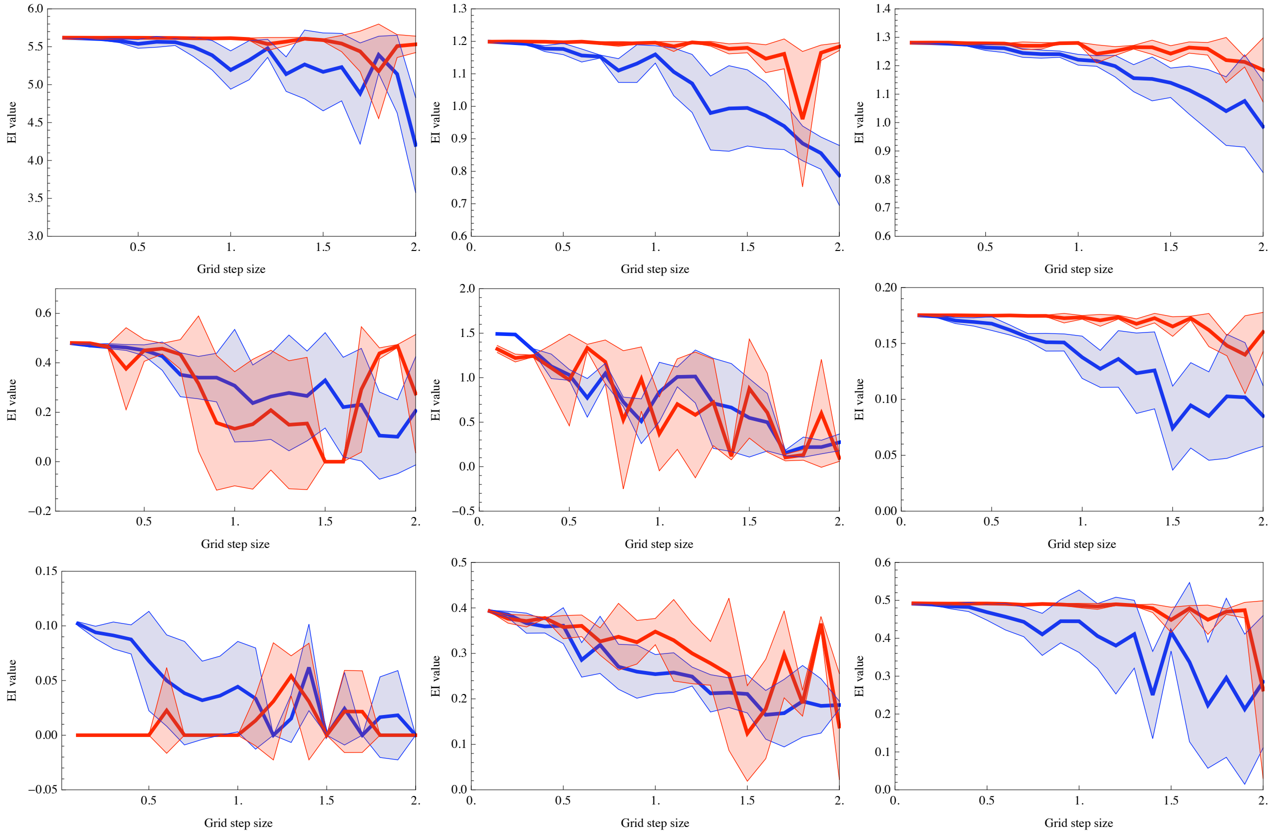

Figure 3. Empirical comparison of grid search (blue) and mixture search (red) on the three benchmark functions (Sphere, Rastrigin and Ackley, from left to right) in dimension $d = 2$ with different training set sizes $n = 4 + n' = 9, 24, 44$ (from top to bottom). Thick lines represent the mean of the best EI values obtained, while shaded areas represent one σ error bars.

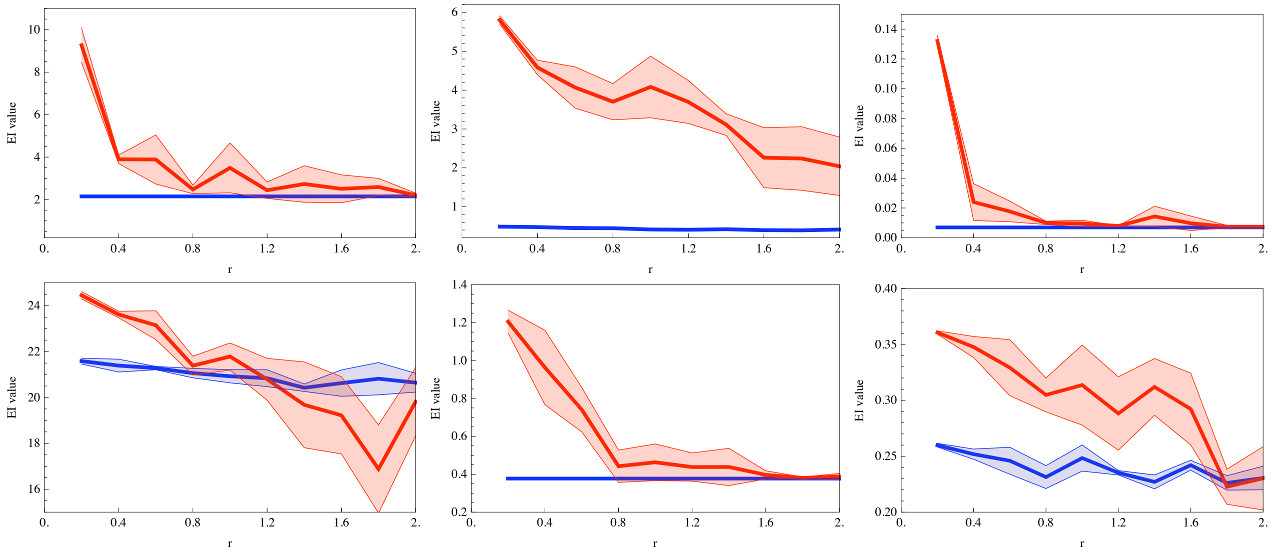

Figure 4. Empirical comparison of grid (blue) and mixture search (red) on the three benchmark functions (Sphere, Rastrigin and Ackley, from left to right) in dimension $d = 10$ with training set sizes $n = 10 + n' = 17, 25$, from top to bottom. The x-axis r value corresponds to a budget of $(1 + 10/r)^2$. Thick lines represent the mean of the best EI values obtained, while shaded areas represent one σ error bars.

evolutionary update of a single Gaussian. Notice that we forced in our tests the mixture search to perform at least five iterations, whatever the budget was (by decreasing the number of points per iteration).

5. Conclusion

We have derived a new adaptive search method based on mixtures to solve the problem of merit maximization in GP-based global optimization. We gave a sound theoretical basis through the link with the CEM, for which general convergence results are still in progress. Our method has been experimentally shown to compare favorably with grid search in 2D with noticeable robustness to budget reduction, and to globally outperform Latin hypercube sampling in 10D. We believe that the proposed method can be particularly useful in GP-based global optimization when the merit function is not analytical so it needs Monte Carlo sampling from the GP.

Although the basic setup of the mixture CE method is theoretically sound, we had to make several heuristic algorithmic choices when implementing the practical method (the number of sample points from the mixture distribution, the number of components, etc.). The triangulation initialization procedure is definitly a step that may be improved, especially in higher dimensions where the curse of dimensionality must be addressed. We could obviously accelerate the method by not throwing away the CE mixture components from one EI iteration to another, since it is likely that the EI surface would change significantly only around the newly added test point. This would eliminate the need of a heuristic initialization procedure in each iteration, however, it would probably make the procedure more sensitive to the birth/death policy of the mixture components.

An interesting direction to explore would be to further relate the method to Monte Carlo Markov chain (MCMC) integration. On the one hand, the optimized surrogate GP covariance could lead to an adaptive initialization of the CE mixture components, similarly to adaptive Metropolis-Hastings with Gaussian proposals (Haario et al., 1998). On the other hand, the birth/death policy could be governed by a principled procedure based on Reversible Jump MCMC techniques (Green, 1995).

Acknowledgments

We would like to thank three anonymous reviewers and Emmanuel Vazquez for their useful comments on a preliminary version of this paper. This work was supported by the ANR-07-JCJC-0052 grant of the French National Research Agency.

References

Green, P. J. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Haario, H., Saksman, E., and Tamminen, J. An adaptive Metropolis algorithm. *Bernoulli*, 7:223–242, 1998.

Hansen, N. The CMA evolution strategy: a comparing review. In Lozano, J.A., Larranaga, P., Inza, I., and Bengoetxea, E. (eds.), *Towards a new evolutionary computation. Advances on Estimation of Distribution Algorithms*, pp. 75–102. Springer, 2006.

Hornus, S. and Boissonnat, J. An efficient implementation of Delaunay triangulations in medium dimensions. Research Report RR-6743, INRIA, 2008.

Jones, D.R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

Larrañaga, P. and Lozano, J. (eds.). *Estimation of Distribution Algorithms*. Springer, 2001.

Mockus, J., Tiesis, V., and Zilinskas, A. The application of Bayesian methods for seeking the extremum. In Dixon, L.C.W. and Szego, G.P. (eds.), *Towards Global Optimization*. North Holland, New York, 1978.

Preparata, F.P. and Shamos, M.I. *Computational Geometry, an Introduction*. Springer-Verlag, 1988.

Rasmussen, C.E. and Williams, C. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2004.

Rubinstein, R. Y. and Kroese, D. P. *The Cross-Entropy Method*. Springer, 2004.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process bandits: An experimental design approach. NIPS Workshop on Adaptive Sensing, Active Learning and Experimental Design, 2009.

Villemonteix, J., Vazquez, E., and Walter, E. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 2006.