Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

ENHANCING PATTERN SEARCH FOR GLOBAL OPTIMIZATION WITH AN ADDITIVE GLOBAL AND LOCAL GAUSSIAN PROCESS MODEL

Qun Meng
Szu Hui Ng

Department of Industrial Systems Engineering and Management
National University of Singapore
SINGAPORE

ABSTRACT

Optimization of complex real-time control systems often requires efficient response to any system changes over time. By combining pattern search optimization with a fast estimated Gaussian Process model, we are able to perform global optimization more efficiently for response surfaces with multiple local minimums or even dramatic changes over the design space. Our approach extends pattern search for global optimization problems by incorporating the global and local information provided by an additive global and local Gaussian Process model. We further develop a global search method to identify multiple promising local regions for parallel implementation of local pattern search. We demonstrate our methods on a standard test problem.

1 INTRODUCTION

Significant advances in computing capacity have contributed to increase in computational efficiency of simulation models. These models are then more applicable for some real-time control systems which require effective response to any system changes over time. In this paper, we are interested in optimizing real-time control systems where the objective function $f(x)$ is continuous and multimodel. Let $\mathbb{X}_\Omega$ be a compact set in $\mathbb{R}^d$. The general target of global optimization is to find x^* that satisfies $f(x^*) = \min_{x \in \mathbb{X}_\Omega} f(x)$. For stochastic simulation models which are evaluated with the effect of random noise, we are interested in optimizing the expectation of the sampled observations, $\min_{x \in \mathbb{X}_\Omega} E(y(x))$, where $y(x)$ are sample observations.

In those real-time control systems whose behaviours change dynamically, a fast optimization algorithm is required because the systems need to effectively respond to those changes with new optimal settings in a short time. For example, in maritime transportation, the Safe Sea Traffic Assistant (S^2TA) (Pedrielli et al.) applies a look-ahead approach with an agent based simulation model (ABM) to detect potential conflicts/collisions for vessels in heavy traffic regions. At any point, it looks 10 minutes ahead to determine the safety of the current trajectory and detect potential conflicts. If a potential conflict of high risk is detected on the pre-specified trajectory of an own vessel, the optimizer is called. Then the optimizer needs to find an alternative trajectory that minimizes the probability of conflict within 5 minutes given the current and predicted traffic conditions.

To solve such problems where response time is critical, direct optimization algorithms like pattern search (Torczon 1997, Taddy et al. 2009) can be applied as their speed is essential to conduct extensively fast search. On the other hand, the global and local information with a metamodel and a metamodel based optimization algorithm can also be very useful and informative (Jones et al. 1998, Picheny et al. 2013, Quan et al. 2013), especially for a large design space or highly complex functions. This is because they can help to drive the search to the correct optimal region for highly complex functions.

Many direct search methods have been used fruitfully in conjunction with metamodels for optimization problems with complex computer simulations. The asynchronous parallel pattern search (APPS) and the

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

treed Gaussian Process (TGP) model are combined to generate a set of candidate locations that are queued for evaluations (Taddy et al. 2009). The mesh adaptive direct search (MADS) also uses the TGP as a surrogate and to evaluate the EI criterion (Gramacy and Digabel 2015). Another common strategy for metamodel assisted direct search methods is to construct a coarse metamodel in the entire design space and use it to identify promising local regions. Then a more refined metamodel can be built in smaller local regions. It is then possible to explore several local regions simultaneously (Booker et al. 1999). The adaptive response surface method (ARSM) (Wang et al. 2001) disregarded regions with large function values as predicted by a surrogate and generated experimental designs using central composite designs (Montgomery 1991) in the reduced region. The predictive uncertainty, however, is not well considered when reducing the design space based on the surrogate predictions.

A more efficient optimization algorithm can incorporate parallel implementation of local pattern search. By generating multiple distinct points in each iteration based on the predictions from Gaussian Process model, the pattern search better exploits different local areas simultaneously. Sóbester et al. (2004) essentially parallelized the EGO methods (Jones et al. 1998) by generating multiple evaluation points that have the best local maximums of the EI functions. Ginsbourger et al. (2009) first introduced the multivariate Expected Improvement (q-EI) and implemented it via Monte Carlo sampling. The implementation of q-EI is further studied by Clark and Frazier (2012) and Chevalier and Ginsbourger (2013).

In this paper, we propose an enhancing pattern search algorithm for the global optimization problem. The general idea is to generate global search patterns based on the global and local information from an Additive Global and Local Gaussian Process (AGLGP) model and then to locally generate evaluation points based on local search patterns. We call it parallel global and local optimization (PGLO) algorithm. Moreover, pattern search is used as an example when deriving the PGLO algorithm. The choice of pattern search scheme is made on the grounds of its specific search patterns. It is noted that PGLO is not restricted to the pattern search algorithm, in fact any direct search methods can be adopted. Here we consider the multiple-core processors on personal computers or servers only as they are more readily available to general users. Hence it is assumed that the time to load simulations to different processors and transmitting data among processors is almost negligible. The rest of the paper is organized as follows. In section 2, we review the background of pattern search and the AGLGP model, and discuss desirable search patterns for global optimization. A simple one-dimension problem is then used to illustrate these desired properties. The details of the parallel global and local search algorithm are provided in section 3. Numerical results and conclusions are given in sections 4 and 5.

2 BACKGROUND AND OVERVIEW

2.1 Pattern Search

Pattern search is one of the direct search methods applied directly into the simulator to address problems whose objective function has no closed form or analytical solutions are intractable. By a predefined pattern of points, it has been shown to perform well in small local areas (Torczon 1997). Pattern search algorithms are characterized by their meshes and polling conditions. A mesh is a lattice on which the search for an iterate is restricted. At each iteration k, three basic steps are executed:

1. Generate a set of trial points Q_k within a mesh M_k around the current best point x_k.
2. (a) Obtain a set of function evaluations F_k from computer models. If $\exists x_{k+1} \in M_k$ such that $f(x_{k+1}) < f(x_k)$, the search is successful.
   (b) Else, polling conditions are applied to refine the mesh. Generally the mesh M_{k+1} is obtained by halving the mesh size, i.e. $M_{k+1} = M_k/2$. Repeat Step 1.
3. Update the best point by x_{k+1}.

Essentially, pattern search performs the search using a predefined ”pattern” of points that are independent of objective functions f and the design points that have been observed. Hence, the computational time for generating trial points is almost negligible and the majority of the computational time is spent on function evaluations. In this way, it efficiently exploits a small area for a local optimal solution with a shrinking mesh size, but it may not be able to sufficiently explore the entire space.

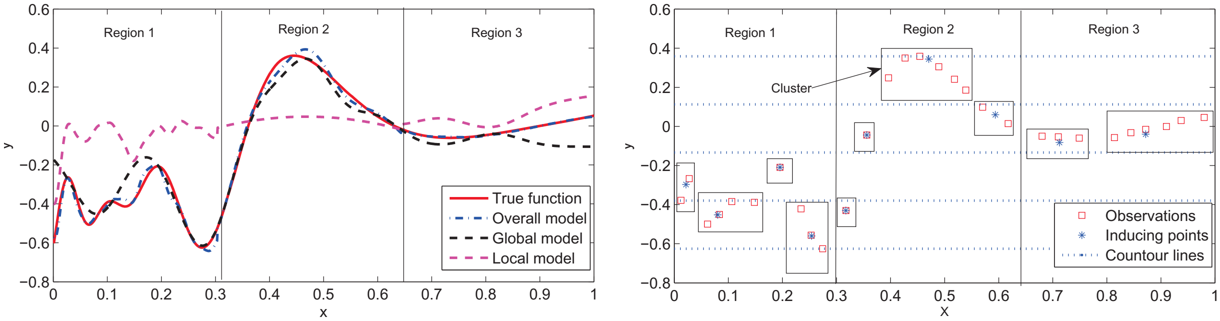
2.2 AGLGP Model

The AGLGP model (Meng and Ng 2015) is a flexible surrogate for stochastic computer models. It attempts to capture the overall global spatial trend and the local trends of the responses separately, to enable more accurate modeling of the surfaces that are nonstationary in both the underlying function and the stochastic noise. The additive structure of the model reduces the computational complexity in model fitting, and allows for more efficient predictions with large data sets.

Specifically, the AGLGP model assumes that the stochastic simulation response can be modeled as a realization of a random process

$$y(x) = f(x) + \varepsilon(x) = f_{global}(x) + \sum_{k=1}^{K} w_k f_{local}^k(x) + \varepsilon(x), \quad w_k = \begin{cases} 1, & x \in \mathbf{R}_k \\ 0, & x \notin \mathbf{R}_k \end{cases}$$

where $f(x)$ is the deterministic mean function of the stochastic response. The random noise $\varepsilon(x)$ has a normal distribution $\varepsilon(\mathbf{x}) \sim N(0, \sigma_\varepsilon^2(x))$, which is independent and identically distributed across replications and uncorrelated at different locations. The mean function $f(x)$ can be further decomposed to a global model $f_{global}(x)$, which models the global trend, and K local models with each local model $f_{local}^k(x)$ modeling the residual process that is unexplained by $f_{global}(x)$ in local region $\mathbf{R}_k$, where $\cup_{k=1}^K \mathbf{R}_k = \mathbb{X}_\Omega$. To capture the global trend, a set of m inducing points are used (where $m \ll n$). The GP models $f_{global}(x)$ and $f_{local}^k(x)$ are assumed to be piece-wise independent.



(a) The global, local and overall model.

(b) Clustering and local regions.

Figure 1: Plot of function $y(x) = \sin(30(x-0.9)^4)\cos(2(x-0.9)) + (x-0.9)/2$.

Figure 1 illustrates the general idea of the AGLGP model, which is an additive combination of a global model and local models. To fit the model, the entire space is divided into several nonoverlapping local regions via classification techniques like Support Vector Machine (SVM) (as shown in Figure 1b). To obtain inducing points to fit the global model, within each local region, we further separate points into clusters based on a set of equally spaced contour lines from the minimum observation value to the maximum observation value. This avoids clustering points that have large variability together. The purpose of this clustering of similar observations (in both x and y space) is to obtain a cluster centroid which can reasonably represent the observations in the cluster and be a representative point (called an *inducing point*) of the cluster. With a good spread of evaluation points in the domain, the inducing points (although fewer) will also have a good spread in the entire space and will be able to capture the global trend.

It is assumed that $f_{global}(x)$ can be modeled by a deterministic GP model with a mean β_0 and covariance $\sigma^2 r_g(x_i - x_j, \boldsymbol{\theta})$, where σ^2 is the variance of the global component and $r_g(\cdot)$ is the correlation structure with a sensitivity parameter $\boldsymbol{\theta}$. Given the set of inducing points and the global evaluations $\mathbf{y}_g$, the best linear unbiased global predictor can then be written as

$$\widehat{y}_{global}(x) = \beta_0 + \mathbf{g}'\mathbf{G_m^{-1}}(\mathbf{y}_g - \mathbf{1}'\beta_0), \tag{1}$$

where $\mathbf{g} = (g(x-x_g^i), \cdots, g(x-x_g^m))$, $\mathbf{G_m}$ is an $m \times m$ covariance matrix with ijth element $g(x_g^i - x_g^j)$. The global predictor interpolates $\mathbf{y}_g$ since $\widehat{y}_{global}(\mathbf{x}_g^j) = \beta_0 + \mathbf{e_i'}(\mathbf{y}_g - \mathbf{1}'\beta_0) = y_g^i$. With the fitted global model, the global predictors at $\mathbf{x}$ are $\widehat{\mathbf{y}}_{global} = (\widehat{y}_{global}(x_1), ..., \widehat{y}_{global}(x_n))$. The residuals, which include both the residuals from the signal function and the random noise, are then obtained by $\mathbf{y_l} = \mathbf{y} - \widehat{\mathbf{y}}_{global}$ and modeled by another stochastic GP model $y_{local}(x) = \sum_{k=1}^{K} w_k f_{local}^k(x) + \varepsilon(x)$, where $f_{local}^k(x) \sim N(0, \tau_k{}^2 r_k(x_l^i - x_l^j, \boldsymbol{\alpha}_k))$, and $\boldsymbol{\alpha}_k$ is the sensitivity parameter for the local model. We assume that the residual process is correlated within a local region while independent across the regions, so different correlation functions are allowed in different regions. This enables the flexibility to capture nonstationarity in the process. Given K local regions $\mathbf{R} = (\mathbf{R_1}, ...\mathbf{R_K})$, the local predictor is given by

$$\widehat{y}_{local}(x) = \mathbf{l_k}'(\mathbf{L_k} + \boldsymbol{\Sigma_\varepsilon})^{-1}\mathbf{y_l^k}, \quad \forall x \in \mathbf{R}_k, \tag{2}$$

where $\mathbf{l_k} = (l_k(x - x_l^i), ...l_k(x - x_l^{r_k}))$ and $\mathbf{L_k}$ is covariance matrix with (jh) element $l_k(x_l^h - x_l^j)$, $\forall x_l^j, x_l^h \in \mathbf{x_l^k}$. The quantities $\boldsymbol{\Sigma_\varepsilon} = diag(\hat{\sigma}_\varepsilon^2(x_l^1), ..., \hat{\sigma}_\varepsilon^2(x_l^{r_k}))$ and $\hat{\sigma}_\varepsilon^2(x_l^i)$ can be estimated from the sample variance. So the overall AGLGP predictor for $\forall x \in \mathbf{R}_k$ can be expressed by

$$\widehat{y}(x) = \widehat{y}_{global}(x) + \widehat{y}_{local}(x) = \beta_0 + \mathbf{g}'\mathbf{G_m^{-1}}(\mathbf{y}_g - \mathbf{1}'\beta_0) + \mathbf{l_k}'(\mathbf{L_k} + \boldsymbol{\Sigma_\varepsilon})^{-1}\mathbf{y_l^k}. \tag{3}$$

As $\mathbf{y_g}$ and $\mathbf{y_l^k}$ are latent processes that cannot be observed directly, the predictive distribution of any input x can be derived by integrating out the random variable $\mathbf{y_g}$ and $\mathbf{y_l^k}$, see (Meng and Ng 2015).

2.3 Desired Properties of Global and Local Search Pattern

Although the AGLGP model, as a fast surrogate, can provide efficient predictions with large data sets, for optimization problems where observations tend to be clustered in promising local regions, the local estimation can still become computationally challenging in regions with dense observations. Hence, it will be more efficient (and desirable) to exploit local promising regions with fast local search patterns, and then generate more detailed and guided global search patterns with the AGLGP model. A desirable parallel framework should at least have the following properties,

- evaluate multiple evaluation points quickly around the current best local optimal solution to potentially find better solutions around the current best one.
- evaluate multiple evaluation points simultaneously around different local optimal solutions to better explore the whole space for a global optimal solution.

To better understand the desired properties of a parallel framework, we first look at the following example where the noisy test function is given as

$$y(x) = (2x + 9.96)cos(13x - 0.26) + \varepsilon(x) \tag{4}$$

and $\varepsilon(x)$ is the noise function that is normally distributed with mean 0 and variance $\sigma_\varepsilon^2 = 4$ and $x \in [0, 1]$. The test function has a local minimum at 0.2628 and a global minimum at 0.7460. Here we adopt a single region for simple illustration. An initial seven points Latin Hypercube Design (LHD) experiment is conducted with 10 replications at each point and the initial AGLGP model is fitted as shown on the left plot of Figure 2. Based on the initial AGLGP model fit, we generate a global search pattern from a modified expected improvement (mEI) function (Quan et al. 2013) that identifies a local optimal area close

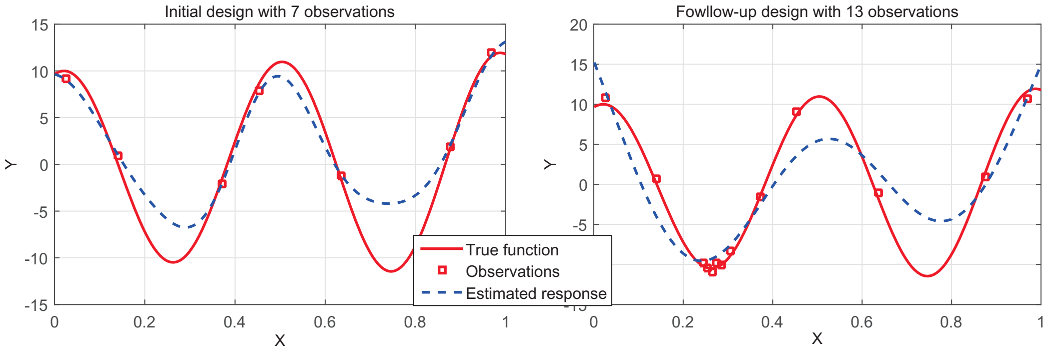


Figure 2: AGLGP model fit with design point selected by pattern search.

to $x = 0.3$. A local pattern search is then applied to select the six subsequent points that exploit this local optimal area (the right plot of Figure 2). With this global search pattern, the algorithm is able to identify the local optimal area more efficiently.

The overall search, however, has mistakenly identified the global optimal point after the six iterations because of the low observations near the local minimum on the left. This problem could be mitigated if multiple local minimums can be identified and exploited simultaneously. Based on the same initial fit, the mEI function is displayed in the left panel in Figure 3. Suppose the two local maximums of the mEI function are selected for simultaneous evaluations, followed by local pattern search that exploits around each of these two points. As we can see in the right panel of Figure 3, the global optimal solution is much improved with two series of local pattern search (searching for the two local minimums at the same time). In this example, the two local maximums of the mEI function are located around the two local minimums of the function too. In general, this will not be the case. The maximum mEI locations can indicate areas with high spatial uncertainty but not low (or optimal) predictions. In these cases, however, pattern search can still guide the search towards local minimums because pattern search can continue exploiting for better solutions. Hence, they can combine to better exploit multiple local minimums simultaneously.

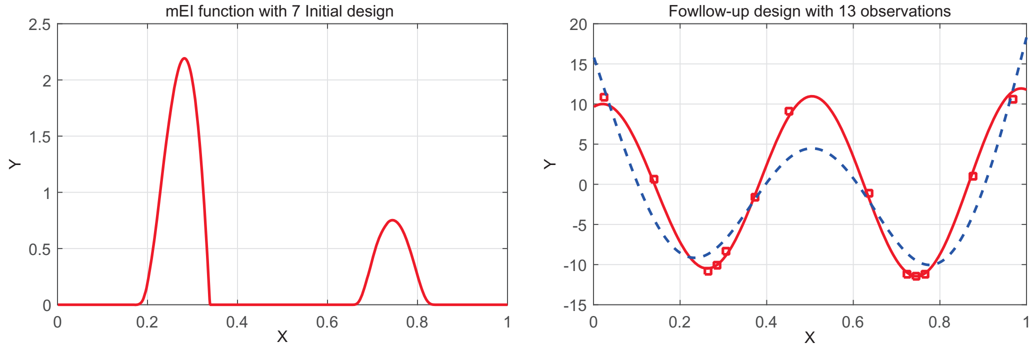


Figure 3: mEI function and AGLGP model fit with design point selected by multistart pattern search.

To incorporate these desirable properties in the search, we propose the PGLO algorithm to consist of a global search stage to identify multiple promising local regions and a parallel local search stage searching for multiple local optimal solutions. The evaluations in the local search stage are initialized from a set of global search patterns that have either low predictions or high spatial uncertainty. Then, local pattern search is applied to select additional follow-up evaluation points to quickly exploit around each of those initial points for better local optimal solutions. In section 3, we explain the development of PGLO in detail. Selecting the best local maximums of the mEI function is a straightforward and intuitive approach to select the initial evaluation points, more sophisticated selection criterion can be adopted.

3 PARALLEL GLOBAL AND LOCAL OPTIMIZATION WITH AN AGLGP MODEL

In this section, we present the framework for a master-worker parallel global and local optimization (PGLO) algorithm. It is an efficient algorithm that can quickly select multiple promising local regions and focus on searching for multiple local optimal solutions in each promising local region. Here, we assume one master processor and q available worker processors. The details of this algorithm are introduced in the following subsections.

We begin by fitting the AGLGP model in Equation (3) from an initial set of points (typically from a space-filling experimental design like Latin Hypercube Design (LHD)). After the initial fit, each subsequent iteration of the algorithm is composed of a global search stage based on the global model (Equation (1)) that exploits the entire space globally for several promising local regions, and a local search stage based on the overall model (Equation (3)) that exploits within each promising local region simultaneously. Considering a total budget of T, in each iteration t, the budget exhausted is denoted as B_t, where $B_t = B_{t,s} + B_{t,a}$. The budget $B_{t,s}$ is for the parallel local search stage to find new design points (preferred as a multiplier of q) and the budget $B_{t,a}$ is for the allocation stage to allocate replications to existing design points. Here we simply assume a constant budget B_t over iterations. n_{max} is the maximum number of new evaluation points at each iteration. p_{max} is the maximum number of new evaluation points selected from pattern search before refitting of the local models. An overview of the algorithm is given in Table 1.

3.1 Global Search Stage

In the global search stage in Step 3, PGLO focuses on identifying promising local regions to focus the search. It randomly generates n_c candidate points Ω_g and selects q points that maximize the multi-point global expected improvement from Ω_g. Each local region $\mathbf{D_k}$ with q_k selected candidate points is then allocated with q_k processors for parallel local search, where $\sum q_k = q$. By smoothing out the localized features in each local region, the global search avoids putting efforts exploiting one single local region that has multiple neighboring local optimal solutions from the start.

3.1.1 Multi-points Global Expected Improvement

Here we apply a multi-point global expected improvement (q-gEI) criterion as an extension of the global expected improvement (Meng and Ng 2016), which is defined as

$$gEI(x) = E\{\max(y_{gmin} - y_g(x), 0)\} \cdot \frac{1}{1 + e^{n_i/v - 5}}. \tag{5}$$

where y_{gmin} is the lowest predicted global evaluation,and $y_g(x)$ is a normal random variable with mean given by the global predictor $\hat{y}_g(x) = \mu + \mathbf{g}'\mathbf{Q_m^{-1}}\mathbf{G_{mn}}(\Lambda + \Sigma_\varepsilon)^{\mathbf{-1}}(\mathbf{y} - \mathbf{1}'\mu)$ and global variance given by $\hat{s}_g^2(x)) = [G_{nn} - \mathbf{g}'\mathbf{G_m^{-1}}\mathbf{g}] + \mathbf{g}'\mathbf{Q_m^{-1}}\mathbf{g}$ (Meng and Ng 2016). Based on the global model which smooths out the localized features in each local region, the predictive global trend can provide a guide towards the promising solutions in the design space. The additional factor is designed to account for observations around each point that has been aggregated away with the inducing points. It serves as a penalty for points with many observations around it, giving diminishing returns overall for points with increasing number of observations around it.

The multi-point gEI criterion (q-gEI) is the expectation of the improvement brought by the q points,

$$q\text{-}gEI(x_1, ..., x_q) = \mathbb{E}[\max\{(y_{gmin} - y_g(x_1))^+ \cdot \frac{1}{1 + e^{n_1/v - 5}} \cdots, (y_{gmin} - y_g(x_q))^+ \cdot \frac{1}{1 + e^{n_q/v - 5}}\}] \tag{6}$$

The Multi-point Expected Improvement (also called qEI) criterion was first defined by Schonlau et al. (1998). In most situations, maximizing the expectation of Equation (6) of dimension $q \times d$ requires demanding computational effort. To address this, we approximate the maximization of Equation (6) by selecting the q points $x_1^g, \cdots, x_q^g$ sequentially (Ginsbourger et al. 2010) through optimizing the gEI function.

Table 1: Overview of PGLO.

```
Parallel Global and Local Optimization Algorithm
Step 1:  (Initialization) Run a size $n_0$ space filling design, with $r_{min}$ replications allocated to each
         point. Total initial replications $B_0 = n_0 r_{min}$. Set $t = 0$.
Step 2:  (Validation of overall model) Fit an AGLGP response model to the set of sample means
         and variances, and use cross validation to ensure that the AGLGP prediction is satisfactory.
While the available replications $A = T - \sum_{i=0}^{t} B_i > 0$, $t = t+1$
Step 3:  (Global Search Stage) Generate $n_c$ candidate points $\Omega_g$. Select $q$ points $x_1^g, \cdots, x_q^g$ from $\Omega_g$
         based on the global model and a multi-point global criterion. Identify the promising local
         regions, where $x_1^g, \cdots, x_q^g$ are located. Each promising local region $\mathbf{D_k}$ with $q_k$ selected
         candidate points is allocated with $q_k$ processors for the local search and evaluation. Hence
         $q_1 + \cdots + q_K = q$.
Step 4   (Parallel Local Search Stage) While $n_t < n_{max}$ and $A > 0$,
         (Fit/Update local models) Fit or update the local models in all the promising local regions.
         (Generate Candidate Points) Randomly generate $n_l$ candidate points $\Omega_l^k$ independently in
         each promising local region $\mathbf{D_k}$.
         (Select the Initial Evaluation Points) In each promising local region $\mathbf{D_k}$, select $q_k$ points
         $x_{n_t}^1, \cdots, x_{n_t}^{q_k}$ from the candidate points $\Omega_l^k$ based on the overall model and a multi-point
         local search criterion. Simultaneously evaluate at all $q$ selected points $x_{n_t}^1, \cdots, x_{n_t}^q$ from
         all promising local regions with $r_{min}$ replications on $q$ processors. Hence the number of
         observed points $n_t = n_t + q$.
         (Select the Follow-up Evaluation Points by Pattern Search) Set $p_t = 0$,
         While $n_t < n_{max}$ and $p_t < p_{max}$, the $q$ processors continue to evaluate at $q$ new points from
         the predefined search patterns. Hence, $n_t = n_t + q$ and $p_t = p_t + q$.
         Break follow-up selection if a stopping criterion is satisfied. end
         end
Step 5:  (Allocation Stage) Allocate $B_{t,a}$ replications for additional evaluations among all evaluated
         points.
end
Step 6:  (Return the Optimal Solution) Return the point with the lowest sample mean.
```

As we expect the global expected improvement function to change with each new point $x_i^g, i = 1, \cdots, q-1$ added, to deliver a set of distinct points spread out across the entire space, we update the global model for each new x_i^g that has already been selected. To select x_{i+1}^g that optimizes the updated *gEI* function, the global model is updated given the new 'observation' $y(x_i^g)$ based on the *kriging believer* assumption (Ginsbourger et al. 2010), which assumes the observation value $y(x_i^g)$ equals to the overall AGLGP model prediction $\hat{y}(x_i^g)$ in Equation (3).

To ensure that q distinct candidate points are selected, if the global model does not change after each new x_i^g is added, a minimum number of *artificial points* $\tilde{n}_i$ are assumed to be selected around x_i^g by local search to affect the penalty term such that

$$gEI(x_i^g) = E[(y_{gmin} - y_g(x_i^g))^+] \cdot \frac{1}{1 + e^{(n_i + \tilde{n}_i)/v - 5}} \le \max_{x_c \in \Omega_g \setminus x_i^g} gEI(x_c), \qquad (7)$$

This is a reasonable assumption because the local search stage will help to better exploit a promising area with a set of surrounding points around. This is also done as with a sparse distribution of the global candidate points, each new added point x_{i+1}^g with maximum *gEI* can indicate a different local region from the previous added point x_i^g. This avoids putting all effort in a single local region and allows the algorithm to spread the searching effort in multiple local regions.

3.2 Parallel Local Search Stage

Next we provide the details on the parallel local search stage. Once the promising local regions are selected by the global search, the local search stage then searches more extensively within each promising local region for better solutions. In this stage we first adopt a multi-point modified expected improvement function to select an initial set of q evaluation points for simultaneous evaluations, with q_k points selected from each promising local region $\mathbf{D_k}$. A follow-up local pattern search is then conducted for additional evaluation points exploiting around each of the initial q points.

3.2.1 Initial Evaluation Points Selection

The role of the initial points is to start the local search from promising areas, which have either low predictions or higher spatial uncertainty, and quickly progress to better solutions in the local region with additional follow-up evaluation points. In order to achieve this, we propose a multi-point modified expected improvement function as an extension of the *mEI* criterion (Quan et al. 2013). The initial q_k evaluation points in the promising local region $\mathbf{D_k}$ are selected by maximizing the q_k-*mEI* function defined as

$$q_k\text{-}mEI(x_1,...,x_{q_k}) = \mathbb{E}[\max\{(y_{min}-z(x_1))^+\cdot,...,(y_{min}-z(x_{q_k}))^+\}] \tag{8}$$

where y_{min} is the predicted response at the sampled points in the local region $\mathbf{D_k}$ with the lowest sample mean, and $z(x)$ is a normal random variable with mean given by the AGLGP predictor in Equation (3) and variance given by spatial prediction uncertainty $\hat{s}_z^2(x) = L_{nn} - \mathbf{l}'\mathbf{L_n^{-1}l}$. Instead of searching the global optimal solution for the batch $(x_1,...x_{q_k})$, we approximate it again by optimizing the q_k points sequentially one at a time with each step maximizing the $mEI(x)$,

$$x_1^* = \arg max_{x\in\Omega_l} mEI(x) = \arg max_{x\in\Omega_l} E_1(\max[y_{min}-z(x),0], \tag{9}$$

$$x_{i+1}^* = \arg max_{x\in\Omega_l} mEI(x) = \arg max_{x\in\Omega_l} E(\max[y_{min}-z(x),0]|x_1^*,...,x_i^*,y_1,...,y_i). \tag{10}$$

Here we also optimize the *mEI* with respect to a set of candidate points Ω_l, which are uniformly distributed in the local region (Regis and Shoemaker 2007). To select the optimal point x_{i+1}^*, the local model and the *mEI* function are updated with the selected points $x_1^*,...,x_i^*$ and their 'observations' $y_1,...,y_i$. We approximate the 'observations' $y_1,...,y_i$ equal to the AGLGP model prediction $\hat{y}(x_1),...,\hat{y}(x_i)$ in Equation (3). As the updated model variance $\hat{s}_z^2(x) = L_{nn} - \mathbf{l}'\mathbf{L_{n+i}^{-1}l}$ is reduced around the selected points $x_1^*,...,x_i^*$, it avoids selecting new points near the selected ones.

3.2.2 Follow-up Evaluation Points Selection

In this step, we continue the local search by selecting additional follow-up evaluation points to quickly exploit the promising areas around each of the q initial points for better solutions in the promising local region. To achieve this, we apply q pattern search to quickly select q new evaluation points from the pre-defined search patterns. The pattern search is initialized from each of the q initial points obtained as described in section 3.2.1, and it is sequentially repeated with q new selected evaluation points. Although the pattern search can move efficiently towards a local optimal solution in a small area, it only converges at a local optimal solution. As there may exist more than one local optimal solution in each local region, it is not desirable for the algorithm to converge only to a local optimal solution instead of a global optimal solution in the local region. Hence, to avoid being trapped in a local optimal, we stop the pattern search when a stopping criterion is satisfied and require the algorithm to reselect a new set of initial points that explore the entire local region for potentially better local optimal solutions.

Stopping Criterion In the follow-up pattern search step, the mesh size of the pattern search can keep shrinking as the search progresses and it already identifies a local optimal point, $M_{t_k+1} = 1/2M_{t_k}$. To avoid spending unnecessary budget for unsuccessful searches, we set a stopping criterion where we stop the pattern search when $M_{t_k} \leq M_{min}$, where M_{t_k} is the current mesh size and M_{min} is a predefined minimum mesh size (Torczon 1997). When one of the q pattern search stops at a local minimum before p_{max} budget is exhausted in that promising local region $\mathbf{D}_k$, we stop the follow-up points selection step and require the algorithm to return to the initial points selection step to select a new set of initial points. The local models are then updated in the promising local regions, and new initial evaluation points are selected by maximizing the updated *mEI* function to identify new promising areas.

This stopping criterion allows the pattern search to escape from the current local optimal solution to a potentially better local optimal solution before exhausting all the p_{max} budget. The new initial points can be selected with either low prediction or high predicted mean square error. If the new initial points have low predictions, the pattern search continues exploiting the current promising areas, and if the new initial points have high predicted mean square error, the pattern search starts to exploit new promising areas.

3.3 Allocation Stage

In this stage, we adopt an allocation strategy to evaluate the best few optimal solutions after the local search with an additional number of replications $B_{t,a}$. Specifically, the Optimal Computing Budget Allocation (OCBA) approach (Chen and Lee 2010) is adopted for all the already evalated n points. Suppose each point x_i has a sample mean given by $\bar{y}_i$ and a sample variance $\sigma_\varepsilon^2(x_i)$, then according to Theorem 1 provided by (Chen and Lee 2010), the Approximate Probability of Correct Selection (APCS) can be asymptotically maximized (as the available budget tends to infinity) when

$$\frac{N_{i,b}}{N_{j,b}} = \left(\frac{\sigma_\varepsilon(x_i)/\Delta_{b,i}}{\sigma_\varepsilon(x_j)/\Delta_{b,j}}\right)^2, i,j \in \{1,2,...,n\}, i \neq j \neq b \tag{11}$$

$$N_{b,b} = \sigma_\varepsilon(x_b)\sqrt{\sum_{i=1,i\neq b}^{n}\left(\frac{N_{i,b}}{\sigma_\varepsilon(x_i)}\right)^2}$$

where $\bar{y}_b$ is the lowest observed sample mean in the entire space, $N_{i,b}$ is the number of simulations allocated to point x_i, and $N_{b,b}$ is the number of simulations allocated to the point x_b with the lowest sample mean. Hence, $\sum_i N_{i,b} = B_{t,a}$. $\Delta_{b,i} = \bar{y}_i - \bar{y}_b$. The OCBA technique is able to allocate additional replications to the points with low sample means and high sample variances in order to distinguish the best point from the other competitors. With additional replications allocated at those potential optimal points, the model estimation around those points can be further improved, which in turn can help the global and local search criterion to make better selection in the subsequent iteration. After the allocation stage, the sampled point with the lowest sample mean is selected as the location of the current best response.

4 NUMERICAL STUDIES

In this section, we compare the performance of PGLO (which incorporates the AGLGP model and the efficient AGLGP model-based global and local searching criterion to drive the local pattern search) with other parallelized pattern search techniques.

Here, we compare MultPPS-LHS, using a direct application of multistart parallel pattern search (*MultPPS*) initialized with Latin Hypercube Sample (LHS), and MultPPS-qEI, with *MultPPS* initialized from the multi-point expected improvement (q-EI) of the AGLGP model. MultPPS-LHS is used as a benchmark algorithm for the direct application of pattern search method, while MultPPS-qEI is compared to evaluate the performance of PGLO which incorporates pattern search within a global and local search algorithm against a single level search with q-EI. To replicate the situation of a fast simulation model, a wait time of 0.01 seconds is added to each function evaluation. The sequential pattern search is limited with 200 evaluations to avoid spending too much budget in any sub-optimal areas.

The performance of a parallel optimization algorithm is measured by the wall clock time consumed to find a reasonable solution within a specific level of accuracy. With known global optimum g^*, the relative error of a reasonable solution g_{best}, $|g^* - g_{best}|/|g^*|$, should be less than 1%. Another commonly used measure for the parallel performance is the *speedup*, which is defined by the time required for the sequential optimization on one processor $T(1)$ divided by the time required for the parallel optimization on q processors $T(q)$, i.e., $SP = T(1)/T(q)$. We adopt the following example from Sun et al. (2014),

$$\max_{0 \leq x_1, x_2 \leq 100} g(x_1, x_2) = 10 \cdot \frac{sin^6(0.05\pi x_1)}{2^{((x_1-90)/50)^2}} + 10 \cdot \frac{sin^6(0.05\pi x_2)}{2^{((x_2-90)/50)^2}}. \tag{12}$$

g has a global optimum of $g^*(90,90) = 20$ and the second best local optimum is $g(70,90) = g(90,70) = 18.95$ (see Figure 12). We introduce a noise term that is normally distributed with mean 0 and variance $\sigma_\varepsilon^2(x_1,x_2) = 3(1+x_1/100)^2(1+x_2/100)^2$.

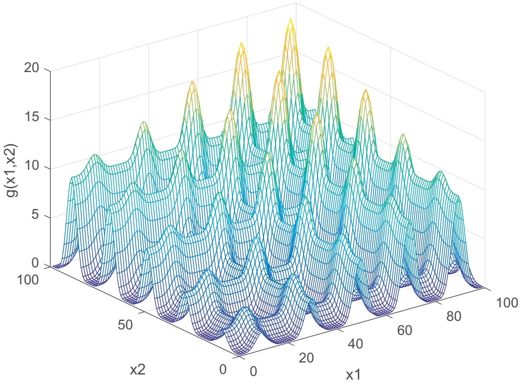


Figure 4: $g(x_1,x_2)$ function.

Table 2 presents the average wall clock time to get a solution with a relative error $|g^* - g_{best}|/|g^*| < 1\%$ over 30 macro-replications, where with $g^* = 20$ the observed global optimum g_{best} end up within $[19.8, 20.2]$. As the second best local optimum is only 18.95, g_{best} is at least able to identify the global optimal area. Even though the simulation runs as fast as 0.01s, the direct application of pattern search (MultPPS-LHS) performs the least efficiently for any number of processors. Because the objective function has 25 local optimal solutions, the MultPPS-LHS can not explore the entire space sufficiently without global information. However, as the number of processors increases, the MultPPS-LHS catches up with better exploration.

Table 2: Average wall clock time to get a reasonable solution with a relative error $< 1\%$ using $q = 1,4,8$ Processors.

q	PGLO	MultPPS-LHS	MultPPS-qEI
1	210.3611	272.1705	252.2541
4	42.4318	92.1054	53.1587
8	37.0719	47.9258	39.6525

Note: For each optimization algorithm, the average wall clock time required for a reasonable solution obtained from a sample of 30 macro-replications.

In this example, MultPPS-qEI performed significantly worse with one and four processors at $\alpha = 0.05$, but there is no significant difference between PGLO and MultPPS-qEI with 8 processors. This is because

PGLO better explores the entire space in the first several iterations with a global search stage. Instead of identifying one particular local optimal area, the global search identifies promising local regions (which can include multiple local optimal areas). Once a point is picked up in one local region, it reduces the tendency to pick too many points in the same local region. Therefore, it balances between exploiting too much in the local region and spreading out more points for exploration in more local regions. When there are eight processors, MultPPS-qEI also has sufficient budget to explore each local region, hence the different becomes not significant.

Table 3: Relative speedup of parallel optimization algorithm when using $q = 4, 8$.

q	PGLO	MultPPS-LHS	MultPPS-qEI
4	4.9576	2.9565	4.7547
8	5.6790	5.7872	6.4605

Table 3 presents the relative *speedup* when $q = 4$ and $q = 8$ to evaluate the efficiency of parallelization. The relative speedup measures how well a parallel algorithm scales relatively to its serial version with additional processors. It is worth to mention that as different algorithms can require different wall clock time with one processor, a larger speedup does not mean a better algorithm in absolute time. It is also problem dependent. The results show that in this example, both PGLO and MultPPS-qEI have significant speedup for four processors, but only achieve marginal improvement with additional four processors (eight in total). This is because the additional processors will end with exploiting the same optimal area. Even though initialized from different locations, different pattern search will deliver the same optimal solution. MultPPS-LHS, on the other hand, achieves significant speedup with any additional processors because they will help to better explore the entire space.

5 CONCLUSION

In this paper, we proposed a parallel global and local optimization algorithm to enhance pattern search for global optimization. By incorporating the global and local information in AGLGP model, the enhanced pattern search algorithm better explores the entire space for global optimal solution and is suitable for parallel implementation. We also studied its numerical performance. In the future, we will evaluate its performance for practical real-time control systems and theoretically derive its asymptotic behavior.

REFERENCES

Booker, A. J., J. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. 1999. “A Rigorous Framework for Optimization of Expensive Functions by Surrogates”. *Structural optimization* 17 (1): 1–13.

Chen, C.-H., and L. H. Lee. 2010. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. World scientific.

Chevalier, C., and D. Ginsbourger. 2013. “Fast Computation of the Multi-points Expected Improvement with Applications in Batch Selection”. In *International Conference on Learning and Intelligent Optimization*, 59–69. Springer.

Clark, S. C., and P. I. Frazier. 2012. “Parallel Global Optimization Using An Improved Multi-points Expected Improvement Criterion”. In *INFORMS Optimization Society Conference, Miami.*

Ginsbourger, D., R. Le Riche, and L. Carraro. 2010. “Kriging is Well-suited to Parallelize Optimization”. In *Computational Intelligence in Expensive Optimization Problems*, 131–162. Springer.

Ginsbourger, D., R. L. Riche, L. Carraro, and D. Mi. 2009. “A Multi-points Criterion for Deterministic Parallel Global Optimization Based on Gaussian Processes”. *Journal of Global Optimization, in revision*.

Gramacy, R. B., and S. L. Digabel. 2015. “The Mesh Adaptive Direct Search Algorithm with Treed Gaussian Process Surrogates”. *Pacific Journal of Optimization* 11 (03): 419–447.

Jones, D., M. Schonlau, and W. Welch. 1998. “Efficient Global Optimization of Expensive Black-box Functions”. *Journal of Global Optimization* 13 (4): 455–492.

Meng, Q., and S. H. Ng. 2015. “An Additive Global and Local Gaussian Process Model for Large Data Sets”. In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 505–516. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Meng, Q., and S. H. Ng. 2016. “Combined Global and Local Method for Stochastic Simulation Optimization with an AGLGP Model”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 827–838. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Montgomery, D. C. 1991. *Design and Analysis of Experiments*. John Wiley & Sons.

Pedrielli, G., Y. Xing, J. H. Peh, , and S. H. Ng. “A Real Time Simulation Optimization Framework for Vessel Collision Avoidance and the Case of Singapore Strait”. *submitted to IEEE Intelligent Transportation Systems Trasactions*.

Picheny, V., T. Wagner, and D. Ginsbourger. 2013. “A Benchmark of Kriging-based Infill Criteria for Noisy Optimization”. *Structural and Multidisciplinary Optimization* 48 (3): 607–626.

Quan, N., J. Yin, S. H. Ng, and L. H. Lee. 2013. “Simulation Optimization via Kriging: a Sequential Search Using Expected Improvement with Computing Budget Constraints”. *IIE Transactions* 45 (7): 763–780.

Regis, R. G., and C. a. Shoemaker. 2007, nov. “A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions”. *INFORMS Journal on Computing* 19 (4): 497–509.

Schonlau, M., W. J. Welch, and D. R. Jones. 1998. “Global Versus Local Search in Constrained Optimization of Computer Models”. *Lecture Notes-Monograph Series*:11–25.

Sóbester, A., S. J. Leary, and A. J. Keane. 2004. “A Parallel Updating Scheme for Approximating and Optimizing High Fidelity Computer Simulations”. *Structural and Multidisciplinary Optimization* 27 (5): 371–383.

Sun, L., L. J. Hong, and Z. Hu. 2014. “Balancing Exploitation and Exploration in Discrete Optimization via Simulation Through a Gaussian Process-Based Search”. *Operations Research* 62 (6): 1416–1438.

Taddy, M. a., H. K. H. Lee, G. a. Gray, and J. D. Griffin. 2009. “Bayesian Guided Pattern Search for Robust Local Optimization”. *Technometrics* 51 (4): 389–401.

Torczon, V. 1997. “On the Convergence of Pattern Search Algorithms”. *SIAM Journal on optimization* 7 (1): 1–25.

Wang, G. G., Z. Dong, and P. Aitchison. 2001. “Adaptive Response Surface Method-a Global Optimization Scheme for Approximation-based Design Problems”. *Engineering Optimization* 33 (6): 707–733.

AUTHOR BIOGRAPHIES

QUN MENG received the B.E. degree in industrial engineering and logistics management from Shanghai Jiao Tong University, Shanghai, China, in 2012. She holds Ph.D. degree in Industrial Systems Engineering and Management from National University of Singapore. Her research interests are in the area of simulation optimization. Her email address is mengqun@u.nus.edu.

SZU HUI NG is an Associate Professor in the Department of Industrial Systems Engineering and Management at the National University of Singapore. She holds B.S., M.S., and Ph.D. degrees in Industrial and Operations Engineering from the University of Michigan. Her research interests include computer simulation modeling and analysis, design of experiments, and quality and reliability engineering. She is a member of IEEE and INFORMS and a senior member of IIE. Her email address is isensh@nus.edu.sg and her web page is http://www.isem.nus.edu.sg/staff/ngsh/index.html.