

# Development of an Adaptive Infill Criterion for Constrained Multi-Objective Asynchronous Surrogate-Based Optimization

Jolan Wauters · Andy Keane · Joris Degroote

Received: date / Accepted: date

**Abstract** The use of surrogate modeling techniques to efficiently solve a single objective optimization (SOO) problem has proven its worth in the optimization community. However, industrial problems are often characterized by multiple conflicting and constrained objectives. Recently, a number of infill criteria have been formulated to solve multi-objective optimization (MOO) problems using surrogates and to determine the Pareto front. Nonetheless, to accurately resolve the front, a multitude of optimal points must be determined, making MOO problems by nature far more expensive than their SOO counterparts. As of yet, even though access to of high performance computing (HPC) is widely available, little importance has been attributed to batch optimization and asynchronous infill methodologies, which can further decrease the wall-clock time required to determine the Pareto front with a given resolution. In this paper a novel infill criterion is developed for Generalized Asynchronous Multi-objective constrained Optimization (GAMO), which allows multiple points to be selected for evaluation in an asynchronous manner while the balance between design space exploration and objective exploitation is adapted during the optimization process in a simulated annealing like manner and the constraints are taken into account. The method relies on a formulation of the expected improvement for multi-objective optimization, where the improvement is formulated as the Euclidean distance from the Pareto front taken to a higher power. The infill criterion is tested on a series of test cases and proves the effectiveness of the novel scheme.

---

Conducted as part of the SBO research project 140068 EUFORIA (Efficient Uncertainty quantification For Optimization in Robust design of Industrial Applications) under the financial support of the IWT, the Flemish agency of Innovation through Science and Technology.

---

Jolan Wauters, Joris Degroote  
Department of Flow, Heat and Combustion Mechanics, Ghent University  
Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium  
Flanders Make, Belgium  
E-mail: {jolan.wauters, joris.degroote}@ugent.be

Andy Keane  
Computational Engineering and Design Group, University of Southampton  
Boldrewood Innovation Campus, Burgess Rd, Southampton SO16 7QF, United Kingdom  
E-mail: andy.keane@soton.ac.uk

**Keywords** Surrogate Modeling · Kriging · Expected Improvement · Batch Optimization · Multi-Objective Optimization · Constrained Optimization

## 1 Introduction

With the advent of *high performance computing* (HPC) it is nowadays possible to model complex systems, such as *unmanned aerial vehicles* (UAV), turbine cascades and wind farms, using computationally costly simulation codes, such as *computational fluid dynamics* (CFD). These complex systems are often characterized by the absence of a closed form: the problem is given as a black-box which provides an output for a given input. Even though modeling has become feasible, performing the entire design process or even carrying out an optimization on such a system might be an unfeasible problem without the use of a proper framework, even with access to HPC. An established method to handle this kind of problems is the introduction of an intermediate level in the form of a surrogate model, which is constructed using a few well selected evaluations and on which the optimization is performed. A well-known surrogate modeling technique is Kriging [11], building forth on the concepts of Gaussian processes and the theorem of Bayes. The stochastic nature of the aforementioned allows the definition of the expected improvement, which can serve as criterion for the selection of the next point to evaluate by balancing design space exploration and objective exploitation [11]. Control over the exploration-to-exploitation ratio during the optimization process can be obtained through the Generalized Expected Improvement [36].

An example of a computationally challenging problem is the optimization of UAVs (or drones). The widespread use of UAVs has become clear over recent years. Within the extensive range of UAVs that exists nowadays, of particular interest are those that operate at a chord-based Reynolds numbers ( $Re_c$ ) below  $5 \times 10^5$ , the condition which is referred to as *low Reynolds number flow* [29]. During landing, the UAV is characterized by high angles of attack and by the appearance of a large range of frequencies in the flow. This implies that both a small time step size and a large number of time steps is required to correctly resolve the flow and obtain an averaged value of the quantities of interest. As the time steps are solved sequentially, the speed-up by parallelization cannot be increased beyond a certain limit. Optimal use of available computing power is thus obtained by calculating multiple infill points at once. In an attempt to accelerate the optimization procedure by reducing the number of iterations and making full use of parallelized computational power, a multitude of infill points is selected. This can be done in a synchronous manner [32], but with a different convergence rate of the problem in each infill point, a part of the available computing power might be idle, which makes an asynchronous manner more desirable.

Different methodologies exist that tackle the asynchronous selection of multiple design points, which in this case translates itself in finding multiple local optima of infill criteria [36]. Gradient-based optimizers in combination with a multi-start or sequential global optimization with the addition of a penalty function in the ‘global’ optimum [35] serve the purpose, but lack the elegance that Gaussian processes have to offer. To complicate matters further, design problems rarely see themselves translated as a *single objective optimization* (SOO) problem, but rather as a multitude of conflicting demands and thus a *multi-objective*

*optimization* (MOO). Since the turn of the century a number of multi-objective formulations of the expected improvement have been formulated which allows a surrogate-based global multi-objective optimization [23, 41, 18, 22, 32, 3].

In this paper we bring together three different concepts: the formulation of the expected improvement for multi-objective optimization, the generalized formulation of the expected improvement which allows control over the importance of exploration and exploitation, and the ability to calculate the expected improvement in an asynchronous manner. In §2 we present the concept of Kriging followed by an overview of existing infill criteria in §3 and development of novel updating schemes in §4. The concepts are compared and tested on a series of test cases in §5.

## 2 Surrogate Modeling: Kriging

Confronted with the staggering computational cost of a single evaluation in certain optimizations, an efficient methodology is sought to find the optimal set of design parameters  $\mathbf{x} = [x_i, i = 1, \dots, d]$ . An established methodology to answer the problem at hand is found in the field of surrogate modeling. This implies that, after defining the objective function  $\mathbf{y}(\mathbf{x})$  and the design space, a design of experiments (DoE) is set up to select  $n$  samples in the design space, for which the objective function is subsequently calculated and of which a surrogate is defined. This cheap to evaluate surrogate or meta-model can thereafter be sampled to define the entire characteristics in function of the geometric design variables and can be updated with additional data during the optimization.

The surrogate model used in this work is Kriging, which can be seen as the sum of a trend function and Gaussian process:  $\mathcal{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \mathcal{Z}(\mathbf{x})$  with  $\mathbf{f}(\mathbf{x}) = [f_i(\mathbf{x}), i = 1, \dots, m]$  the vector of basis functions,  $\boldsymbol{\beta}$  the vector of coefficients and  $\mathcal{Z}(\mathbf{x})$  a Gaussian process  $\mathcal{GP}(0, \text{cov}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}))$ , with zero mean and fully described by the covariance function  $\text{cov}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) = \sigma^2 \text{cor}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)})$ , where  $\sigma$  is the process variance and  $\text{cor}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)})$  is the correlation function between two designs and is noted for being a function of their inputs and typically written as  $\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ . Here the Matérn covariance function is used with  $\nu = 3/2$ . The trend is typically the solution of a regression problem and the Gaussian process captures the variation on this trend to exactly interpolate the evaluated data.

In order to determine the parameters of the covariance function, typically referred to as hyperparameters, we maximize the likelihood  $\mathbf{L}$  that the aforementioned surrogate can reproduce the evaluated data. Solving the *maximum likelihood estimation* (MLE) problem, we can define the *Best Linear Unbiased Prediction* (BLUP), which allows the prediction of unsampled locations  $\mathbf{x}$  with respectively the predicted mean and predicted variance:

$$\mathbf{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\Psi}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \quad (1)$$

$$s^2(\mathbf{x}) = \sigma^2 \left( 1 - \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi}(\mathbf{x}) + \left( \mathbf{F}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi}(\mathbf{x}) - \mathbf{f}(\mathbf{x}) \right)^T \left( \mathbf{F}^T \boldsymbol{\Psi}^{-1} \mathbf{F} \right)^{-1} \left( \mathbf{F}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi}(\mathbf{x}) - \mathbf{f}(\mathbf{x}) \right) \right) \quad (2)$$

with  $\mathbf{F}$  the  $n \times m$  model matrix:  $F_{i,j} = f_i(\mathbf{x}^{(j)})$  and  $\Psi$  the  $n \times n$  correlation matrix:  $\Psi_{i,j} = \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ ,  $\boldsymbol{\psi}(\mathbf{x}) = [\psi(\mathbf{x}^{(1)}, \mathbf{x}), \dots, \psi(\mathbf{x}^{(n)}, \mathbf{x})]$  and the MLE of the coefficient vector and the process variance respectively defined by  $\boldsymbol{\beta} = (\mathbf{F}\Psi^{-1}\mathbf{F})^{-1}\mathbf{F}^T\Psi^{-1}\mathbf{y}$  and  $\sigma^2 = \frac{1}{n}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T\Psi^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})$ .

The construction of the Kriging model is performed using an open source toolbox ooDACE (*object-orientated Design and Analysis of Computer Experiments*) [4]. The maximization of the concentrated ln-likelihood function is performed through a multi-start *sequential quadratic programming* (SQP) methodology.

### 3 Infill Criteria

#### 3.1 Single Objective

The improvement of a predicted point on the current best evaluated point,  $y_{min} = \min(Y(\mathbb{X}))$  with  $\mathbb{X}$  the set of all samples, is defined as (following Janusevskis et al.'s notation [17]):  $I(\mathbf{x}) = (y_{min} - Y(\mathbf{x}) | Y(\mathbb{X}) = y(\mathbb{X}))^+$  with  $(\cdot)^+ = \max(\cdot, 0)$ . The stochastic nature of Kriging allows the assessment of the uncertainty in the prediction. This can be used to define the expected improvement  $\mathbb{E}[I(\mathbf{x})] = EI(\mathbf{x})$  [27]:

$$\begin{aligned} EI(\mathbf{x}) &= \int_{-\infty}^{\infty} I(\mathcal{Y})\phi(\mathcal{Y}, \mathbf{x})d\mathcal{Y} \\ &= \int_{-\infty}^{y_{min}} (y_{min} - \mathcal{Y}) \frac{1}{s(\mathbf{x})\sqrt{2\pi}} \exp\left[-\frac{(\mathcal{Y} - Y(\mathbf{x}))^2}{2s^2(\mathbf{x})}\right] d\mathcal{Y}. \end{aligned} \quad (3)$$

This leads to the well-known formula of the expected improvement, of which the proof is given in Appendix A,

$$EI(\mathbf{x}) = \begin{cases} (y_{min} - Y(\mathbf{x}))\Phi(u(\mathbf{x})) + s(\mathbf{x})\phi(u(\mathbf{x})) & \text{if } s > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

with  $\Phi$  and  $\phi$  respectively the standard normal cumulative and probability density function,  $\phi$  the one-dimensional Gaussian probability density function and  $u(\mathbf{x}) = (y_{min} - Y(\mathbf{x}))/s(\mathbf{x})$ . Note that the first term drives the minimization of the objective (exploitation), while the second term drives the minimization of the uncertainty of the prediction (exploration). Alternatively, the expected improvement can be formulated as

$$\begin{aligned} EI(\mathbf{x}) &= y_{min}\Phi(u(\mathbf{x})) - \int_{-\infty}^{y_{min}} \mathcal{Y}\phi(\mathcal{Y}, \mathbf{x})d\mathcal{Y} \\ &= \left( y_{min} - \frac{\int_{-\infty}^{y_{min}} \mathcal{Y}\phi(\mathcal{Y}, \mathbf{x})d\mathcal{Y}}{\Phi(u(\mathbf{x}))} \right) \Phi(u(\mathbf{x})) \\ &= (y_{min} - C_{P[(\mathbf{x})]}) P[(\mathbf{x})]. \end{aligned} \quad (5)$$

This allows a geometrical interpretation of the expected improvement: the product of the area under the probability density function cut off by the best evaluation

( $P[I(x)]$ ) and the distance between the centroid of the aforementioned area and the best evaluation  $|y_{min}, C_{P[I(x)]}|$  (figure 1) [40].

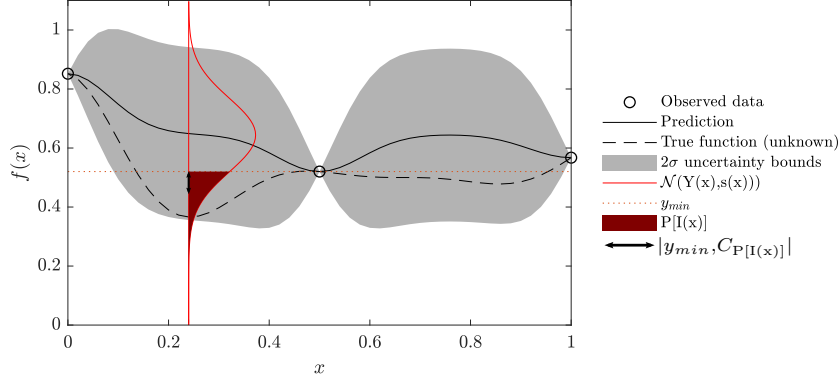


Fig. 1: Graphical interpretation of the expected improvement, corresponding to the product of the probability of improvement  $P[I(x)]$  and the distance of its centroid to the current best evaluation  $|y_{min}, C_{P[I(x)]}|$ . Reproduced from Forrester et al. [11]

---

#### Algorithm 1 Efficient Global Optimization (EGO) [20]

---

**Require:** Evaluated sampling plan  $(\mathbb{X}, y(\mathbb{X}))$

**Ensure:** The best observation  $(\mathbf{x}_{min}, y_{min})$

- 1: **while** stopping criteria are not met **do**
  - 2:     Estimate hyperparameters ▷ Section 2
  - 3:      $\mathbf{x}_{new} = \arg \max_{\mathbf{x}} EI(\mathbf{x})$  ▷ Equation 4
  - 4:      $\mathbb{X} \leftarrow \mathbb{X} \cup \mathbf{x}_{new}$
  - 5:      $y_{min} \leftarrow \min_{\mathbf{x}} (y(\mathbb{X}))$
  - 6:      $\mathbf{x}_{min} \leftarrow \arg \min_{\mathbf{x}} (y(\mathbb{X}))$
- 

This infill criterion forms the basis of the well known efficient global optimization (EGO) algorithm by Jones et al. [20] (see Algorithm 1). However, based on the observation that Kriging is prone to an underestimation of the prediction error  $s$  caused by the assumption of a fully known parameter and hyperparameter set, the expected improvement is prone to higher emphasis on the estimated value [36]. Especially at earlier stages of the optimization, this may lead to the infill criterion getting stuck in a possibly local optimum and overemphasizing the region before exploring the parameter space further. To counter this phenomenon, Schonlau et al. [36] proposed a generalized formulation of the improvement by taking it to a higher power, effectively deriving the generalized expected improvement  $\mathbb{E}[I^g(\mathbf{x})] = GEI^g(\mathbf{x})$

$$GET^g(\mathbf{x}) = s^g \sum_{k=0}^g (-1)^k \binom{g}{k} u(\mathbf{x})^{g-k} T_k \quad (6)$$

with  $T_k$  a recursive function given by  $T_k = -u^{k-1}\phi(u) + (k-1)T_{k-2}$  with  $T_0 = \Phi(u)$  and  $T_1 = -\phi(u)$  and  $u = (y_{min} - Y(\mathbf{x}))/s(\mathbf{x})$ . Sasena et al. proposed to have  $g$  decrease in a ‘Simulated Annealing’-like manner with the objective of attributing more importance to objective space exploration early in the optimization and more importance to objective exploitation later on [34]. Table 1 shows the value of  $g$  during the optimization as proposed by Sasena et al.

Table 1: Sasena et al.’s cooling scheme [34]

Iteration	1-4	5-9	10-19	20-24	25-34	$\geq 35$
$g$ -value	20	10	5	2	1	0

In case that more computing nodes are available than the number of nodes over which a single evaluation of the objective function can be spread, it becomes worthwhile to determine multiple infill points at once in a synchronous or asynchronous manner.

Examples of existing synchronous methods are amongst others the *Clustered Multiple Generalized Expected Improvement* (CMGEI), which uses the generalized formulation of the expected improvement and was proposed by Ponweiser et al. [32]. In their work they cluster based on the Mahalanobis distance to select a multitude of points to be evaluated simultaneously. A similar approach is brought forward by Sóbester et al. [35], which relies on a multitude of local optima of the EI criterion. Ginsbourger et al. [14] came up with the q-steps EI, in which the design space is enhanced q-fold such that the q-steps EI optimum corresponds to q optimal points to be filled in simultaneously. An analytical expression was formulated by Chevalier [1] relying on Tallis’ formula. An expression of the gradient was subsequently derived by Marmin et al. [25].

In case of an asynchronous setting, a straightforward approach is the use of penalty functions. Schonlau et al. [36] proposed a *sequential design in stages* where the global optimum of the expected improvement is sought and subsequently the model’s prediction is added to the training set. After retraining the model, the variance reduces locally to zero and repeating the procedure ensures that the next infill point moves away from previously added samples. While use is made of the stochastic nature of the surrogate model, this method relies on the assumption that the predictor is accurate enough. Ginsbourger et al. referred to this method as *Kriging Believer* (KB) and alternatively proposed a *Constant Liar* (CL) approach [14]. Furthermore, the addition of an infill point and the accompanied retraining of the model can be computationally costly. More accurate is the calculation of the expectation of the EI (EEI [13]). However, since there is no tractable analytical expression of the aforementioned, this is typically calculated using *Monte Carlo Integration* (MCI), maintaining its expensive nature. All of the aforementioned methods have a sequential approach and are thus in essence asynchronous updating methods. This may become prohibitive, up to the point that not all the computing

nodes get filled up because the calculation of a new infill point might exceed its evaluation.

Janusevskis et al. [16,17] defined the asynchronous multi-points expected improvement  $EI^{(\mu,\lambda)}$  when  $\mu$  computing nodes are evaluating the objective and  $\lambda$  computing nodes are available to be filled, allowing the aforementioned methods to be grouped:

$$EI^{(\mu,\lambda)}(\mathbb{X}_\lambda) = \mathbb{E}[\min(Y(\mathbb{X} \cup \mathbb{X}_\mu)) - \min(Y(\mathbb{X}_\lambda)) \mid Y(\mathbb{X}) = \mathbf{y}] \quad (7)$$

with  $\mathbb{X}_\mu = \mathbf{x}_b^{(1)}, \dots, \mathbf{x}_b^{(\mu)}$  the set of parameter combinations that is currently being evaluated and  $\mathbb{X}_\lambda = \mathbf{x}_a^{(1)}, \dots, \mathbf{x}_a^{(\lambda)}$  the set of parameter combinations that is to be determined for evaluation. If  $\mu = 0$  and  $\lambda = 1$  we obtain the well known  $EI$ . In case that  $\mu = 0$  and  $\lambda \geq 2$  we obtain the earlier described  $q$ -steps  $EI$ . Janusevskis et al.'s [16,17] proposed a mixed strategy to determine the most promising batch by combining MCI with bounds to allow ranking of candidate points, thus reducing the number of samples required. Here, we present a reformulation of the MCI of  $q$ - $EI$  (as introduced in [14]) for  $EI^{(\mu,\lambda)}$  (see algorithm 2). The method relies on the numerical integration of the multivariate distribution of the  $\mu + \lambda$  unknown responses conditional on the observations  $\mathcal{Y}(\mathbb{X}_{\mu+\lambda}) \sim \mathcal{N}(\mathbf{Y}(\mathbb{X}_{\mu+\lambda}), \boldsymbol{\Psi}_{\mu+\lambda})$  with  $\mathbb{X}_{\mu+\lambda} = [\mathbb{X}_\mu, \mathbb{X}_\lambda]^T$  and

$$\boldsymbol{\Psi}_{\mu+\lambda}^{(i,j)} = \psi(\mathbb{X}_{\mu+\lambda}^{(i)}, \mathbb{X}_{\mu+\lambda}^{(j)} \mid Y(\mathbb{X}) = \mathbf{y}) \quad \text{where } i, j = 1, \dots, \mu + \lambda. \quad (8)$$

Consequently, the law of large numbers (LLN) gives us

$$I_{sim}^{(\mu,\lambda)}(i) = [\min(\mathbf{y}, \mathbf{M}_\mu(i)) - \min(\mathbf{M}_\lambda(i))]^+ \quad (9)$$

$$EI^{(\mu,\lambda)}(\mathbb{X}_\lambda) = \lim_{n_{sim} \rightarrow \infty} \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} I_{sim}^{(\mu,\lambda)}(i) \quad (10)$$

with  $n_{sim}$  the number of random sampling sets,  $\mathbf{M}_{\mu+\lambda}(i) = [\mathbf{M}_\mu(i), \mathbf{M}_\lambda(i)]^T$  a  $(\mu + \lambda) \times 1$  vector and a random sampling of  $\mathcal{Y}(\mathbb{X}_{\mu+\lambda})$  obtained through Kaiser & Dickman's method, which is a straightforward procedure relying on the concepts of *Principal Component Analysis* (PCA) [21]. PCA states that a sample matrix  $\mathbf{Z}$  from a multivariate normal distribution specified by the correlation matrix  $\boldsymbol{\Psi}_{\mu+\lambda}$  can be obtained from a linear transformation of a set of independent, identically distributed univariates  $\mathbf{N} \sim \mathcal{N}(\mathbf{0}_{\mu+\lambda}, \mathbf{I}_{\mu+\lambda})$  through  $\mathbf{Z} = \mathbf{L} \cdot \mathbf{N}$  with  $\mathbf{L}$  the Cholesky decomposition of the covariance matrix  $\mathbf{L} = \text{chol}(\boldsymbol{\Psi}_{\mu+\lambda})$ , such that  $\boldsymbol{\Psi}_{\mu+\lambda} = \mathbf{L} \cdot \mathbf{L}^T$ . This leads to the following results for the sampling:  $\mathbf{M}_{\mu+\lambda} = \mathbf{Y}(\mathbb{X}_{\mu+\lambda}) + \mathbf{L} \cdot \mathbf{N}$ . A more extensive overview of the sampling of a multivariate normal distribution using a decomposition of its covariance matrix and alternative methods is given in [12].

### 3.2 Multi-Objective

When dealing with a design problem, engineers are often confronted with conflicting objectives. This issue can be resolved by creating a single objective function which is made up out of a weighted sum of the different objectives. However, this

**Algorithm 2** Asynchronous Multi-Points Expected Improvement ( $EI_{MC}^{(\mu,\lambda)}$ )

---

```

1: Construct  $\Psi_{\mu+\lambda}$  ▷ Equation 8
2:  $\mathbf{L} = \text{chol}(\Psi_{\mu+\lambda})$  ▷ Cholesky decomposition of covariance matrix
3: for  $i \leftarrow 1, n_{sim}$  do ▷ Loop over simulation size
4:    $\mathbf{N}(i) \sim \mathcal{N}(\mathbf{0}_{\mu+\lambda}, \mathbf{I}_{\mu+\lambda})$  ▷ Sample from i.i.d. univarites
5:    $\mathbf{M}_{\mu+\lambda}(i) = Y(\mathbb{X}_{\mu+\lambda}) + \mathbf{L} \cdot \mathbf{N}(i)$  ▷ Kaiser & Dickman's method
6:    $[\mathbf{M}_{\mu}(i), \mathbf{M}_{\lambda}(i)]^T = \mathbf{M}_{\mu+\lambda}(i)$ 
7:    $I_{sim}^{(\mu,\lambda)}(i) = [\min(\mathbf{y}, \mathbf{M}_{\mu}(i)) - \min(\mathbf{M}_{\lambda}(i))]^+$  ▷ Equation 9
8:  $EI_{MC}^{(\mu,\lambda)} = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} I_{sim}^{(\mu,\lambda)}(i)$  ▷ Equation 10

```

---

only moves the problem further downstream: what weight is attributed to which objective? An alternative is to determine the (hyper-)surface of non-dominated objectives for which we cannot improve on one without deteriorating on the other(s). This surface is typically referred to as the Pareto front [9]. During the 90s a number of multi-objective evolutionary algorithms (MOEAs) were proposed to solve this issue, amongst others Fonseca and Fleming's MOGA [8], Srinivas and Deb's NSGA [37], Horn et al.'s NPGA [15], Zitzler and Thiele's SPEA [44], Knowles and Corne's PAES [24] and at its pinnacle Deb et al.'s NSGA-II [5], to name but a few. While many of these algorithms have proven their abilities, their stochastic nature leads to a very high computational cost if the objectives are expensive to evaluate.

A solution of the aforementioned is the introduction of an intermediate surrogate to overcome the prohibitive computational cost of MOEAs [39]. Inspired by the EGO algorithm, the turn of the century has seen the birth of a series of multi-objective reformulations of the expected improvement. One of the earlier approaches, ParEGO [23], reformulates the multi-objective problem as a single objective problem using an augmented Tchebycheff function. A generalization of this approach, MOEA/D (Multi Objective Evolutionary Algorithm based on Decomposition) was presented by Zhang et al. [41] and allows a level of parallelization through the formulation of a series of single-objective optimization subproblems. The generation of the Pareto front is in these approaches subjected to the weighting function. Especially in case of nonconvex Pareto fronts this may lead to clustering of evaluations in the extremes of the front and a poor resolution and spread of the region in between. Jeong & Obayashi's Multi-EGO [18] uses a multi-objective genetic algorithm to determine the Pareto front of the EIs of both objectives and selects three points on the front (both extremes and a point in the middle) as infills for the next iteration, thus enforcing a level on parallelization in the algorithm. Keane [22] presented a formulation of the expected improvement for two objectives based on the geometrical interpretation of the  $EI$  (Wagner et al. refer to this as the Euclid-EI [40]). Emmerich et al.'s SExI ( $\mathcal{S}$ -metric Expected Improvement) algorithm [7], Ponweiser et al.'s SMS ( $\mathcal{S}$ -Metric Selecting)-EGO algorithm [32] and Couckuyt et al.'s EMO (Efficient Multi-objective Optimization) algorithm [3] use the hypervolume (HV) or  $\mathcal{S}$ -metric, which corresponds to the Lebesgue measure of the hyperspace dominated by the Pareto front. A strong theoretical assessment of the aforementioned can be found in Wagner et al. [40].

In this paper we start from Keane's formulation, which we briefly summarize here. For a SOO problem, the  $EI$  can be understood as the product of the area under the tail of the prediction cut off by the best evaluation and the distance



between the centroid of the aforementioned area and the best evaluation [40]. For two objectives Keane reformulates this as the product of the probability that a prediction is found outside the Pareto front,  $P[\mathbf{I}(\mathbf{x})]$ , with the Euclidean distance of the nearest point on the Pareto front,  $\mathbf{y}_{1,2}^* = \{[y_1^{*(1)}, y_2^{*(1)}], \dots, [y_1^{*(m)}, y_2^{*(m)}]\}$ , with  $m$  the number of points on the Pareto front, to the centroid of the area outside the Pareto front,  $(\bar{\mathcal{Y}}_1(\mathbf{x}), \bar{\mathcal{Y}}_2(\mathbf{x}))$ :

$$\mathbb{E}[\mathbf{I}(\mathbf{x})] = P[\mathbf{I}(\mathbf{x})] \sqrt{(\bar{\mathcal{Y}}_1(\mathbf{x}) - y_1^*)^2 + (\bar{\mathcal{Y}}_2(\mathbf{x}) - y_2^*)^2} \quad (11)$$

In this work, building forth on the assumption of uncorrelated objectives, a two-dimensional Gaussian probability density function  $\phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x})$  for the predicted responses is constructed

$$\phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) = \frac{1}{s_1(\mathbf{x})\sqrt{2\pi}} \exp\left[-\frac{(\mathcal{Y}_1 - Y_1(\mathbf{x}))^2}{2s_1^2(\mathbf{x})}\right] \cdot \frac{1}{s_2(\mathbf{x})\sqrt{2\pi}} \exp\left[-\frac{(\mathcal{Y}_2 - Y_2(\mathbf{x}))^2}{2s_2^2(\mathbf{x})}\right] \quad (12)$$

By integrating this two dimensional Gaussian probability density function over the region outside the Pareto front<sup>1</sup> we obtain the multi-objective probability of improvement  $P[\mathbf{I}(\mathbf{x})] = P[\mathcal{Y}_1(\mathbf{x}) < y_1^* \cap \mathcal{Y}_2(\mathbf{x}) < y_2^*]$ .

$$\begin{aligned} P[\mathbf{I}(\mathbf{x})] &= \int_{-\infty}^{y_1^{*(1)}} \int_{-\infty}^{\infty} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \\ &+ \sum_{i=1}^{m-1} \int_{y_1^{*(i)}}^{y_1^{*(i+1)}} \int_{-\infty}^{y_2^{*(i+1)}} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \\ &+ \int_{y_1^{*(m)}}^{\infty} \int_{-\infty}^{y_2^{*(m)}} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \end{aligned} \quad (13)$$

The first, second and third term in equation 13 correspond to respectively the series of blocks I, II and III as presented in figure 2. The calculation of the centroid  $\bar{\mathcal{Y}}_1(\mathbf{x})$  is performed by determining the first moment of area and dividing it by the multi-objective probability of improvement:

$$\begin{aligned} \bar{\mathcal{Y}}_1(\mathbf{x}) &= \left[ \int_{-\infty}^{y_1^{*(1)}} \int_{-\infty}^{\infty} \mathcal{Y}_1 \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \right. \\ &+ \sum_{i=1}^{m-1} \int_{y_1^{*(i)}}^{y_1^{*(i+1)}} \int_{-\infty}^{y_2^{*(i+1)}} \mathcal{Y}_1 \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \\ &\left. + \int_{y_1^{*(m)}}^{\infty} \int_{-\infty}^{y_2^{*(m)}} \mathcal{Y}_1 \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \right] \cdot \frac{1}{P[\mathbf{I}(\mathbf{x})]} \end{aligned} \quad (14)$$

<sup>1</sup> Depending on whether it is the intent to improve upon the existing Pareto front (figure 2, hatched area) or refine the existing front (figure 2, gray area), the limits of the integrals are slightly different. Here, improvement upon the Pareto is presented. For further details on this matter, see [22].

$\bar{Y}_2(\mathbf{x})$  is calculated in an analogous way.

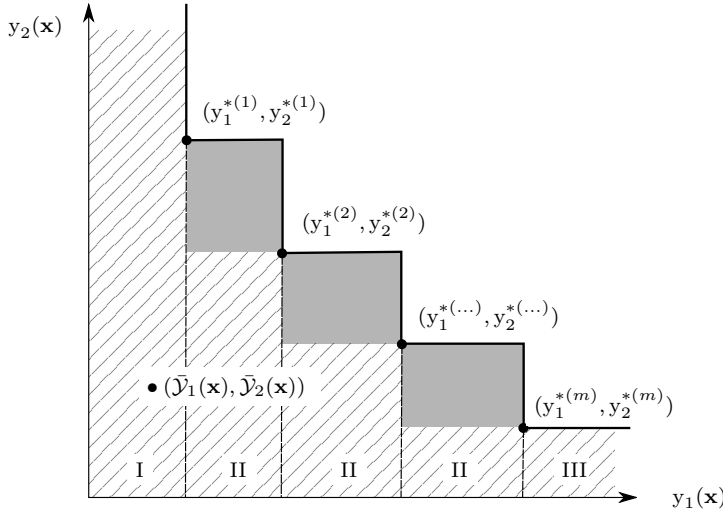


Fig. 2: Schematic representation of the partitioning of the area outside of the Pareto front of two contesting objectives. The hatched area corresponds to the area where the Pareto front can be improved. The grey area corresponds to the area where the Pareto front can be refined. Reproduced from Keane [22].

#### 4 New Infill Criterion

A limitation of Keane's EI is the absence of a clear formulation of the improvement [40]. This is related to the geometric origin of the way by which it is built up and the intrinsic difficulty of the multidimensional untractable integral that comes with it. Therefore we formulate the improvement as the distance from the Pareto front taken to a power  $g$ :  $I^g(\mathbf{x}) = [|\text{Pareto}(\mathbf{Y}(\mathbb{X})), \mathbf{Y}(\mathbf{x})|_{min}^g]^+$ . Here  $|\text{Pareto}(\mathbf{Y}(\mathbb{X})), \mathbf{Y}(\mathbf{x})|$  is defined as the distance measured from the Pareto front to  $\mathbf{Y}(\mathbf{x})$  and  $[\cdot]_{min}^+$  is defined as the smallest value evaluated outside of the Pareto front. To facilitate the expansion, the equations will be limited to two objectives in the following step. For two objectives<sup>2</sup>, this corresponds with  $I^g(\mathbf{x}) = [(\sqrt{(y_1^* - Y_1(\mathbf{x}))^2 + (y_2^* - Y_2(\mathbf{x}))^2})^g)_{min}^+]^+$ . If  $g$  is an even number (with  $h = g/2$ ), the formulation simplifies due to the disappearance of the square root. This leads to a tractable definition of the expected improvement  $\mathbb{E}[I^g(\mathbf{x})]$ , to which we refer as the generalized Euclidean/Multi-objective Expected Improvement (GMOEI). By making use of the binomial theorem<sup>3</sup>, we obtain

<sup>2</sup> Theoretically the method can be rewritten for any number of objectives. However this method, as any other, is subjected to the curse of dimensionality (no-free-lunch theorem). In practice, defining the computational domain outside the Pareto front for more than two objectives can become cumbersome [4] and is for most methods the limiting factor in their applicability.

<sup>3</sup> For a higher number of objectives, the multinomial theorem leads to the desired solution.

$$\mathbb{E}[I^g(\mathbf{x})] = \mathbb{E} \left[ \sum_{k=0}^h \binom{h}{k} (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2k} (\mathbf{y}_2^* - Y_2(\mathbf{x}))^{2(h-k)} \right]. \quad (15)$$

This can be rewritten under the assumption that the objectives are uncorrelated.

$$\begin{aligned} \mathbb{E}[I^g(\mathbf{x})] &= \mathbb{E} \left[ (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2n} \right] + \sum_{k=1}^{h-1} \binom{h}{k} \mathbb{E} \left[ (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2k} \right] \\ &\quad \cdot \mathbb{E} \left[ (\mathbf{y}_2^* - Y_2(\mathbf{x}))^{2(h-k)} \right] + \mathbb{E} \left[ (\mathbf{y}_2^* - Y_2(\mathbf{x}))^{2n} \right] \end{aligned} \quad (16)$$

By partitioning the integrand space in boxes as was done by Keane and is presented in figure 2,  $\mathbb{E} \left[ (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2k} \right]$  can be rewritten as

$$\begin{aligned} \mathbb{E} \left[ (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2k} \right] &= \int_{-\infty}^{y_1^{*(1)}} \int_{-\infty}^{\infty} (y_1^{*(1)} - \mathcal{Y}_1(\mathbf{x}))^{2k} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \\ &\quad + \sum_{i=1}^{m-1} \int_{y_1^{*(i)}}^{y_1^{*(i+1)}} \int_{-\infty}^{y_2^{*(i+1)}} (y_1^{*(i)} - \mathcal{Y}_1(\mathbf{x}))^{2k} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1 \\ &\quad + \int_{y_1^{*(m)}}^{\infty} \int_{-\infty}^{y_2^{*(m)}} (y_1^{*(m)} - \mathcal{Y}_1(\mathbf{x}))^{2k} \phi(\mathcal{Y}_1, \mathcal{Y}_2, \mathbf{x}) d\mathcal{Y}_2 d\mathcal{Y}_1. \end{aligned} \quad (17)$$

Analogously for  $\mathbb{E} \left[ (\mathbf{y}_2^* - Y_2(\mathbf{x}))^{2k} \right]$ . The result of equation 17 is given by

$$\begin{aligned} \mathbb{E} \left[ (\mathbf{y}_1^* - Y_1(\mathbf{x}))^{2k} \right] &= \mathbb{E} \left[ (\mathbf{y}_1^{*(1)} - Y_1(\mathbf{x}))^{2k} \right] + \sum_{i=1}^{m-1} \left( \mathbb{E} \left[ (\mathbf{y}_1^{*(i+1)} - Y_1(\mathbf{x}))^{2k} \right] \right. \\ &\quad \left. - \mathbb{E} \left[ (\mathbf{y}_1^{*(i)} - Y_1(\mathbf{x}))^{2k} \right] \right) \Phi \left( \frac{y_2^{*(i+1)} - Y_2(\mathbf{x})}{s_2(\mathbf{x})} \right) \\ &\quad + \mathbb{E} \left[ (\mathbf{y}_1^{*(m)} - Y_1(\mathbf{x}))^{2k} \right] \Phi \left( \frac{y_2^{*(m)} - Y_2(\mathbf{x})}{s_2(\mathbf{x})} \right) \end{aligned} \quad (18)$$

where  $\mathbb{E}[(\mathbf{y}_1^{*(i)} - Y_1(\mathbf{x}))^{2k}]$  can be calculated using equation 6. With a higher value of  $g$ , a larger size of the current Pareto front and a larger number of infills, the recursive formulation can become cumbersome, up to the point that a simple Monte Carlo approach can become more feasible.

In a similar manner to the single objective case, the asynchronous multi-points GMOEI can be formulated.

$$GMOEI^{(g, \mu, \lambda)}(\mathbb{X}_\lambda) = \mathbb{E}[\text{Pareto}(\mathbf{Y}(\mathbb{X} \cup \mathbb{X}_\mu), \mathbf{Y}(\mathbb{X}_\lambda) \mid \mathbf{Y}(\mathbb{X}) = \mathbf{y}^+_{min})] \quad (19)$$

However, as its SOO variant, the analytical expression becomes challenging, leading us to again resort to numerical multivariate integral approximation techniques in the form of Monte Carlo Integration (MCI).

$$I_{sim}^{(g,\mu,\lambda)}(i) = |\text{Pareto}(\mathbf{y}, \mathbf{M}_{obj,\mu}(i), \mathbf{M}_{obj,\lambda}(i))|_{min}^g \quad (20)$$

$$GMOEI_{MC}^{(g,\mu,\lambda)}(\mathbb{X}_\lambda) = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} I_{sim}^{(g,\mu,\lambda)}(i) \quad (21)$$

The multivariate sampling of each objective  $\{\mathbf{M}_{\lambda+\mu}^{(j)}(i)\}_{j=1\dots n_{obj}}$  (with  $n_{obj}$  the number of objectives) is performed as in algorithm 2 using Kaiser & Dickman's method.

If  $(g, \mu, \lambda)$  corresponds to  $(1, 0, 1)$  we obtain a variant<sup>4</sup> of Keane's EI. When  $(g, \mu, \lambda)$  corresponds to  $(1, 0, 2)$  we obtain a multi-objective formulation of the  $q$ -steps EI. In the case of a single variable problem, the latter can be visualized as in figure 3. It can be seen that the figure is symmetric around its diagonal, which corresponds to  $(1, 0, 1)$ <sup>5</sup> with local minima at 0, 0.25, 0.5, 0.75 and 1, where the function has been evaluated and local maxima in between these points. Combination of these local optima on the diagonal create the global hot spots. While the local maxima on the diagonal are near equal in value, the hot spots become brighter when moving towards the top left (or bottom right), which indicates that distance between the sampled points is taken into account. Methods relying on local optima for batch optimization will fail when only one optimum exists. However, this method will select a multitude of points around the optimum at a distance from each other [14].

---

<sup>4</sup> Keane's formulation is characterized by the absence of a definition of the improvement, but the presence of an analytical form of the expected improvement, which relies on the calculation of the centroid of the prediction outside the Pareto front and determining the distance to the nearest point on the front. Here the improvement is analytically defined as the Euclidean distance from the front. However, this leads to the absence of an analytical formulation of the expected improvement, unless the improvement is taken to an even power, as seen above. The calculation of the improvement relies on MCI where the distance of each sample is taken to the nearest point on the front.

<sup>5</sup> The addition of a point leads to a reduction of the improvement that can be made at the point to zero, thus  $GMOEI^{(1,0,2)}(\mathbf{x}_1, \mathbf{x}_2) = GMOEI^{(1,0,1)}(\mathbf{x}_1) = GMOEI^{(1,0,1)}(\mathbf{x}_2)$  if  $\mathbf{x}_1 = \mathbf{x}_2$ .

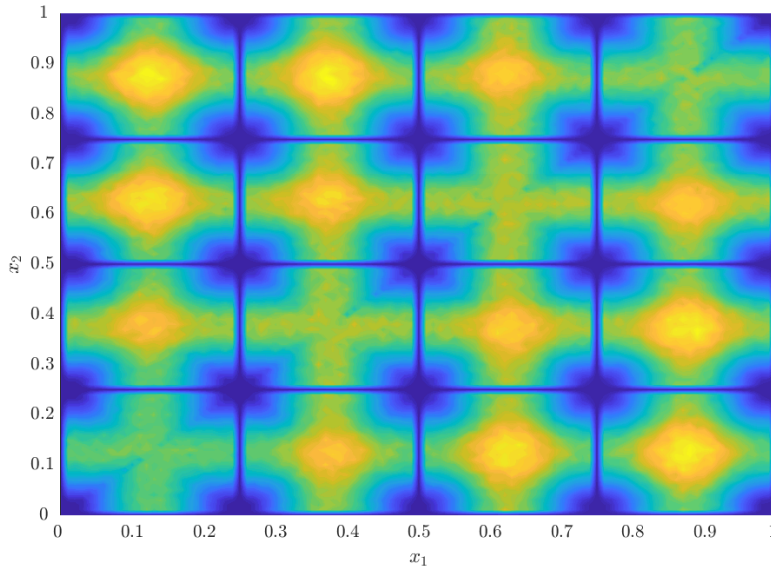


Fig. 3: Contour plot colored by  $GMOEI^{(1,0,2)}$  as a function of two infill points  $(x_1, x_2)$  for a simple one variable problem ( $\min y_1(x) = (10x - 10)^2$  and  $y_2(x) = (10x - 12)^2$ ) after five equally spaced fill-ins, namely  $x = 0, 0.25, 0.50, 0.75$  and  $1$ .

When dealing with a constrained optimization problem,  $EI$  should decrease to zero when the constraint is violated. This can be obtained through an extended expected improvement [36] or a penalty function, such as the Augmented Lagrangian function [2]. Here, the former is used: given the surrogate of the constraint  $G$  with  $G(\mathbb{X}) = \mathbf{g}$ , we can calculate the probability of the prediction not violating the constraint limit  $g_{min}$ , i.e. the probability that the constraint is met,  $P[F(\mathbf{x})]$ , where  $F$  is the measure of feasibility  $G(\mathbf{x}) - g_{min}$ . Under the assumption of uncorrelated objectives and constraints, it is easy to reformulate the expected improvement such that it accounts for the probability of feasibility  $\mathbb{E}[I(\mathbf{x}) \cap F(\mathbf{x})] = \mathbb{E}[I(\mathbf{x})]P[F(\mathbf{x})]$ . This implies that at a given point in the design space, while the predicted constraints might be violated, the predicted errors in the constraint models are different from zero and as such the expectation of feasible improvement will be low, but not zero, since there is a finite possibility that a full evaluation of the constraints may actually reveal a feasible design. This allows design space exploration in the early stages of the optimization methodology, but ensures convergence to the exact constrained optimum [11]. Nevertheless, Parr et al. remark that if the constraints are approximated inaccurately early on, the algorithm may discard optimal solutions [31].

The inclusion of the probability of feasibility of the constraint(s) in a sampling-based evaluation of the expected improvement corresponds to the multiplication of the aforementioned with the numerical integration (through sampling) of the heaviside function evaluation  $\mathcal{H}$  of these samples drawn from the multivariate dis-

tribution of the constraints  $G - g_{min}$ . It is further on assumed that  $G$  is formulated such that  $g_{min} = 0$ .

$$F_{sim}^{(k)}(i) = \prod_{l=1}^{\lambda} \mathcal{H}(\{M_{con,\lambda}^{(k)}(i)\}_l) \quad (22)$$

$$GMOEI_{c,MC}^{(g,\mu,\lambda)} = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} I_{sim}^{(g,\mu,\lambda)}(i) \prod_{k=1}^{n_{con}} \left( \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} F_{sim}^{(k)}(i) \right) \quad (23)$$

The multivariate sampling of each objective  $\{M_{obj,\lambda+\mu}^{(j)}\}_{j=1\dots n_{obj}}$  and each constraint  $\{M_{con,\lambda+\mu}^{(k)}(i)\}_{k=1\dots n_{con}}$  (with  $n_{con}$  the number of constraints) is again performed as in algorithm 2 using Kaiser & Dickman's method. The feasibility of each constraint corresponds to the product of the Heaviside evaluations of each element of the sampling of the multivariate distribution of the available computing nodes to be filled.

---

**Algorithm 3** Generalized Asynchronous Constrained Multi-Objective Expected Improvement  $GMOEI_{c,MC}^{(g,\mu,\lambda)}$

---

```

1: for  $j \leftarrow 1, n_{obj}$  do ▷ Loop over objectives
2:    $L_{obj}^{(j)} = \text{chol}(\Psi_j[\mathbb{X}_{\mu+\lambda} \mid Y_j(\mathbb{X}) \equiv \mathbf{y}_j])$ 
3: for  $k \leftarrow 1, n_{con}$  do ▷ Loop over constraints
4:    $L_{con}^{(k)} = \text{chol}(\Psi_k[\mathbb{X}_{\mu+\lambda} \mid G_k(\mathbb{X}) \equiv \mathbf{g}_k])$ 
5: for  $i \leftarrow 1, n_{sim}$  do ▷ Loop over simulation size
6:    $\mathbf{N}(i) \sim \mathcal{N}(\mathbf{0}_{\mu+\lambda}, \mathbf{I}_{\mu+\lambda})$  ▷ Sample from i.i.d. univariates
7:   for  $j \leftarrow 1, n_{obj}$  do
8:      $M_{obj,\mu+\lambda}^{(j)}(i) = Y_j(\mathbb{X}_{\mu+\lambda}) + L_{obj}^{(j)} \cdot \mathbf{N}(i)$  ▷ Kaiser & Dickman's method
9:    $I_{sim}^{(g,\mu,\lambda)}(i) = |\text{Pareto}(\mathbf{y}, M_{obj,\mu}^{(i)}), M_{obj,\lambda}^{(i)}|_{min}^g$  ▷ Equation 20
10:  for  $k \leftarrow 1, n_{con}$  do
11:     $M_{con,\mu+\lambda}^{(k)}(i) = G_k(\mathbb{X}_{\mu+\lambda}) + L_{con}^{(k)} \cdot \mathbf{N}(i)$  ▷ Kaiser & Dickman's method
12:     $F_{sim}^{(k)}(i) = \prod_{l=1}^{\lambda} \mathcal{H}(\{M_{con,\lambda}^{(k)}\}_l(i))$  ▷ Equation 22
13:  $GMOEI_{c,MC}^{(g,\mu,\lambda)}(\mathbb{X}_{\lambda}) = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} I_{sim}^{(g,\mu,\lambda)}(i) \cdot \prod_{k=1}^{n_{con}} \left( \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} F_{sim}^{(k)}(i) \right)$  ▷ Equation 23

```

---

## 5 Results

A comparative study is set up from the perspective that we have a fixed computational budget and a large number of computation nodes, but are limited in the parallelization of each computer experiment and therefore must resort to batch optimization to make full use of the computational power available. Furthermore, it is assumed that the generation of the next infill is negligible compared to the computational cost of its evaluation. The GAMO algorithm (algorithm 4) is used employing six different settings of  $(g, \mu, \lambda)$ , namely  $(1, 0, 1)$ , which resembles Keane's EI,  $(1, 0, 2)$ , which resembles a multi-objective formulation of the  $q - steps$  EI,

$(1, 1, 1)$ , which resembles an asynchronous formulation of Keane’s EI, and a generalized formulation of the three aforementioned with  $g$  decreasing in a simulated annealing way as will be explained further, so  $g = f(l)$ , with  $l$  the current iteration number. For the DoE we use a Latin-Hypercube Sampling (LHS) approach [26] and Morris and Mitchell’s maximin-criterion to quantify the space-filling property by maximizing in ascending order the distance between pairs of points and simultaneously minimizing the number of corresponding pairs [28]. The stopping criteria used corresponds to a maximum of 100 function evaluations after the DoE or if the normalized expected improvement drops below 0.1%. This means  $(1, 0, 2)$ ,  $(f, 0, 2)$  perform a maximum of 50 iterations, while the other infill criteria stop after a maximum 100 iterations. Alternatively, the wall clock time could be set as a stopping criterion, in which case, under the assumptions above,  $(1, 0, 2)$ ,  $(f, 0, 2)$ ,  $(1, 1, 1)$  and  $(f, 1, 1)$  would be able to evaluate twice as much simulations as  $(1, 0, 1)$  and  $(f, 0, 1)$ . Both criteria are probable to occur in industry, but here we have chosen the former because it allows the assessment of the performance of the infill criteria for a near similar number of evaluations. Furthermore, the selection of a new infill point, especially later in the optimization process at which point the generation of the surrogate model and the sampling of the surrogate become computationally more demanding, can become non-negligible in comparison to the evaluation of a sample. At this point, the asynchronous approaches  $(1, 1, 1)$  and  $(f, 1, 1)$  will outperform the synchronous approaches  $(1, 0, 2)$  and  $(f, 0, 2)$ .

---

**Algorithm 4** Generalized Asynchronous Multi-objective Optimization (GAMO)
 

---

**Require:** Evaluated sampling plan  $(\mathbb{X}, \mathbf{y}(\mathbb{X}))$

**Ensure:** The Pareto front  $(\mathbf{x}^*, \mathbf{y}_{1,2}^*)$

```

1: while stopping criteria are not met do
2:    $\lambda \equiv n(\{\text{completed}(\mathbf{Y}(\mathbb{X}_\mu^{(i)}))\}_{i=1\dots\mu})$            ▷ Determine number of available nodes
3:    $\mathbb{X}_c \leftarrow \mathbb{X}_\mu : \{\text{completed}(\mathbf{Y}(\mathbb{X}_\mu^{(i)}))\}_{i=1\dots\mu}$ 
4:    $\mathbb{X} \leftarrow \mathbb{X} \cup \mathbb{X}_c$ 
5:   Estimate hyperparameters for every objective/constraint           ▷ Section 2
6:    $\mathbf{y}_{1,2}^* \leftarrow \text{Pareto}(\mathbf{y}(\mathbb{X}))$ 
7:    $\mathbf{x}^* \leftarrow \text{argPareto}(\mathbf{y}(\mathbb{X}))$ 
8:    $\mathbb{X}_\mu \leftarrow \mathbb{X}_\mu \setminus \mathbb{X}_c$ 
9:    $\mu \equiv n(\mathbb{X}_\mu)$                                            ▷ Determine number of calculating nodes
10:   $\mathbb{X}_\lambda = \text{argmax}_{\mathbf{x}} \text{GMOEI}_{c,MC}^{(g(l),\mu,\lambda)}(\mathbf{x})$            ▷ Algorithm 3
11:   $\mathbb{X}_\mu \leftarrow \mathbb{X}_\lambda \cup \mathbb{X}_\mu$ 
12:   $\mu \equiv n(\mathbb{X}_\mu)$ 
13:   $l \leftarrow l + 1$                                            ▷ Update iteration counter (table 2)

```

---

We adopt here a different cooling function than Sasena et al. [34] (table 2) based on the observation that decreasing the power of the improvement to zero, which corresponds to the multi-objective probability of improvement, can lead to a stagnation of the optimization process in the form of clustering. Therefore, we spread out the change of power over a larger number of iterations and let it decrease to 1, which corresponds to the first case. By no means can it be expected that the cooling scheme proves to be effective in every case, nor can it be stated that it corresponds to the best performing scheme on a broader average sense, but we expect that placing the emphasis on exploration rather than exploitation early on in the optimization algorithm can accelerate its convergence.

Table 2: Modified cooling scheme based on Sasena et al. [34]

Iteration	1-4	5-9	10-24	25-49	$\geq 50$
$g$ -value	20	10	5	2	1

The methodology is tested on a total of six test functions. The first five test functions are taken from Zitzler et al. [42] (referred to as ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6) and have a particular feature that causes convergence issues for MOO: convexity, non-convexity, discrete Pareto front, multimodality and biased search spaces. All of these have been rewritten in the appendix for six parameters to ensure a certain level of complexity, without using the surrogate outside of its typical application: Kriging is known to have performance issues when dealing with more than 15 parameters or 500 infills [22]. The sixth test function, Osyczka & Kundu’s constrained test problem [30] (abbreviated as OSY), has six parameters and six constraints, all of which are active on different parts of the Pareto front. The solutions and formulas of the aforementioned are given in the appendix. Conform Jones et al’s suggestion, a DoE-size of 65 samples is selected [20].

In their theoretical assessment of performance metrics for multi-objective optimization Zitzler et al. [43] state that performance refers to both quality of outcome and the computational resources needed to generate that outcome. Where the performance of a SOO solving algorithm can be assessed by measuring the difference between the exact minimum and the predicted minimum in function of number of iterations, the MOO case is more complex due to its competing nature: on the one hand we want to assess the closeness to the optimal front solutions and on the other hand we want to assess the coverage of the front. Furthermore, we want both to be obtained with the smallest computational budget possible. However, coverage of the front can only improve further with additional computational resources. Therefore, performance of MOO algorithms cannot be measured with a single metric as pointed out by Van Veldhuizen & Lamont [38]. Riquelme et al. [33] report over 50 metrics used to assess the performance of evolutionary multi-objective optimization (EMO) algorithms of which the hypervolume metric is the most used. Deb et al. [5] presented two metrics to assess the performance: the convergence metric  $\gamma$ , which measures the mean distance of the evaluated Pareto front to the exact one,

$$\bar{\gamma} = \frac{1}{m} \sum_{i=1}^m \gamma_i \quad (24)$$

$$\gamma_i = \min \sqrt{\left(y_1^p - y_1^{*(i)}\right)^2 + \left(y_2^p - y_2^{*(i)}\right)^2} \quad (25)$$



with  $(\mathbf{y}_1^p, \mathbf{y}_2^p)$  the exact Pareto front, and the diversity metric  $\Delta$ , defined as

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{m-1} |d_i - \bar{d}|}{d_f + d_l + (m-1)\bar{d}} \quad (26)$$

$$d_i = \sqrt{\left(y_1^{*(i)} - y_1^{*(i+1)}\right)^2 + \left(y_2^{*(i)} - y_2^{*(i+1)}\right)^2} \quad (27)$$

$$d_f = \sqrt{\left(y_1^{min} - y_1^{*(1)}\right)^2 + \left(y_2(\mathbf{x}|y_1(\mathbf{x}) = y_1^{min}) - y_2^{*(1)}\right)^2} \quad (28)$$

$$d_l = \sqrt{\left(y_1(\mathbf{x}|y_2(\mathbf{x}) = y_2^{min}) - y_1^{*(m)}\right)^2 + \left(y_2^{min} - y_2^{*(m)}\right)^2} \quad (29)$$

$$\bar{d} = \frac{1}{m-1} \sum_{i=1}^{m-1} d_i \quad (30)$$

where  $m$  is the number of points on the front,  $d_i$  is the Euclidean distance between two consecutive points on the front and  $\bar{d}$  its average. Furthermore,  $d_f$  and  $d_l$  represent the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set. The convergence metric is reformulated to determine the correctness of the surrogate around the Pareto front. Even when we run out of computational resources, the cheap to evaluate surrogate can be sampled to refine the regions of the Pareto front which are of particular interest. The convergence metric is defined here as the mean distance between 100 points on the exact Pareto front and their corresponding predicted points. This metric is normalized by dividing it by the largest distance in the objective space. As such  $\bar{\gamma}_{norm} = 0$  corresponds with a perfectly predicted Pareto front. Furthermore,  $\bar{\gamma}$  (the subscript will from now on be omitted for clarity) can be bigger than 1, when the predicted objective space is larger than the exact objective space. Similarly,  $\Delta = 0$  corresponds to a perfectly spread evaluated Pareto front where the extremities of the front are found. Again,  $\Delta$  can be larger than 1. It can be noted that with as little as three points  $\Delta$  can be zero (the extremities and a point exactly in the middle) and with  $\bar{\gamma}_{norm}$  equal to zero, the surrogate predicts the Pareto front perfectly. However, this is by no means an optimal solution. Therefore, the hypervolume of the non-dominated objective space is added as another metric with its reference point at the border of the objective space and normalized by dividing it by the area of the entire objective space.

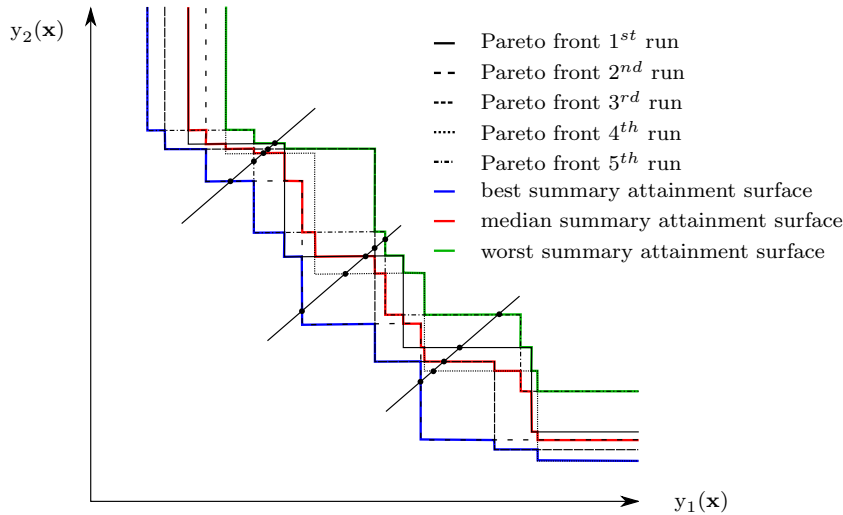


Fig. 4: Schematic representation of the Pareto fronts of five independent runs of an optimizer with three diagonal lines intersecting the aforementioned fronts which allow the formation of the  $p$ -summary attainment surfaces, with the 1<sup>st</sup>, 3<sup>rd</sup> and 5<sup>th</sup> summary attainment surfaces respectively corresponding to the best, median and worst surfaces. Reproduced from Knowles [23].

The stochastic nature of the optimization caused by the randomly generated DoE can lead to a varying performance. To take this aspect into account, the optimization is repeated 10 times and the average and variance of  $\bar{\gamma}_{norm}$  and  $\Delta$  are determined. Furthermore, the  $p$ -summary attainment surfaces are determined, of which the best, median and worst hypervolumes are calculated following Knowles in his analysis of the ParEGO algorithm [23]. Given  $n$  runs of the algorithm, the obtained attainment surfaces/Pareto fronts<sup>6</sup> can be plotted on top of each other. If a diagonal, running in the direction of the improvement in all objectives, is drawn through this set of fronts, it can be stated that the intersections,  $n$  in total, lie on their respective summary attainment surface. The best summary attainment surface thus corresponds to the attainment surface formed by the Pareto optimal points on the intersecting diagonals. If we then remove the points that make up the best summary attainment surface and select the Pareto optimal set of the remaining points, we obtain the 2<sup>nd</sup>-summary surface and so on. We emphasize three of these summary attainment surfaces from a statistical point of view: the best, which corresponds to the best front the algorithm can hope to produce, the worst, which corresponds to the best attainable surface that will be dominated during every run of the algorithm, and the median, which corresponds to the performance that can be expected half of the time. The concept is visualized in figure 4. The hypervolume of the three aforementioned summary attainment surfaces  $\mathcal{S}_p$  is calculated as the final metric to assess convergence and the reference point is placed at the border of the objective space. This metric is rewritten such

<sup>6</sup> The attainment surface was formally defined by Fonseca & Fleming and corresponds to the Pareto front [10]. Both terms are interchangeably used in the optimization community, but in the context of metrics the former is preferred.

Table 3: Results of GAMO for different infill settings and different test functions.

Problem	Metric		Infill settings					
			(1, 0, 1)	(f, 0, 1)	(1, 0, 2)	(f, 0, 2)	(1, 1, 1)	(f, 1, 1)
ZDT1	$\bar{\gamma}$	mean	0.0038	0.0046	0.0624	0.0279	0.0146	0.0077
		var	$4e-6$	$1e-6$	0.0024	$8e-4$	$5e-5$	$2e-5$
	$\Delta$	mean	0.8768	0.8910	0.8568	1.0816	1.3281	0.9983
		var	0.0300	0.0021	0.0099	0.1107	0.0242	0.0106
	$A_S$	best	0.0249	0.0188	0.0313	0.0166	0.0195	0.0137
		median	0.1018	0.1035	0.1787	0.1267	0.2995	0.1192
	worst	0.5708	0.4574	0.6432	0.9452	0.6377	0.3365	
ZDT2	$\bar{\gamma}$	mean	0.0162	0.0163	0.0068	0.0094	0.0239	0.0134
		var	$2e-4$	$1e-4$	$5e-5$	$9e-5$	$2e-4$	$2e-4$
	$\Delta$	mean	1.3444	1.0576	1.1747	1.2369	1.1490	1.1055
		var	0.2011	0.1220	0.1372	0.2952	0.1102	0.1547
	$A_S$	best	0.0337	0.0297	0.0165	0.0246	0.0211	0.0277
		median	0.0349	0.0856	0.0349	0.0359	0.1924	0.0792
	worst	0.7150	0.3884	0.1950	0.1692	0.9911	0.1544	
ZDT3	$\bar{\gamma}$	mean	0.1394	0.0686	0.1167	0.1671	0.1594	0.0891
		var	0.0075	0.0013	0.0057	0.0297	0.0079	0.0037
	$\Delta$	mean	0.9293	1.0589	0.8872	1.0154	1.0364	1.0395
		var	0.0054	0.0286	0.0038	0.0214	0.0078	0.0179
	$A_S$	best	0.0660	0.0345	0.0761	0.0806	0.0451	0.0634
		median	0.1459	0.0757	0.1672	0.1161	0.1530	0.1213
	worst	0.1994	0.2095	0.2090	0.1812	0.1928	0.2051	
ZDT4	$\bar{\gamma}$	mean	0.5829	0.6287	0.6072	0.5599	0.4482	0.4558
		var	0.0034	0.0102	0.0013	0.0087	$2e-4$	$4e-4$
	$\Delta$	mean	1.0175	1.0140	0.9683	1.0152	1.0372	1.0363
		var	0.0068	0.0081	$7e-4$	$6e-4$	0.0077	0.0032
	$A_S$	best	0.1309	0.1012	0.1725	0.1326	0.1298	0.1458
		median	0.2188	0.2170	0.2667	0.2687	0.1809	0.2080
	worst	0.3241	0.3043	0.3804	0.4067	0.2096	0.2381	
ZDT6	$\bar{\gamma}$	mean	0.7514	0.7257	0.6653	0.6642	0.6830	0.6613
		var	0.0072	0.0102	0.0114	0.0238	0.0363	0.0151
	$\Delta$	mean	0.9897	0.9916	1.0102	1.0014	1.0209	1.0273
		var	0.0017	$4e-4$	0.0029	0.0014	0.0065	0.0055
	$A_S$	best	0.6089	0.6183	0.5705	0.5846	0.2558	0.2512
		median	0.7524	0.7338	0.6748	0.7023	0.7347	0.6750
	worst	0.7956	0.7809	0.7347	0.7923	0.8415	0.7129	
OSY	$\bar{\gamma}$	mean	0.0062	0.0058	0.0034	0.0028	0.0147	0.0147
		var	$1e-5$	$4e-6$	$1e-6$	$1e-6$	$3e-5$	$2e-5$
	$\Delta$	mean	1.0734	1.0907	1.1277	1.0604	1.0661	1.0690
		var	0.0151	0.0073	0.0206	0.0249	0.0069	0.0084
	$A_S$	best	0.3278	0.4586	0.3074	0.6182	0.4838	0.4963
		median	0.8588	0.8525	0.7442	0.7265	0.7430	0.7939
	worst	0.9489	0.9065	0.8073	0.8448	0.8871	0.9382	

that  $A_S \in [0, 1]$  with  $A_S = 0$  the perfect solution. This is done by calculating the hypervolume of the exact solution  $\mathcal{S}_e$  and defining the metric as  $A_S = (\mathcal{S}_e - \mathcal{S}_p) / \mathcal{S}_e$ . The use of the median and best/worst instead of mean and variance is preferred due to the heavy-tailed distribution of the metric. The results are presented in table 3.

The definition of the improvement is for every infill the geometric distance away from the Pareto front taken to a higher power, which implies that the difference in performance between the different test functions can be linked to this definition.

*ZDT2*, noted by its non-convex Pareto front, is prone to clustering, which can be seen from a higher value of  $\Delta$ . *ZDT4*, characterized by the multi-modality of the objective space, and *ZDT6*, noted by its strong non-uniformity of the objective space, are both subjected to a higher value of  $\bar{\gamma}$ , which implies that the surrogate is not reproducing the exact Pareto front well. The origin of this higher value of  $\bar{\gamma}$  can be found in the calculation of the GMOEI, which, when using MCI, can be discontinuous and hard to optimize. This feature was also laid bare by Wagner et al. in their assessment of improvement formulations [40]. Possible remedies of this issue would be to improve on the convergence rate of the MCI, using a higher number of samples and thus smoothing the objective or using an alternative optimizer more suited for the problem such as the DIRECT method by Jones [19].

*OSY*, characterized by a set of constraints, has a very good predicted Pareto front, but a very poor attainment surface performance. This implies that the formulation of the expected improvement as the product of the unconstrained expected improvement with the probability of feasibility still leads to the selection of points that do not meet the constraints by augmenting the unconstrained Pareto front later on in the optimization process. Using an alternative formulation of the constraints, for example using a Augmented Lagrangian formulation, or artificially altering the variance on the prediction of the surrogates of the constraints, similar to the generalized formulation of the expected improvement, might bring solace in this matter.

It is expected that  $(*, 0, 1)$  will outperform the other settings since at all times it selects the next infill point with the maximum available information. Preferably the performance of the other settings (synchronous and asynchronous) would be not too strongly inferior to the sequential setting such that the gain that can be obtained in wall clock time, would not be lost due to the need for additional function evaluations to obtain the same evaluated Pareto front. However, comparison of the infill settings for every test function shows varying results. In the case of *ZDT2*, the synchronous settings  $(*, 0, 2)$  clearly outperform both sequential and asynchronous infill. This is caused by the fact that the conditional vector enforces a spread of the infill points, which counters to some extent the clustering which is prone to occur on a non-convex Pareto front. The synchronous settings  $(*, 1, 1)$  outperform the sequential approach in *ZDT4* and *ZDT6*, but is outperformed in *OSY*. The observation that  $(*, 0, 1)$  does not unambiguously outperform the other settings implies that a undeniable speed-up in wall clock time can be obtained through asynchronous optimization.

When examining the generalized formulation  $(f, *, *)$  against the standard formulation  $(1, *, *)$ , again the results vary: in case of *ZDT2*, the generalized formulation leads to a smaller variation in the attainment surfaces for all three infill strategies: sequential, synchronous and asynchronous. This illustrates that the early emphasis on exploration can provide a better assurance on global optimization. For the other test function, the results vary, but the generalized formulations never performs significantly worse than the standard formulation.

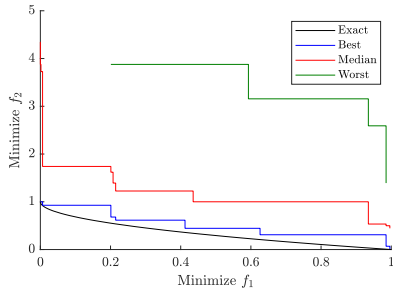
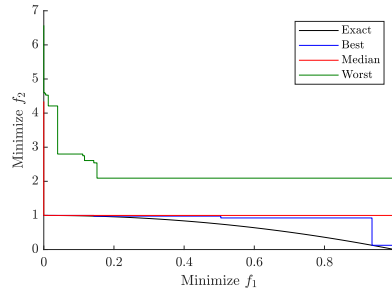
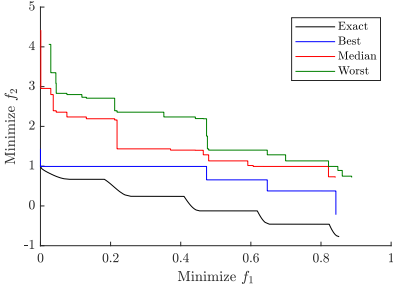
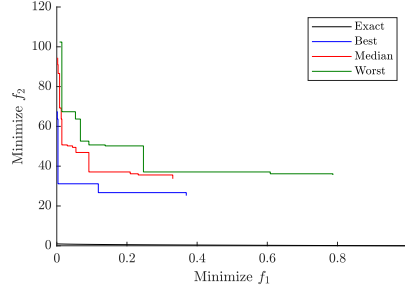
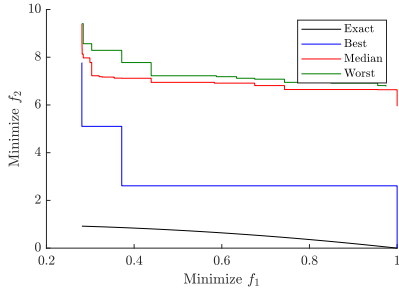
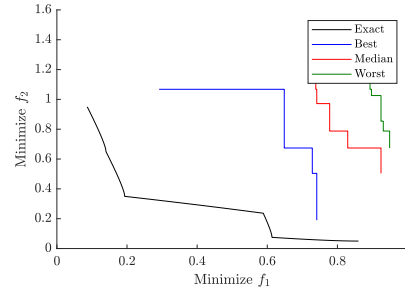
(a) GAMO using  $GMOEI_{MC}^{f,0,1}$  on ZDT1(b) GAMO using  $GMOEI_{MC}^{f,0,2}$  on ZDT2(c) GAMO using  $GMOEI_{MC}^{1,0,1}$  on ZDT3(d) GAMO using  $GMOEI_{MC}^{1,1,1}$  on ZDT4(e) GAMO using  $GMOEI_{MC}^{f,1,1}$  on ZDT6(f) GAMO using  $GMOEI_{MC}^{1,0,2}$  on OSY

Fig. 5: Attainment surfaces (best, worst and median) when the stopping criteria are met for the different test functions

## 6 Conclusion

In this paper a novel infill criterion was formulated: the Generalized Multi-Objective Expected Improvement (GMOEI) for Generalized Asynchronous Multi-objective Optimization (GAMO). The fundamental strength of the method lies in its ability to control exploration and exploitation adaptively and fill the available computational nodes at all times, leading to possibly significant speed-up. The criterion was derived analytically for the sequential infill approach and an algorithm that allows the GMOEI to be calculated for the synchronous/parallel/batch and asynchronous approach. This algorithm relies on Monte Carlo Integration (MCI) and

the calculation of a conditional vector using the Cholesky decomposition. The novel strategy was tested on six test functions showing a varying success of the formulation of the improvement, but nonetheless proves the significant reduction of wall clock time that can be obtained.

The analytical derivation presented in this paper was limited to two objectives. However, by making use of the multinomial theorem, a multiple of objectives can be evaluated. Finding multiple infills simultaneous, while theoretically perfectly feasible, might become impracticable due to the scaling of the dimensionality of the problem. Here a multi-start SQP methodology was followed. However, alternative approaches, such as SIMPLEX, might prove more successful. While effective, the integrated formulation of the constraints in the expected improvement might lead to a loss of optimal solutions early in the optimization. Alternative formulations of the constraints, for example through the Augmented Lagrangian, should be investigated. The theoretical assessment of Wagner et al. [40] showed very promising results for the HV definition of the improvement. Formulation of the former in an asynchronous multi-point manner, as was done in this paper, might push the capabilities of asynchronous surrogate-based multi-objective optimization even further.

**Acknowledgements** The authors would like to thank prof. dr. ir. Jan Vierendeels; his input, supervision, guidance and support during this research has been of critical value.

## Appendix A: Proof of equation 4

$$\begin{aligned}
EI(\mathbf{x}) &= \int_{-\infty}^{\infty} I(\mathcal{Y})\phi(\mathcal{Y}, \mathbf{x})d\mathcal{Y} \\
&= \int_{-\infty}^{y_{min}} (y_{min} - \mathcal{Y}) \frac{1}{s(\mathbf{x})\sqrt{2\pi}} \exp\left[-\frac{(\mathcal{Y} - Y(\mathbf{x}))^2}{2s^2(\mathbf{x})}\right] d\mathcal{Y} \\
&= \int_{-\infty}^{u(\mathbf{x})} (y_{min} - Y(\mathbf{x}) - s(\mathbf{x})\epsilon)\phi(\epsilon)d\epsilon \\
&= (y_{min} - Y(\mathbf{x}))\Phi(u(\mathbf{x})) - s(\mathbf{x}) \int_{-\infty}^{u(\mathbf{x})} \epsilon\phi(\epsilon)d\epsilon \\
&= (y_{min} - Y(\mathbf{x}))\Phi(u(\mathbf{x})) + \frac{s(\mathbf{x})}{\sqrt{2\pi}} \int_{-\infty}^{u(\mathbf{x})} (-\epsilon)\exp\left(-\frac{\epsilon^2}{2}\right) d\epsilon \\
&= (y_{min} - Y(\mathbf{x}))\Phi(u(\mathbf{x})) + \frac{s(\mathbf{x})}{\sqrt{2\pi}} \exp\left(-\frac{\epsilon^2}{2}\right) \Big|_{-\infty}^{u(\mathbf{x})} \\
&= (y_{min} - Y(\mathbf{x}))\Phi(u(\mathbf{x})) + s(\mathbf{x}) (\phi(u(\mathbf{x})) - 0)
\end{aligned} \tag{31}$$

## Appendix B: Test functions

- Zitzler et al.'s first test function (ZDT1),  $\mathcal{F}_1$ , has a convex optimal front with  $m = 6$  and  $x_i \in [0, 1]$ . The front is formed with  $g(\mathbf{x}) = 1$  [42].

$$\begin{aligned}
 f_1(x_1) &= x_1 \\
 g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1) \\
 h(f_1, g) &= 1 - \sqrt{f_1/g} \\
 f_2(\mathbf{x}) &= g \cdot h
 \end{aligned} \tag{32}$$

- Zitzler et al.'s second test function (ZDT2),  $\mathcal{F}_2$ , has a non-convex optimal front with  $m = 6$  and  $x_i \in [0, 1]$ . The front is formed with  $g(\mathbf{x}) = 1$  [42].

$$\begin{aligned}
 f_1(x_1) &= x_1 \\
 f_2(\mathbf{x}) &= g \cdot h \\
 g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1) \\
 h(f_1, g) &= 1 - (f_1/g)^2
 \end{aligned} \tag{33}$$

- Zitzler et al.'s third test function (ZDT3),  $\mathcal{F}_3$ , has a non-continuous convex front caused by the introduction of the sine function in  $h(\mathbf{x})$  with  $m = 6$  and  $x_i \in [0, 1]$ . The front is formed with  $g(\mathbf{x}) = 1$  [42].

$$\begin{aligned}
 f_1(x_1) &= x_1 \\
 f_2(\mathbf{x}) &= g \cdot h \\
 g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m - 1) \\
 h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)
 \end{aligned} \tag{34}$$

- Zitzler et al.'s fourth test function (ZDT4),  $\mathcal{F}_4$ , has a convex front with  $m = 6$  and  $x_i \in [0, 1]$ . There is a multitude of local fronts formed with  $g(\mathbf{x}) = 1.25$  and a global one with  $g(\mathbf{x}) = 1$  [42].

$$\begin{aligned}
 f_1(x_1) &= x_1 \\
 f_2(\mathbf{x}) &= g \cdot h \\
 g(x_2, \dots, x_m) &= 1 + 10(m - 1) + \sum_{i=2}^m ((10x_i - 5)^2 - 10\cos(4\pi(10x_i - 5))) \\
 h(f_1, g) &= 1 - \sqrt{f_1/g}
 \end{aligned} \tag{35}$$

- Zitzler et al.’s sixth test function (ZDT6),  $\mathcal{F}_6$ , has a strong non-uniformity of the search space with the Pareto front found in the lowest density region with  $m = 6$  and  $x_i \in [0, 1]$ . The global front is formed with  $g(\mathbf{x}) = 1$  [42].

$$\begin{aligned}
f_1(x_1) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
f_2(\mathbf{x}) &= g \cdot h \\
g(x_2, \dots, x_m) &= 1 + 9 \cdot \left(\sum_{i=2}^m x_i / (m-1)\right)^{0.25} \\
h(f_1, g) &= 1 - (f_1/g)^2
\end{aligned} \tag{36}$$

- Osyczka & Kundu constrained test problem [30] (OSY). The Pareto front is made up out of 5 sections with different constraints active. The parameter combinations are given in table 4.

$$\begin{aligned}
f_1(\mathbf{x}) &= \frac{-(10x_1 - 2)^2}{60} - \frac{(10x_2 - 2)^2}{300} - \frac{4x_3^2}{75} - \frac{(3x_4 - 2)^2}{150} - \frac{4x_5^2}{75} + 1 \\
f_2(\mathbf{x}) &= \frac{5x_1^2}{9} + \frac{5x_2^2}{9} + \frac{(4x_3 + 1)^2}{300} + \frac{x_4^2}{5} + \frac{5x_5^2}{36} + \frac{5x_6^2}{9} \\
g_1(\mathbf{x}) &= 10x_1 + 10x_2 - 2 \geq 0 \\
g_2(\mathbf{x}) &= 6 - 10x_1 - 10x_2 \geq 0 \\
g_3(\mathbf{x}) &= 2 + 10x_1 - 10x_2 \geq 0 \\
g_4(\mathbf{x}) &= 2 - 10x_1 + 30x_2 \geq 0 \\
g_5(\mathbf{x}) &= 4 - (4x_3 - 2)^2 - 6x_4 \geq 0 \\
g_6(\mathbf{x}) &= (4x_5 - 2)^2 + 10x_6 - 4 \geq 0
\end{aligned} \tag{37}$$

Table 4: Solution of the Osyczka & Kundu constrained test problem [30] taken from Deb et al. [6]

Region	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	Constraints
AB	5	1	(1,...,5)	0	5	0	2,4,6
BC	5	1	(1,...,5)	0	1	0	2,4,6
CD	(4.056,...,5)	$(x_1-2)/3$	1	0	1	0	4,5,6
DE	0	2	(1,...,3.732)	0	1	0	1,3,6
EF	(0,...,1)	$2-x_1$	1	0	1	0	1,5,6

## References

1. Chevalier, C.: Fast Computation of the Multi-Points Expected Improvement with Applications in Batch Selection (2013). DOI 10.1007/978-3-642-44973-4\_7
2. Conn, A.R., Gould, N.I.M., Toint, P.L.: A Comprehensive Description of the Mathematical Algorithms Used in LANCELOT, pp. 102–132. Springer Berlin Heidelberg, Berlin, Heidelberg (1992). DOI 10.1007/978-3-662-12211-2\_3
3. Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization* **60**(3), 575–594 (2014). DOI 10.1007/s10898-013-0118-2
4. Couckuyt, I., Dhaene, T., Demeester, P.: ooDACE toolbox: a flexible object-oriented kriging implementation. *J. Mach. Learn. Res.* **15**(1), 3183–3186 (2014)



5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002). DOI 10.1109/4235.996017
6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization, pp. 105–145. Springer London, London (2005). DOI 10.1007/1-84628-137-7\_6
7. Emmerich, M., Deutz, A., W Klinkenberg, J.: The computation of the expected improvement in dominated hypervolume of pareto front approximations. Report Technical Report 4-2008, Leiden Institute of Advanced Computer Science (2008)
8. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation discussion and generalization (1993)
9. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* **3**(1), 1–16 (1995). DOI 10.1162/evco.1995.3.1.1
10. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: H.M. Voigt, W. Ebeling, I. Rechenberg, H.P. Schwefel (eds.) *Parallel Problem Solving from Nature — PPSN IV*, pp. 584–593. Springer Berlin Heidelberg (1996)
11. Forrester, A., Sobester, A., Keane, A.: *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley (2008)
12. Gentle, J.E.: *Computational Statistics*. Springer, New York, USA (2009). DOI 10.1007/978-0-387-98144-4
13. Ginsbourger, D., Janusevskis, J., Le Riche, R.: Dealing with asynchronicity in parallel gaussian process based global optimization. Report (2011). URL <https://hal.archives-ouvertes.fr/hal-00507632>
14. Ginsbourger, D., Le Riche, R., Carraro, L.: Kriging Is Well-Suited to Parallelize Optimization, pp. 131–162. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-10701-6\_6
15. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pp. 82–87 vol.1 (1994). DOI 10.1109/ICEC.1994.350037
16. Janusevskis, J., Le Riche, R., Ginsbourger, D.: Parallel expected improvements for global optimization: summary, bounds and speed-up. Report (2011). URL <https://hal.archives-ouvertes.fr/hal-00613971>
17. Janusevskis, J., Le Riche, R., Ginsbourger, D., Girdziusas, R.: Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In: Y. Hamadi, M. Schoenauer (eds.) *Learning and Intelligent Optimization*, pp. 413–418. Springer Berlin Heidelberg (2012)
18. Jeong, S., Obayashi, S.: Efficient global optimization (EGO) for multi-objective problem and data mining. In: *2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2138–2145 Vol. 3 (2005). DOI 10.1109/CEC.2005.1554959
19. Jones, D.R.: *Direct global optimization algorithm*, pp. 431–440. Springer US, Boston, MA (2001). DOI 10.1007/0-306-48332-7\_93
20. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13**(4), 455–492 (1998). DOI 10.1023/A:1008306431147
21. Kaiser, H.F., Dickman, K.: Sample and population score matrices and sample correlation matrices from an arbitrary population correlation matrix. *Psychometrika* **27**(2), 179–182 (1962). DOI 10.1007/BF02289635
22. Keane, A.J.: Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal* **44**(4), 879–891 (2006). DOI 10.2514/1.16875
23. Knowles, J.: Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **10**(1), 50–66 (2005). DOI 10.1109/TEVC.2005.851274
24. Knowles, J., Corne, D.: *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation*, vol. 1 (1999). DOI 10.1109/CEC.1999.781913
25. Marmin, S., Chevalier, C., Ginsbourger, D.: Differentiating the multipoint expected improvement for optimal batch design. In: P. Pardalos, M. Pavone, G.M. Farinella, V. Cutello (eds.) *Machine Learning, Optimization, and Big Data*, pp. 37–48. Springer International Publishing (2015)

26. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979). DOI 10.2307/1268522
27. Mockus, J., Tiesis, V., Zilinskas, A.: The application of bayesian methods for seeking the extremum. In: L.D. 2, G. Szego (eds.) *Towards Global Optimization 2: Proceedings of a Workshop at the University of Cagliari, Italy, October 1974*, vol. 2, pp. 117–129 (1978)
28. Morris, M.D., Mitchell, T.J.: Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43**(3), 381–402 (1995). DOI 10.1016/0378-3758(94)00035-T
29. Mueller, T.J., DeLaurier, J.D.: Aerodynamics of small vehicles. *Annual Review of Fluid Mechanics* **35**, 89–111 (2003). DOI 10.1146/annurev.fluid.35.101101.161102
30. Osyczka, A., Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural optimization* **10**(2), 94–99 (1995). DOI 10.1007/BF01743536
31. Parr, J.M., Keane, A.J., Forrester, A.I.J., Holden, C.M.E.: Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization* **44**(10), 1147–1166 (2012). DOI 10.1080/0305215X.2011.637556
32. Ponweiser, W., Wagner, S., Vincze, M.: Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3515–3522 (2008). DOI 10.1109/CEC.2008.4631273
33. Riquelme, N., Von Lücken, C., Barán, B.: Performance metrics in multi-objective optimization. In: *XLI Latin American Computing Conference (CLEI)* (2015)
34. Sasena, M., Papalambros, P., Goovaerts, P.: Exploration of meta-modeling sampling criteria for constrained global optimization. *Engineering Optimization* **34**, 263–278 (2002)
35. Sóbester, A., Leary, S.J., Keane, A.J.: A parallel updating scheme for approximating and optimizing high fidelity computer simulations. *Structural and Multidisciplinary Optimization* **27**(5), 371–383 (2004). DOI 10.1007/s00158-004-0397-9
36. Schonlau, M., Welch, W.J., Jones, D.R.: Global versus local search in constrained optimization of computer models, *Lecture Notes–Monograph Series*, vol. Volume 34, pp. 11–25. Institute of Mathematical Statistics, Hayward, CA (1998). DOI 10.1214/lnms/1215456182
37. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* **2**(3), 221–248 (1994). DOI 10.1162/evco.1994.2.3.221
38. Van Veldhuizen, D.A., Lamont, G.B.: On measuring multiobjective evolutionary algorithm performance. In: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, pp. 204–211 vol.1 (2000). DOI 10.1109/CEC.2000.870296
39. Voutchkov, I., Keane, A.: *Multi-Objective Optimization Using Surrogates*, pp. 155–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-12775-5\_7
40. Wagner, T., Emmerich, M., Deutz, A., Ponweiser, W.: On expected-improvement criteria for model-based multi-objective optimization. In: R. Schaefer, C. Cotta, J. Kolodziej, G. Rudolph (eds.) *Parallel Problem Solving from Nature, PPSN XI*, pp. 718–727. Springer Berlin Heidelberg (2010)
41. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation* **14**(3), 456–474 (2010). DOI 10.1109/TEVC.2009.2033671
42. Zitzler, E., Deb, K., Thiele, F.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* **8**(2), 173–195 (2000). DOI 10.1162/106365600568202
43. Zitzler, E., Thiele, F., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132 (2003). DOI 10.1109/TEVC.2003.810758
44. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: A.E. Eiben, T. Bäck, M. Schoenauer, H.P. Schwefel (eds.) *Parallel Problem Solving from Nature — PPSN V*, pp. 292–301. Springer Berlin Heidelberg (1998)