

# sParEGO – A Hybrid Optimization Algorithm for Expensive Uncertain Multi-Objective Optimization Problems

João A. Duro<sup>1</sup>, Robin C. Purshouse<sup>1</sup>, Shaul Salomon<sup>1,2</sup>, Daniel C. Oara<sup>1</sup>,  
Visakan Kadirkamanathan<sup>1</sup>, and Peter J. Fleming<sup>1</sup>

<sup>1</sup> University of Sheffield, UK {j.a.duro,r.purshouse}@sheffield.ac.uk

<sup>2</sup> ORT Braude College of Engineering, Israel, shaulsal@braude.ac.il

**Abstract.** Evaluations of candidate solutions to real-world problems are often expensive to compute, are characterised by uncertainties arising from multiple sources, and involve simultaneous consideration of multiple conflicting objectives. Here, the task of an optimizer is to find a set of solutions that offer alternative robust trade-offs between objectives, where robustness comprises some user-defined measure of the ability of a solution to retain high performance in the presence of uncertainties. Typically, understanding the robustness of a solution requires multiple evaluations of performance under different uncertain conditions – but such an approach is infeasible for expensive problems with a limited evaluation budget. To overcome this issue, a new hybrid optimization algorithm for expensive uncertain multi-objective optimization problems is proposed. The algorithm – sParEGO – uses a novel uncertainty quantification approach to assess the robustness of a candidate design without having to rely on expensive sampling techniques. Hypotheses on the relative performance of the algorithm compared to an existing method for deterministic problems are tested using two benchmark problems, and provide preliminary indication that sParEGO is an effective technique for identifying robust trade-off surfaces.

**Keywords:** Expensive optimization, surrogate-based optimization, robust optimization, multi-objective optimization

## 1 Introduction

The ability of simulations to predict the performance of a candidate design is constantly increasing. While some simulations can produce high-fidelity outputs relatively quickly, a typical mesh-based simulation can run for several hours, and even days. Even if a design team has access to supercomputing resources, the extensive run-time still implies that perhaps only a few hundred candidate designs can be explored using high-fidelity modelling resources. Unfortunately, conventional multi-objective optimization algorithms implemented in commercial packages typically require tens of thousands of function evaluations to converge on a high quality solution [1]. Therefore, the search for a promising design

using expensive evaluation functions on a limited computational budget poses a great challenge.

To exacerbate this problem, optimizing for a robust solution is itself a computationally demanding task. In order to gain confidence over the robustness of a solution to uncertainties, the statistical properties of the expected solution’s performance must be quantified. In a world where the complexity of the high-fidelity models essentially produces a black-box mapping of inputs to outputs, such statistical properties would typically be found through repeated evaluation of the same solution using those high-fidelity models. However, repeatedly sampling a single candidate design is computationally expensive.

To address the above, we propose a framework for expensive uncertain multi-objective optimization problems. The key aims are to: (i) exploit expensive, black-box evaluation function for a candidate design; (ii) account for multiple sources of uncertainty, such as fidelity of evaluation functions and manufacturing tolerances; and (iii) provide an understanding of the risk and opportunity trade-offs between candidate designs with respect to a given robustness metric. To achieve this the framework leverages ParEGO [2], an algorithm for multi-objective optimization, which has been demonstrated to provide good results for optimization runs limited to a small number of function evaluations. ParEGO itself is a multi-objective extension to Jones et al.’s [3] seminal efficient global optimization (EGO) algorithm for single-objective problems. The main limitation of ParEGO is that it has not been designed to handle problems featuring uncertainty (although there is some evidence that it can perform favourably in noisy environments [4]). Therefore a fundamental part of the framework is how ParEGO can be extended to consider evaluation functions as samples of random variates. We refer to this new algorithm as *stochastic ParEGO* or *sParEGO*.

In the remainder of this paper, first the robustness metric used is described in Section 2, and the proposed framework is presented in Section 3. The test suite used is defined in Section 4, while the experimental settings and obtained results are in Section 5. The paper concludes with Section 6.

## 2 Threshold-based robustness metric

A general single-objective robust optimisation problem can be formulated as:

$$\min_{\mathbf{x} \in \Omega} S = \mathbf{f}(\mathbf{x}, \mathbf{U}). \quad (1)$$

Here,  $\mathbf{U}$  is a vector of random variables that includes all the uncertainties associated with the optimisation problem. These uncertainties may be an outcome of manufacturing tolerances, a noisy environment, evaluation inaccuracies etc. A single scenario of the variate  $\mathbf{U}$  is denoted as  $\mathbf{u}$ . Since uncertainties are involved, the scalar objective  $S$  is also a random variate, where every scenario of the uncertainties,  $\mathbf{u}$ , is associated with an objective value  $s$ .

In a robust optimisation scheme, the random objective value is replaced with a robustness criterion, denoted by the indicator  $I[S]$ . Several criteria are com-

monly used in the literature, which can be broadly categorised into three main approaches:

1. **Worst-Case Scenario.** The worst objective vector, considering a bounded domain in the neighbourhood of the nominal values of the uncertain variables.
2. **Aggregated Value.** An integral measure of robustness that amalgamates the possible values of the uncertain variables (e.g. mean value or variance).
3. **Threshold Probability.** The probability for the objective function to be better than a defined threshold.

In this framework the third approach, suggested by Beyer and Sendhof [5], is used. A threshold  $q$  is considered as a satisfying performance for the objective value  $s$ . When  $s$  is uncertain, denoted by the random variable  $S$ , the probability for  $S$  to satisfy the threshold level can be seen as a confidence level  $c$ . For a minimization problem this can be written as:

$$c(S, q) = \Pr(S < q). \quad (2)$$

A robustness indicator used in this paper is based on minimization of the threshold  $q$  for a pre-defined confidence level  $c$ , meaning that the confidence in the resulting performance can be specified (say by a decision-maker).

A stochastic unconstrained multi-objective optimization problem (MOP), which is the focus of this study, can be formulated as:

$$\min_{\mathbf{x} \in \Omega} \mathbf{Z} = \mathbf{f}(\mathbf{x}, \mathbf{U}). \quad (3)$$

where  $\mathbf{x}$  is a vector of  $n_x$  decision variables in a feasible domain  $\Omega$ ,  $\mathbf{Z}$  is a multivariate random vector of  $n_z$  performance criteria and  $\mathbf{f}$  is a set of functions mapping from decision-space to objective-space. Due to uncertainties over the problem parameters or the mapping functions themselves, every evaluation of the same decision vector may result in a different realisation of the objective vector  $\mathbf{z}$ .

### 3 The Framework of the sParEGO Algorithm

#### 3.1 General Framework

sParEGO is a surrogate-based multi-objective optimization algorithm for dealing with stochastic multi-objective optimization problems (MOPs). The algorithm shares many similarities with ParEGO including the ability to approximate expensive MOPs over a realistically small number of function evaluations. The main idea is that the uncertain distribution in objective space of every candidate solution is not quantified through uncertainty quantification methods (e.g. Monte Carlo sampling). Instead, every solution is evaluated once, and the distribution is approximated based on the performance of nearby solutions. A pseudo-code of sParEGO is presented in Algorithm 1. Its stages are explained in detail in the remainder of this section.

---

**Algorithm 1** SPAREGO Pseudo-code

---

**Parameters:** initial set size  $n_{\text{init}}$ , robustness criterion  $I$ , evaluation functions  $\mathbf{f}$ ,  
neighbourhood distance  $\delta$ , number of reference direction vectors  $n_d$

```
1:  $\mathcal{D} \leftarrow$  a set of all reference direction vectors
2:  $\mathcal{X} \leftarrow$  initialise a set with neighbourhoods of solutions ▷ Section 3.3
3:  $\mathcal{Z} \leftarrow \mathbf{f}(\mathcal{X})$  ▷ evaluate the initial set
4: while stopping criteria not satisfied do
5:   Shuffle the set  $\mathcal{D}$ 
6:   for all  $\mathbf{d} \in \mathcal{D}$  do
7:     for all  $\mathbf{x}^i \in \mathcal{X}$  do
8:       update neighbourhood  $\mathcal{N}^i$  ▷ Section 3.4
9:     end for
10:    update ideal and nadir vectors
11:     $\mathcal{S} \leftarrow$  calculate scalar fitness value of all solutions
12:     $\mathcal{I} \leftarrow \emptyset$ 
13:    for all  $\mathbf{x}^i \in \mathcal{X}$  do
14:      approximate the distribution of  $S^i$  ▷ Section 3.4
15:      calculate robustness indicator  $I[S^i]$  ▷ Section 3.5
16:       $\mathcal{I} \leftarrow \mathcal{I} \cup I[S^i]$ 
17:    end for
18:     $model \leftarrow$  fit a Surrogate model to the indicator values  $\mathcal{I}$  ▷ Section 3.6
19:     $\mathbf{x}^{\text{new}} \leftarrow$  maximize the expected improvement based on  $model$ 
20:     $\mathbf{x}^{\text{pert}} \leftarrow$  add a neighbour to  $\mathbf{x}^{\text{new}}$  ▷ Section 3.6
21:     $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{x}^{\text{new}}, \mathbf{x}^{\text{pert}}\}$ 
22:     $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{\mathbf{f}(\mathbf{x}^{\text{new}}), \mathbf{f}(\mathbf{x}^{\text{pert}})\}$  ▷ evaluate the new solutions
23:  end for
24: end while
```

---

### 3.2 Parameters

- $\delta$  – Neighbourhood distance. Maximum Euclidean distance between solutions, in normalised decision space (i.e. between 0 and 1), to be considered as neighbours. Default value is  $0.1\sqrt{n_x}$ .
- $n_{\text{init}}$  – Size of initial population.
- $n_{\text{indp}}$  – Portion of solutions in the initial population that are independently generated. Must be smaller or equal to  $n_{\text{init}}/2$ .
- $\delta_{\text{pert}}$  – Maximum distance between newly generated solutions. Default value is  $\delta/2$ .
- $c$  – Desired level of confidence in the performance of the solutions.
- $n_{\text{max}}$  – Maximum number of solutions to construct the Surrogate model.

### 3.3 Initialisation

To provide good initial coverage of potential designs, a space filling design (Latin hypercube) is used for the initial set. However, the algorithm requires all solutions to reside within a distance of  $\delta$  from other solutions (at least one). To accomplish this, only  $n_{\text{indp}} \leq n_{\text{init}}/2$  solutions are generated by Latin hypercube sampling. The rest of the solutions are generated as follows:

1. For every existing solution, another solution is randomly generated within a hypersphere with a radius of  $\delta_{\text{pert}}$ .
2. The rest of the solutions are generated by randomly selecting an existing solution and creating a nearby solution within a distance of  $\delta_{\text{pert}}$ .

The first stage enforces that every solution has at least one neighbour. The second stage seeds the initial population with neighbourhoods of different sizes.

### 3.4 Uncertainty quantification

The most important difference between sParEGO and ParEGO is that the former assumes that the outcome of an evaluation function is a realization of a random variate. Therefore, the scalarised function value cannot be used directly to construct the surrogate model, and a utility indicator value is used instead. For every direction vector, the surrogate model is constructed to search for a design that will optimize a given robustness indicator (described later in Section 3.5). The guiding principle is to avoid having to repeatedly sample every candidate design to assess its statistical properties in objective-space. Instead, these properties (specifically, measures of central tendency and dispersion) are approximated from the available information of other candidate design evaluations.

**Approximation of the central tendency** The stochasticity of the problem might originate from a variety of sources, including variations in decision-space. For this type of uncertainty, two designs with similar nominal values can be identical when realised. Therefore, the performance of a candidate design should be calculated from the performance of neighbouring designs as well. A distance<sup>3</sup> in normalised<sup>4</sup> decision space  $\delta$  is defined, such that two solutions  $\mathbf{x}^i$  and  $\mathbf{x}^j$  are considered as neighbours if:

$$\|\mathbf{x}^i - \mathbf{x}^j\|_2 \leq \delta. \quad (4)$$

For a solution  $\mathbf{x}^i$ , the statistical properties of the scalar fitness are approximated from the neighbouring solutions as follows: First, the neighbourhood  $\mathcal{N}^i$  of the solution is defined<sup>5</sup>:

$$\mathcal{N}^i = \{\mathbf{x}^j \in \mathcal{X} \mid \|\mathbf{x}^i - \mathbf{x}^j\|_2 \leq \delta\}. \quad (5)$$

Next, the approximated mean function value,  $\mu_s$ , is derived from the neighbourhood. Members that are closer to  $\mathbf{x}^i$  are given a larger weight, denoted as  $v$  in Equation (6), in approximating its properties. Since the weight of most solutions in the neighbourhood is smaller than 1, the overall "neighbourhood size"  $N^i$  is

---

<sup>3</sup> Distance is measured in decision-space by the Euclidean norm.

<sup>4</sup> The decision vectors are normalised between 0 and 1.

<sup>5</sup> Note that according to (5),  $\mathbf{x}^i$  is included in the neighbourhood  $\mathcal{N}^i$ .

smaller than  $|\mathcal{N}^i|$ :

$$v^j = \frac{\delta - \|\mathbf{x}^i - \mathbf{x}^j\|_2}{\delta}, \quad \forall \mathbf{x}^j \in \mathcal{N}^i, \quad (6)$$

$$N^i = \sum_{\mathbf{x}^j \in \mathcal{N}^i} v^j, \quad (7)$$

$$\mu_s^i = \sum_{\mathbf{x}^j \in \mathcal{N}^i} v^j s^j / N^i. \quad (8)$$

**Approximation of the dispersion** Once the expected mean is known, the expected value for the variance is calculated:

$$\sigma_s^{2,i} = \sum_{\mathbf{x}^j \in \mathcal{N}^i} v^j (s^j - \mu_s^i)^2 / N^i. \quad (9)$$

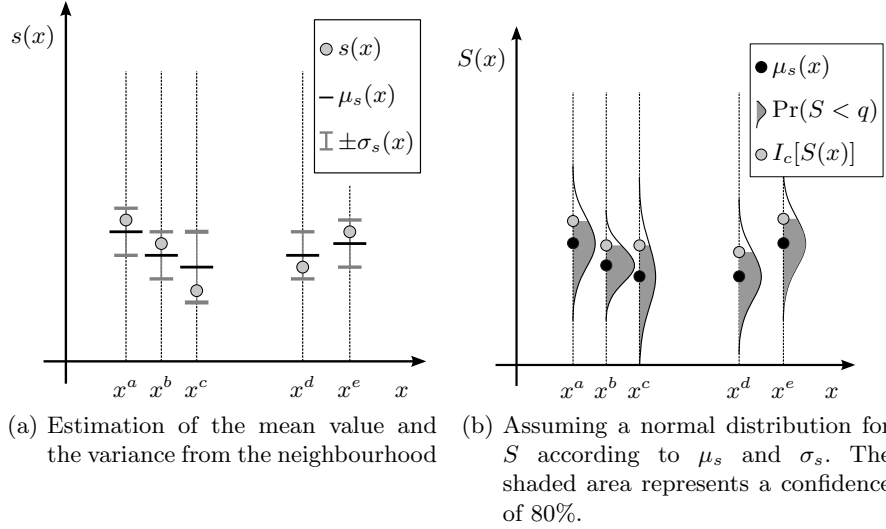


Fig. 1: Approximation of the statistical properties, and estimation of robustness indicator  $I_c[\bullet]$ .

### 3.5 Estimating the robustness indicator value

Once the mean and variance of the scalar fitness function have been estimated for every solution in the set, the random variable  $S(\mathbf{x})$  is assumed to follow a normal distribution. Given a desired confidence level  $c$ , the robustness indicator  $I_c[S]$  can be calculated. An example for  $I_c[S]$  is given in Figure 1(b) where  $I_c[S]$

value corresponds to the 80<sup>th</sup> percentile of the normal distribution. This implies that the obtained  $I_c[S]$  value depends on the chosen confidence. Following this, the indicator value  $I_c[S]$  is considered as the solution's fitness at the current iteration.

### 3.6 Fitting a Surrogate Model to the Fitness

Now that every solution is associated with a scalar fitness value based on the robust indicator, the algorithm proceeds in a similar fashion to EGO and ParEGO [3, 6]. A surrogate model is fitted to the fitness values, and the expected improvement function is constructed from the model. This requires the variance in the expected improvement function ( $\hat{\sigma}^2$ ) to be estimated. For this we use the density of the solutions in decision-space by knowing that the variance has an inverse correlation to the density of the solutions. A suitable way to estimate the density at a given point  $\mathbf{x}$  is to use a non-parametric statistical approach, and in this case we use a kernel density model given by:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{(2\pi h^2)^{D/2}} \exp \left\{ -\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2h^2} \right\}, \quad (10)$$

where  $N$  is the number of points,  $D$  is the number of dimensions, and  $h$  is the bandwidth. We let the bandwidth be equal to  $1/100^{\text{th}}$  of the mean span of all points, that is:

$$h = \frac{1}{100N} \sum_{n=1}^N (\max(\mathbf{x}_n) - \min(\mathbf{x}_n)). \quad (11)$$

Based on experimental results we have observed that the kernel density model can be very sensitive to any changes in the density, thus we have used a smoothing function (in this case the arctan function), and the estimated variance at input  $\mathbf{x}$  is given by:

$$\hat{\sigma}^2(\mathbf{x}) = \left( \frac{1}{\pi/2} \arctan \left( \frac{1}{p(\mathbf{x})} \right) \right)^2. \quad (12)$$

Above a certain size (approximately 50 solutions), the surrogate model becomes prohibitively expensive to construct. When the number of evaluated solutions exceed this size, a subset of size  $n_{\text{max}}$  is chosen according to Algorithm 2.

The next step is to search for the solution that maximises the expected improvement. For this task, any off-the-shelf single-objective optimizer can be used, and we have chosen ACROMUSE [7]. Once a solution  $\mathbf{x}^{\text{new}}$  that maximizes the expected improvement is identified, it is added to the population together with a neighbouring solution  $\mathbf{x}^{\text{Pert}}$ , generated in a similar fashion to the initial solutions described in Section 3.3.

## 4 Test Suite

The test suite used in this paper consists of two variants of WFG4 [8]. Both problems have two objectives and five decision variables. For the first problem,

---

**Algorithm 2** Choosing a Subset to Construct the Surrogate Model

---

**Require:** population set  $\mathcal{X}$ , subset size  $n_{\max}$ , current direction vector  $\mathbf{d}$

**Ensure:** a subset  $\mathcal{X}'$  of size  $n_{\max}$

- 1:  $\mathcal{X}' \leftarrow n_{\max}/2$  solutions from  $\mathcal{X}$  with the best indicator value
  - 2:  $\mathcal{X}'' \leftarrow \mathcal{X} \setminus \mathcal{X}'$
  - 3: **for all**  $\mathbf{x} \in \mathcal{X}''$  **do**
  - 4:    $\hat{\mathbf{z}}(\mathbf{x}) \leftarrow$  normalise  $\mathbf{z}(\mathbf{x})$  and project it on the  $k - 1$  simplex
  - 5:    $\Delta(\mathbf{x}, \mathbf{d}) \leftarrow \|\hat{\mathbf{z}}(\mathbf{x}) - \mathbf{d}\|_2$
  - 6: **end for**
  - 7:  $\mathcal{X}' \leftarrow \mathcal{X}' \cup n_{\max}/2$  solutions from  $\mathcal{X}''$  with the smallest  $\Delta$  distance
- 

namely P1, we have modified WFG4 to increase the density and the number of local optima in the periphery of the global optimum. This simulates the effect that stochasticity can have when approaching the Pareto-optimal Front (PF). The second problem, namely P2, is characterised by having a more smooth landscape with no local minima surrounding the global optimum, and stochasticity is added by the toolkit from [9].

The modification applied to WFG4 is as follows. The original formulation of WFG4 applies a transformation to each input parameter ( $y$ ) given by:

$$\begin{aligned} \text{s\_multi}(y, A, B, C) &= (1 + \cos(r_2) + B(r_1)^2) / (B + 2), \\ r_1 &= |y - C| / (\lfloor C - y \rfloor + C), \\ r_2 &= (4A + 2)\pi(0.5 - 0.5r_1), \end{aligned} \tag{13}$$

where  $A$  controls the number of minima,  $B$  controls the magnitude of the “hill sizes” of the multi-modality, and  $C$  is the value for which  $y$  is mapped to zero. We propose a modification to Equation 13 as follows:

$$\begin{aligned} \text{s\_multi}^*(y, A, B, C, D, E) &= (1 + \cos(r_2 r_3) + B|r_1|^E) / (B + 2), \\ r_3 &= (1 - |r_1|)^{2D}, \end{aligned} \tag{14}$$

where  $D$  controls the density of the hills around the optimum, and  $E$  specifies the polynomial order of the base curve. The effect of these parameters is shown in Figure 2, in that:

1. the number of minima increases up-to  $2A + 1 = 11$  for  $A = 5$ , which includes the global optimum at  $C = 0.45$ , as shown in Figure 2(a);
2. density of hills around the optimum increases with an increase in  $D$  as shown in Figure 2(b); and
3. the proximity of local minima from the value zero decreases with an increase in  $E$  as shown in Figure 2(c).

The toolkit from [9] is used here to transform the objective vectors of WFG4 into random vectors. The parameters have been chosen to ensure that uncertainty increases towards more optimal regions, and perturbation is applied to the objective vector in only one direction (known as the radial component),



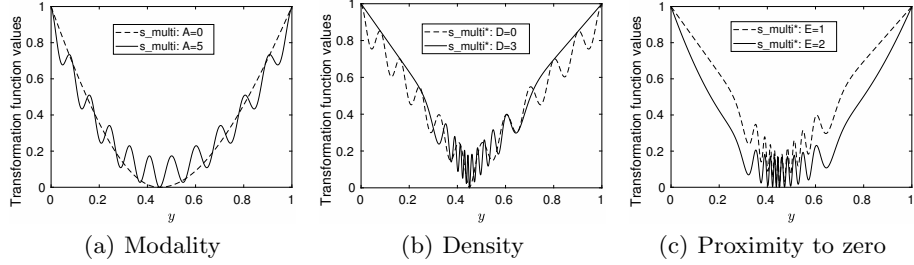


Fig. 2: Transformation function in WFG4 as a function of the input parameter ( $y$ ). The parameters  $B = 10$  and  $C = 0.45$  are kept fixed for all cases. For (b)  $A = 5$ ,  $E = 1$  and for (c)  $A = 5$ ,  $D = 3$ .

hence the perturbation radius is set to zero. The radial component uses a uniform distribution with a lower bound ( $l_b$ ) which depends on the position of the normalised objective vector with respect to the reference vector  $(2/3, 1.0)$ . Another parameter of the distribution is locality ( $l$ ), and its value is determined by an increasing function that uses the remoteness of the objective vector raised to the power of 0.4. The upper bound is given by  $ub = 1 - l(1 - lb)$ .

Following the above, we have chosen  $C = 0.45$  for all test instances. The remaining parameters are:  $A = 5$ ,  $B = 10$ ,  $D = 3$ , and  $E = 1$  for P1; and  $A = 0$ ,  $B = 8$ ,  $D = 0$ , and  $E = 2$  for P2. The obtained PF for P1 and P2 are shown in Figures 3(a) and 3(b), respectively. The performance set for P1 and P2 have been obtained by fixing the values of the last three decision variables to 2.1, 2.8 and 3.5, respectively, as suggested in the WFG toolkit paper [8]. The values of the first two decision variables have been determined by NSGA-II [10]<sup>6</sup> with a population size of 300 solutions, since crowding distance is able to provide a reasonable diverse set across the PF. The robust set for P2 has been obtained by replacing the uniform distribution of the radial component by the 90th percentile of the same distribution (assuming a confidence level of 90%). NSGA-II is used again with the same settings to determine the values of the decision variables that better satisfy the chosen robustness indicator.

Notably for P2 as shown in Figure 3(c), the decision vectors that are Pareto-optimal with respect to the performance criterion are different from the decision vectors that are Pareto-optimal with respect to the robustness criterion. The implication is that an optimal solution set cannot satisfy the performance and robustness criterion simultaneously. Therefore, we use the terms performance PF and robust PF to refer to objective vectors that are Pareto-optimal with respect to either the performance or robustness criterion.

<sup>6</sup> NSGA-II has been run for 10000 generations with the following settings: the distribution index of SBX-crossover and polynomial mutation set to 15 and 20, respectively; and probability of crossover and mutation set to 90% and 10%, respectively.

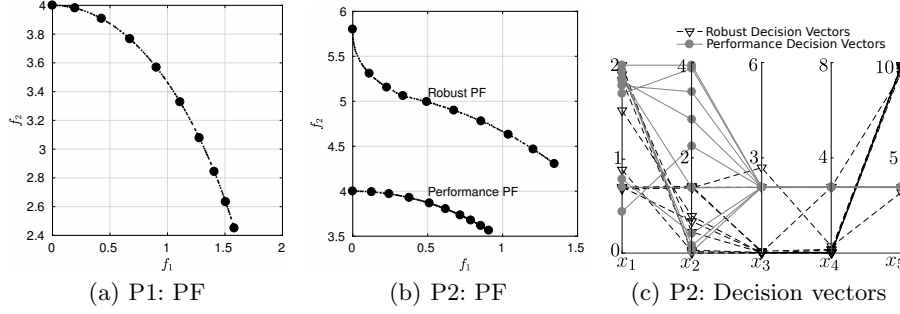


Fig. 3: Performance set for P1, and robust and performance sets for P2.

## 5 Experiment: Comparison of ParEGO and sParEGO

### 5.1 Experimental Settings

In both ParEGO and sParEGO algorithms the scalarisation function corresponds to the augmented Tchebycheff and the simplex lattice approach is used for the generation of weight vectors, both these approaches have been taken from [2]. The weight vectors are modified by Generalised Decomposition approach [11] to provide a better estimate on the PF and we refer to them as reference direction vectors. The parameter settings for both algorithms are as follows:  $n_{init} = 10$ ,  $n_d = 10$ ,  $n_{max} = 50$ , and the optimization budget is set to 1000 function evaluations. For sParEGO the confidence level ( $c$ ) is set to 90%. Inverted Generalisation Distance (IGD) [12] is used to measure the quality of the obtained sets by the optimizers. The 10 solutions that are marked with a filled circle in Figure 3 are used as the reference sets for IGD, and these solutions correspond to the best trade-offs for the chosen direction vectors.

### 5.2 Results and Observations

This section presents the experimental results for P1 and P2 problems. The results shown in Figure 4 provide both a visual and a analytical assessment of the quality of the solutions obtained by the optimizers in terms of their convergence to and diversity across the PF. The objective vectors for P2 have been determined by evaluating 100 times each decision vector, obtained at the end of the optimization run, and the robust performance of each objective is equal to the 90% percentile of its own marginal distribution. The performance of the objective vectors for P1 is equal to their (original) nominal performance since the problem does not contain uncertainty.

For P1, ParEGO approximation to the performance PF is better than sParEGO as shown in Figures 4(a) and 4(c). ParEGO chosen solutions are not fully on the PF, but it is expected for its convergence to improve with more function evaluations since the trend shown in Figure 4(c) indicates a declining IGD value. The same trend is captured for sParEGO, despite the fact that its convergence is

worse than ParEGO. This indicates that the multi-modality in P1, close to the PF, is interpreted by sParEGO as a region of high uncertainty. The robust performance of the solutions in this region is captured as being poor by the statistical inferences made by sParEGO. Hence, most sParEGO solutions are just outside the region where the magnitude of the hill sizes of the multi-modality is considered to be higher. Nevertheless, it is expected for sParEGO to improve its convergence to the PF with more function evaluations, since the statistical assessment made about the true robust performance of the solutions that are on the PF is also expected to improve.

For P2, sParEGO approximation to the robust PF is better than ParEGO as shown in Figures 4(b) and 4(d). The convergence of ParEGO deteriorates along the optimization run as shown in Figure 4(d), while the convergence of sParEGO improves. This indicates that the selection criterion used by ParEGO that promotes solutions with a better nominal performance, leads to a deterioration in the convergence towards the true robust solutions. On the other hand, the uncertainty quantification approach used by sParEGO that is used to estimate the true robustness of the solutions, is found to be a better approach in dealing with the task of finding the robust PF, for a given confidence level.

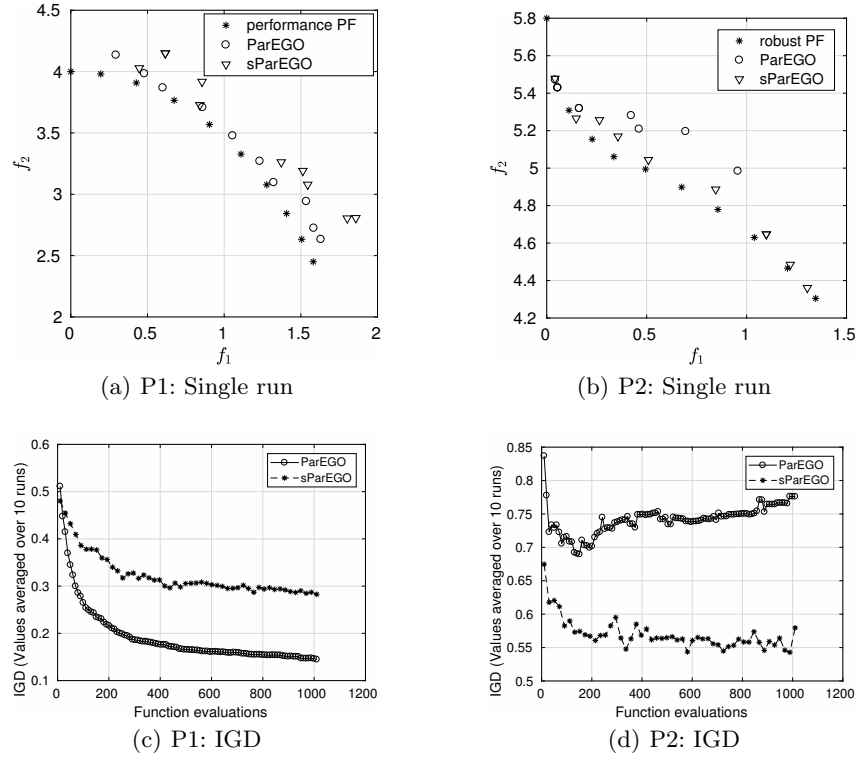


Fig. 4: Comparison of ParEGO and sParEGO for P1 and P2.

## 6 Conclusion

This paper has proposed a new multi-objective optimization algorithm – sParEGO – for expensive uncertain MOPs. The comparative analysis with the existing algorithm ParEGO showed that sParEGO can offer superior performance in approximating a robust PF, but does not perform so well for problems with a highly non-smooth landscape. The latter finding is due to the assumption made by sParEGO that the stochastic performance of neighbouring solutions is similar. Further benchmarking of sParEGO is now needed to confirm its capabilities across a wider set of problem instances.

**Acknowledgements:** This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/L025760/1 as part of the jointly funded Programme for Simulation Innovation.

## References

1. Zhou, A., et al.: Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm Evol Comput* **1**(1) (2011) 32–49
2. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE T Evolut Comput* **10**(1) (2006) 50–66
3. Jones, D., et al.: Efficient global optimization of expensive black-box functions. *J Global Optim* **13**(4) (1998) 455–492
4. Knowles, J., et al.: Noisy multiobjective optimization on a budget of 250 evaluations. In: *Proc. EMO 2009*. (2009) 36–50
5. Beyer, H.G., et al.: Robust optimization - a comprehensive survey. *Comput Method Appl M* **196**(33-34) (2007) 3190–3218
6. Knowles, J., et al.: Multiobjective optimization on a budget of 250 evaluations. In: *Proc. EMO 2005*. (2005) 176–190
7. Ginley, B.M., et al.: Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection. *IEEE T Evolut Comput* **15**(5) (2011) 692–714
8. Huband, S., et al.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE T Evolut Comput* **10**(5) (2006) 477–506
9. Salomon, S., et al.: A Toolkit for Generating Scalable Stochastic Multiobjective Test Problems. In: *Proc. GECCO 2016*. (2016)
10. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T Evolut Comput* **6**(2) (2002) 182–197
11. Giagkiozis, I., et al.: Generalized decomposition and cross entropy methods for many-objective optimization. *Inform Sciences* **282** (2014) 363–387
12. Van Veldhuizen, D., et al.: On measuring multiobjective evolutionary algorithm performance. In: *Proc. CEC 2000*. Volume 1. (2000) 204–211