



MVMOO: Mixed variable multi-objective optimisation

Jamie A. Manson¹ · Thomas W. Chamberlain¹ · Richard A. Bourne¹

Received: 2 September 2020 / Accepted: 25 May 2021 / Published online: 9 July 2021
© The Author(s) 2021

Abstract

In many real-world problems there is often the requirement to optimise multiple conflicting objectives in an efficient manner. In such problems there can be the requirement to optimise a mixture of continuous and discrete variables. Herein, we propose a new multi-objective algorithm capable of optimising both continuous and discrete bounded variables in an efficient manner. The algorithm utilises Gaussian processes as surrogates in combination with a novel distance metric based upon Gower similarity. The MVMOO algorithm was compared to an existing mixed variable implementation of NSGA-II and random sampling for three test problems. MVMOO shows competitive performance on all proposed problems with efficient data acquisition and approximation of the Pareto fronts for the selected test problems.

Keywords Global optimisation · Hypervolume · Multi-objective · Mixed variable · Bayesian optimisation

1 Introduction

As the requirement for optimising process efficiency, whilst simultaneously improving other environmental or economic factors becomes a necessity, utilising optimisation methodologies has become increasingly more important. This has led to optimisation tasks requiring an ever-increasing resource demand to solve. Consequently, efficient and accessible optimisation methodologies are required to guide complex systems to their optimal conditions.

Many real-world optimisation problems consist of multiple conflicting objectives and constraints which can be composed of both continuous and discrete variables. Continuous variables are variables in which we have access to any real value in the desired optimisation range of the variable. Given their inherent continuous nature they are often easier to solve with a wider array of applicable optimisation techniques. Discrete variables can take the form of integer values or categorical values (materials, reaction solvents). In many cases these optimisations can be expensive to evaluate in terms of time or monetary resources.

✉ Richard A. Bourne
R.A.Bourne@leeds.ac.uk

¹ Institute of Process Research and Development, School of Chemistry and School of Chemical and Process Engineering, University of Leeds, Leeds LS2 9JT, UK

It is therefore necessary to utilise algorithms that can efficiently guide the search towards the optimum set of conditions for a given problem. Commonly for real-world problems derivative information may be unavailable, therefore, the use of derivative free optimisation techniques is required.

For many multi-objective problems there does not exist a common optimal for all objectives. As such, algorithms will work towards the trade-off between the competing objectives. This non-dominated set of points is known as the Pareto front [1]. The Pareto front can then be utilised to inform the decision maker as to conditions which act as the best compromise for their given problem and requirements. There are two main techniques for solving multi-objective problems; scalarisation or Pareto based methods. Scalarisation involves converting the multi-objective problem into a problem with a single objective. This can often be done by applying weightings to each objective and additively combining them into a single objective, although there exist many alternative scalarisation techniques [2]. As scalarization combines the objective functions into a single objective, often the methodology needs to be rerun with alternative weightings or weightings changed during the optimisation to develop a complete picture of the Pareto set for a problem. The settings or procedure for weight generation can have a significant effect on the efficacy of the optimisations [3].

Pareto based methods look to iteratively improve a set of non-dominated solutions towards an optimal front. Here, evolutionary algorithms (EAs) have been widely applied for the solution of multi-objective problems [4]. In general, EAs iteratively change a set of candidate solutions towards an optimum set of values. The most notable example of this would be the Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm [5], with the algorithm being successfully applied to a wide array of simulated and real world problems [6, 7]. A key disadvantage for their application on expensive to evaluate problems is the requirement for many function evaluations to progress to an optimum set of points. In these instances, EAs are often coupled with surrogate models of the real process to reduce this burden [8, 9].

Bayesian optimisation approaches have been applied to multi-objective problems. For Bayesian approaches, typically, a Gaussian process surrogate is built from past evaluations, and is utilised in tandem with an acquisition function to determine the next point for the optimisation [10]. The Expected Hypervolume Improvement acquisition function provides an extension of the single objective expected improvement to the multi-objective domain [11]. The algorithm provides good convergence to the true Pareto front, however requires a high computational budget for computing the infill criteria for more than three objectives [12, 13]. Alternative acquisition functions and approaches have been suggested all looking to reduce the computational burden for acquisition evaluation as well as improve the efficiency of the optimisations. Recently, the TSEMO algorithm employing Thompson sampling with an internal NSGA-II optimisation has been proposed [14]. The authors utilise the inherent exploratory nature of Thompson sampling to build Gaussian surrogates, which are subsequently optimised utilising NSGA-II to suggest points that maximise the hypervolume improvement.

Generally, mixed variable multi-objective problems have received less attention when compared with other optimisation fields. However, a significant volume of research has made use of EAs such as genetic algorithms (GAs) to solve a wide variety of real-world problems, often being coupled with simulated systems or surrogate models [15–20]. In many of these works the direct system is rarely optimised with preliminary work instead used to construct accurate surrogate models. Surrogate-based techniques utilise a cheap model constructed from a training dataset generated from the expensive objective

function. For each iteration, the surrogate is used to guide the optimisation and determine the next point of evaluation. Gaussian processes (GPs) have often shown excellent performance in these surrogate based techniques; however, their use has usually been limited to continuous variable only processes. This limitation is linked with the correlation function requiring a distance metric which can be difficult to define between different levels of discrete variables. Previous methods [21–24] propose modifications to the covariance function to ensure a valid covariance matrix can be defined for mixed inputs.

More recently, Zhang et al. [25] adopted a latent variable approach to enable the Bayesian optimisation for material design. The proposed methodology maps the qualitative variables to underlying quantitative latent variables, from which a standard covariance function can be used. The latent variables for each discrete variable combination are determined utilising maximum likelihood estimation with each combination being mapped to a 2D latent variable domain. The authors compared their suggested single objective optimisation methodology to MATLAB's *bayesopt* which utilises dummy variables in place of the qualitative variables.

Herein we present a mixed variable multi-objective Bayesian optimisation algorithm and provide three test cases with comparison to both random sampling and a mixed variable version of NSGA-II provided in jMetalPy [26]. The algorithm looks to extend Bayesian multi-objective methodologies to the mixed variable domain, which to the best of the authors' knowledge has limited prior work [27]. Utilising the recently proposed Expected Improvement Matrix [28] and an adapted distance metric, the algorithm provides an efficient approach to optimising expensive to evaluate mixed variable multi-objective optimisation problems without the need for reparameterization of the discrete variables.

2 Overview

2.1 Gaussian processes

A Gaussian process can be defined as a collection of random variables. A GP is fully defined by a mean function, $m(\mathbf{x})$, and a covariance function, $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ [29].

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \quad (1)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \quad (2)$$

$$f(x) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3)$$

With \mathbf{x} being an arbitrary input vector and $\mathbb{E}[\cdot]$ the expectation. The mean function defines the average of the function, with the covariance function or kernel specifying the covariance or the relationship between pairs of random variables. The kernel is composed of several hyperparameters for which the optimal values can be determined through model training. The application of GP models in optimisation literature is principally restricted to the continuous domain utilising a stationary kernel. When considering mixed variable optimisation, one must ensure that a valid kernel can be constructed.

2.2 Covariance function

The application of Bayesian techniques is often limited to continuous variable optimisation problems with the covariance being a function of the distance between points in the input domain.

The difficulty for mixed variable optimisation problems when using a GP as the surrogate for the response is formulating a proper correlation structure. As discrete variables can vary in similarity by a large degree, assuming continuous methods can be applied to determine the covariance matrix is not appropriate and may result in an invalid covariance matrix. The covariance function that characterises a GP is customisable by the user, however, to ensure that it is valid it is a requirement for the covariance function to be symmetric and positive semi-definite; this will be true if its eigenvalues are all non-negative.

Qian et al. [21] were amongst the first to implement GP modelling for systems with mixed variable types. The method requires an extensive estimation procedure, requiring the solution of an internal semidefinite programming problem to ensure a valid covariance matrix is computed during model training. Due to the large number of hyperparameters required, it can be difficult to obtain their optimum value, especially in efficient optimisation application, where the number of evaluations is often restricted.

Halstrup [30] developed a novel covariance function based upon the Gower distance metric. This metric requires significantly fewer hyperparameters and exhibits good scalability to higher dimensional problems. Gower [31] initially proposed the metric to measure the similarity between two objects with mixed variable data, with the metric modified to represent distance.

$$r_{gow}(\mathbf{w}^i, \mathbf{w}^j) = \frac{\sum_{k=1}^{k=q} \frac{|w_k^i - w_k^j|}{\Delta w_k}}{d + q} + \frac{\sum_{m=1}^{m=d} s_{w_m^i w_m^j}}{d + q} \quad (4)$$

The first term considers the quantitative factors and is the weighted Manhattan distance between the two variables, where Δw_k is the range of the k^{th} quantitative variable $s_{w_m^i w_m^j}$ considers the m^{th} qualitative variable. This is set to 0 if w_m^i is equal to w_m^j and 1 if they are not equal. d and q are the number of qualitative and quantitative variables, respectively.

An applied example for Gower distance calculation for a sample set of experiments is provided below. Consider we have three variables in which two are quantitative and one is qualitative, we have an initial dataset (Table 1) and we want to determine the overall distance between each entry, such that we may be able to model a response.

For each quantitative variable we can initially calculate the variable range normalised Manhattan distance and for the single qualitative variable we can perform a binary comparison between each entry. Finally, we combine each distance matrix using (4), the results of which are displayed in Fig. 1.

Table 1 Example distance calculation input domain for two quantitative variables (temperature and residence time) and one qualitative (solvent)

Index	Temperature (°C)	Solvent	Residence time (min)
1	30	MeCN	1
2	100	DMF	5
3	50	EtOH	8
4	70	DMSO	2

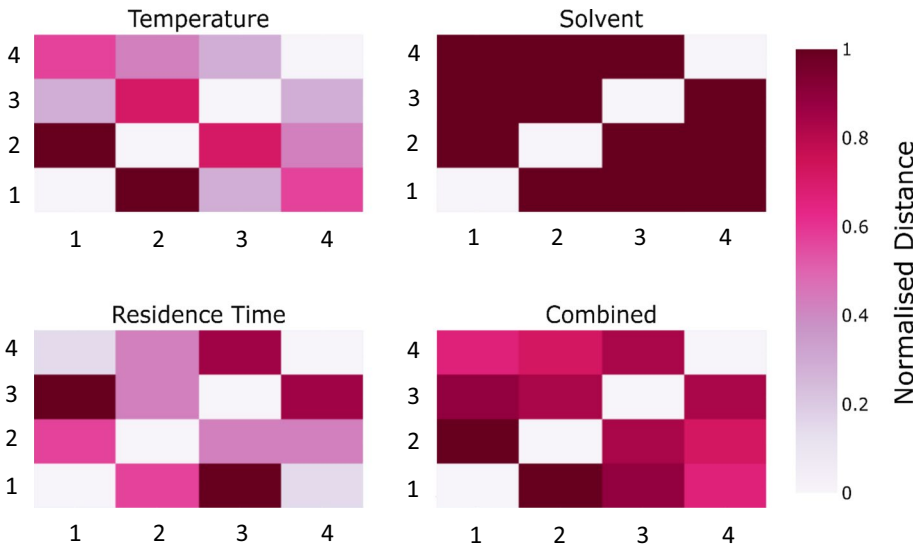


Fig. 1 Example distance calculation. Individual distance matrices are initially calculated and then combined

With a valid distance metric this permits the use of pre-existing correlation functions such as the square exponential or the Matérn kernels. An example of a mixed Matérn 5/2 kernel is given below:

$$\begin{aligned}
 K^{GMat\frac{5}{2}}(\mathbf{w}^i, \mathbf{w}^j | \boldsymbol{\theta}, \sigma) = & \sigma^2 \prod_{n=1}^{dim} \left(1 + \sqrt{5} \frac{dim}{\theta_n} \cdot r_{gow}(\mathbf{w}_n^i, \mathbf{w}_n^j) + \frac{5}{3} \left(\frac{dim}{\theta_n} \cdot r_{gow}(\mathbf{w}_n^i, \mathbf{w}_n^j) \right)^2 \right) \\
 & \exp \left(-\sqrt{5} \frac{dim}{\theta_n} \cdot r_{gow}(\mathbf{w}_n^i, \mathbf{w}_n^j) \right)
 \end{aligned}
 \tag{5}$$

where *dim* is the problem dimension, with $\boldsymbol{\theta}$ and σ the covariance function’s hyperparameters.

Prior work by Halstrup has looked at the application of the Gower distance metric for use in enabling Bayesian single objective optimisation techniques to be used on mixed variable systems. This work was further developed by Pelamatti et al. [32] in which they compared the currently available techniques for optimising mixed variable systems utilising Gaussian approaches. They found the suggested methodology adopted by Halstrup performed comparatively well to other techniques whilst offering the benefit of a reduced number of hyperparameters.

2.3 Model training

For effective use and application of Bayesian techniques it is recommended to optimise the hyperparameters of the GP model prior to application of the acquisition function. The hyperparameters in this instance were tuned through maximising the likelihood of the available data points. The Python package GPflow (version 1.3) [33] was utilised alongside a modified version of the Matérn kernel to construct and fit the GP model to the data for all test examples

investigated. When utilising GPflow this is achieved through maximising the log marginal likelihood:

$$\log p(y|X, \theta) = -\frac{1}{2}y^T K_y^{-1}y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi \tag{6}$$

where K_y is the covariance matrix of noisy response values, $-\frac{1}{2}y^T K_y^{-1}y$ assesses how well the model fits the data given the current hyperparameters, θ . A complexity penalty in the form of $-\frac{1}{2} \log |K_y|$ is included to reduce overfitting of the model. $-\frac{n}{2} \log 2\pi$ is a normalisation constant, where n is the number of training data points.

2.4 Acquisition function

To select the next evaluation point Bayesian optimisation utilises an acquisition function which is computed using the surrogate GP model built upon previously collected data. The acquisition function is typically inexpensive so that it can be evaluated extensively to guide the proceeding data acquisition.

One such acquisition function is the expected improvement matrix (EIM) [28]. EIM is a multi-objective adaptation of EI, with the expected improvement for each candidate and each objective calculated in a matrix.

$$EI_i^j(\mathbf{x}) = \left(f_i^j - \hat{y}_i(\mathbf{x})\right) \Phi\left(\frac{f_i^j - \hat{y}_i(\mathbf{x})}{s_i(\mathbf{x})}\right) + s_i(\mathbf{x})\phi\left(\frac{f_i^j - \hat{y}_i(\mathbf{x})}{s_i(\mathbf{x})}\right) \tag{7}$$

$$\begin{bmatrix} EI_1^1(\mathbf{x}) & \dots & EI_m^1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ EI_1^k(\mathbf{x}) & \dots & EI_m^k(\mathbf{x}) \end{bmatrix} \tag{8}$$

f_i^j are the current best values along the Pareto front with k the number of points in the best solution and m the number of objectives, $\hat{y}_i(\mathbf{x})$ is the predicted mean of the model, with $s_i(\mathbf{x})$ the standard deviation of the prediction. $\Phi(\mathbf{x})$ and $\phi(\mathbf{x})$ are the Gaussian cumulative distribution and probability density functions, respectively.

This matrix representation of expected improvement resembles the current best solution in multi-objective optimisation, which is also a matrix.

$$\begin{bmatrix} f_1^1(\mathbf{x}) & \dots & f_m^1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ f_1^k(\mathbf{x}) & \dots & f_m^k(\mathbf{x}) \end{bmatrix} \tag{9}$$

The matrix representations of EI for each objective are then combined for each point using three possible transformations; Euclidean, min/max distance and hypervolume based transformations. In this work we have elected to use a Euclidean based transformation to give a combined expected improvement for a given point with respect to all the objectives and the current non-dominated front:

$$EIM_e(\mathbf{x}) = \min_{j=1, \dots, k} \sqrt{\sum_{i=1}^m \left(EI_i^j(\mathbf{x})\right)^2} \tag{10}$$

As the EI equation is in its closed form it can be rapidly calculated and optimised utilising evolutionary algorithms or multi-start local search methods. This should allow a global solution for the optimisation of EIM to be found at a reduced computational cost. EIM has been previously implemented in the form of the EIM-EGO algorithm. The implementation was tested on a series of continuous test problems, displaying competitive results to the state-of-the-art multi-objective algorithms, whilst providing efficient scaling when increasing the number of optimised objectives [28].

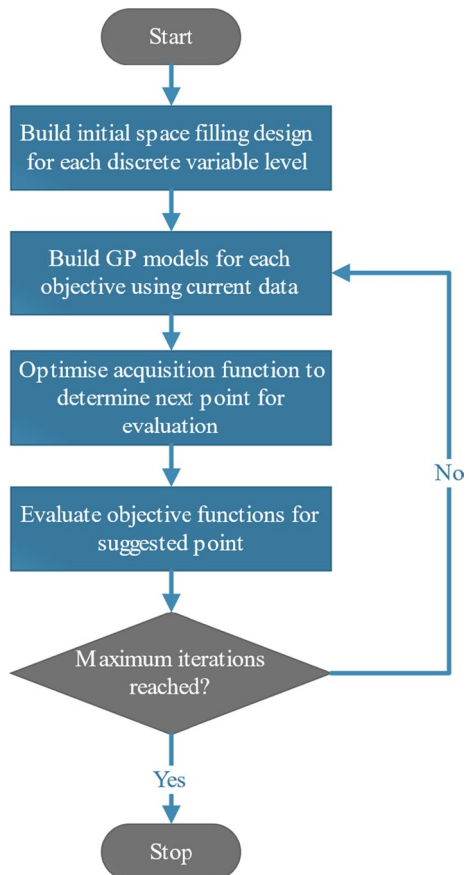
2.5 Algorithm overview

Figure 2 provides an overview of the optimisation process for the MVMOO algorithm.

For the initial space filling design in all cases, we performed a 5 sample Latin hypercube for each combination of the discrete variables. It is noted that this can lead to a large initial cost to the optimisations when there are multiple discrete variables or discrete variable levels in the optimisation problem.

Upon completion of the initial data collection a GP model was constructed for each objective function and hyperparameters optimised using GPflow’s internal Adam optimiser. The trained GP models are then used with the EIM acquisition function to determine

Fig. 2 Flowchart overview for MVMOO



the next point for evaluation. The optimisation of the acquisition function occurs in two stages; initially a large sample of the acquisition function is taken using a Halton sequence for each discrete variable combination. The leading variable combination was then taken forward with additional local optimisation performed using SciPy's [34] implementation of sequential least squares programming (SLSQP) [35].

2.6 Test functions

The algorithm has been applied to three test problems [36–38] to evaluate the selected approach and provide comparison with existing mixed variable multi-objective optimisation methods.

For the first two test problems, the true Pareto front optimal set was known and has been detailed below. For the final test problem, the true Pareto front has been determined through comprehensive evaluation of the function domain utilising the mixed variable NSGA-II algorithm.

Tables 2 and 3 provide details of the initial dataset size, algorithm evaluation budgets and settings for NSGA-II for all test problems. NSGA-II termination criteria was set to be once the prespecified number of generations had been evaluated.

Table 2 Algorithm budgets for test problems

	Discrete VLMOP2	Ordinary differential equation catalytic system	Fuel injector design
MVMOO initial dataset size	10	40	20
MVMOO, LHC and random function evaluation budget	40	125	100
NSGA-II generations	20	85	80
NSGA-II population size	40	125	100

Table 3 NSGA-II settings, where n is the problem input dimension. Recommended settings from the original paper were utilised for all test problems [5]

Setting	Operator	Probability	Distribution index
Selection function	Binary tournament		
Recombination function	SBX crossover Integer SBX crossover	0.9	20
Mutation function	Polynomial mutation Integer polynomial mutation	$\frac{1}{n}$	20

2.7 Discrete VLMOP2

The VLMOP2 test function [36] is a well-known bi-objective test problem. Here we have adapted the original problem to create a mixed variable equivalent:

$$\begin{aligned} & \min (f_1(\mathbf{x}, d), f_2(\mathbf{x}, d)) \\ & x_i \in [-2, 2] \quad d \in [a, b] \\ f_1(\mathbf{x}, d) &= \begin{cases} 1 - \exp\left(-\sum_{i=1, \dots, n} \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) & \text{if } d = a \\ 1.25 - \exp\left(-\sum_{i=1, \dots, n} \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) & \text{if } d = b \end{cases} \\ f_2(\mathbf{x}, d) &= \begin{cases} 1 - \exp\left(-\sum_{i=1, \dots, n} \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) & \text{if } d = a \\ 0.75 - \exp\left(-\sum_{i=1, \dots, n} \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) & \text{if } d = b \end{cases} \end{aligned} \quad (11)$$

Like the original function, the Pareto front is concave with the Pareto optima line generally from $x_i = -\frac{1}{\sqrt{n}}$ to $x_i = \frac{1}{\sqrt{n}}$.

Figure 3 details the Pareto front and Pareto optimal set for the discrete VLMOP2 problem. This provides a useful initial test for the algorithm to determine whether it can effectively switch between the two levels of the discrete variable.

2.8 Ordinary differential equation (ODE) catalytic system

The following test problem involved the optimisation of 3 continuous variables and a single qualitative variable for two objective functions obtained through the solution of a system of ordinary differential equations (ODEs). Solution of the ODE system can be obtained through application of an initial value problem algorithm, such as the Runge–Kutta methods [39]. The system of ODEs was used to describe a chemical reaction where the qualitative variable is the choice of catalyst utilised in system and varies the activation energy of the reaction as described in (12), (13) and Table 4.

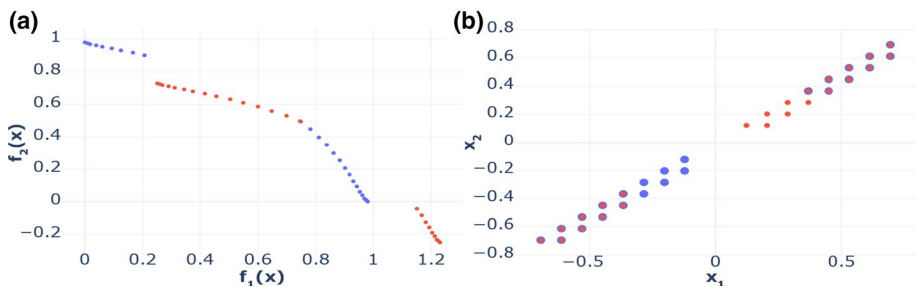


Fig. 3 Pareto front (a) and Pareto optimal set (b) for VLMOP2 test problem. Points where $d=a$ are shown in blue and points where $d=b$ are shown in red. Link to interactive plots are given as follows: **a** <https://chart-studio.plotly.com/~jmanson377/188>, **b** <https://chart-studio.plotly.com/~jmanson377/191>. (Color figure online)

Table 4 Catalytic effect on reaction activation energy

Catalyst	E_{Ai} (kJ mol ⁻¹)
1 (T < 80 °C)	-5.0
1 (T > 80 °C)	-5.0 + 0.3 (T - 80)
2	0.7
3	0.7
4	0.7
5	0.7
6	2.2
7	3.8
8	7.3

This problem was initially reported as a single objective mixed variable problem by Baumgartner et al. [37]; we have modified the problem to formulate a multi-objective example for this test case. The system involves optimising the catalyst, d , catalyst concentration, C_{cat} , temperature, T , and the reaction time, t_{res} , in which the catalyst effects the activation energy of the reaction, E_{Ai} . Equations describing the system are provided below.

$$\begin{aligned}
 & \max (Yield, STY) \\
 & C_{cat} \in [0.835 \text{ mM}, 4.175 \text{ mM}] \quad T \in [30 \text{ }^\circ\text{C}, 110 \text{ }^\circ\text{C}] \quad t_{res} \in [1 \text{ min}, 10 \text{ min}] \\
 & d \in \{1, 2, 3, 4, 5, 6, 7, 8\} \\
 & A + B \xrightarrow{k} P \\
 & k = C_{cat}^{\frac{1}{2}} k_0 e^{-\frac{E_{Ar} + E_{Ai}}{RT}} \quad (12) \\
 & \frac{dA}{d\tau} = -k[A][B] \\
 & \frac{dB}{d\tau} = -k[A][B] \\
 & \frac{dP}{d\tau} = k[A][B]
 \end{aligned}$$

where τ is the residence time of the reactor, k_0 is the Arrhenius constant for the reaction ($3.1 \times 10^7 \text{ L}^{\frac{1}{2}} \text{ mol}^{-\frac{3}{2}} \text{ s}^{-1} \times 10^7 \text{ L}^{\frac{1}{2}} \text{ mol}^{-\frac{3}{2}} \text{ s}^{-1}$), $[A]$ and $[B]$ are the concentrations of A and B in the reactor (Initial values: $[A]_0 = 0.167 \text{ M}$ and $[B]_0 = 0.250 \text{ M}$) and E_{Ar} is the reaction activation energy (55 kJ mol^{-1}).

$$\begin{aligned}
 Yield &= \frac{[P]_{final}}{[A]_{initial}} \\
 STY &= \frac{100 \times [P]_{final}}{t_{res}} \quad (13)
 \end{aligned}$$

The system of ODEs was solved utilising MATLAB's ode45 numerical integration solver with the algorithm optimising for both space time yield and product yield.

Figure 4 details the Pareto front for the simulated systems with details of the Pareto optimal set provided.

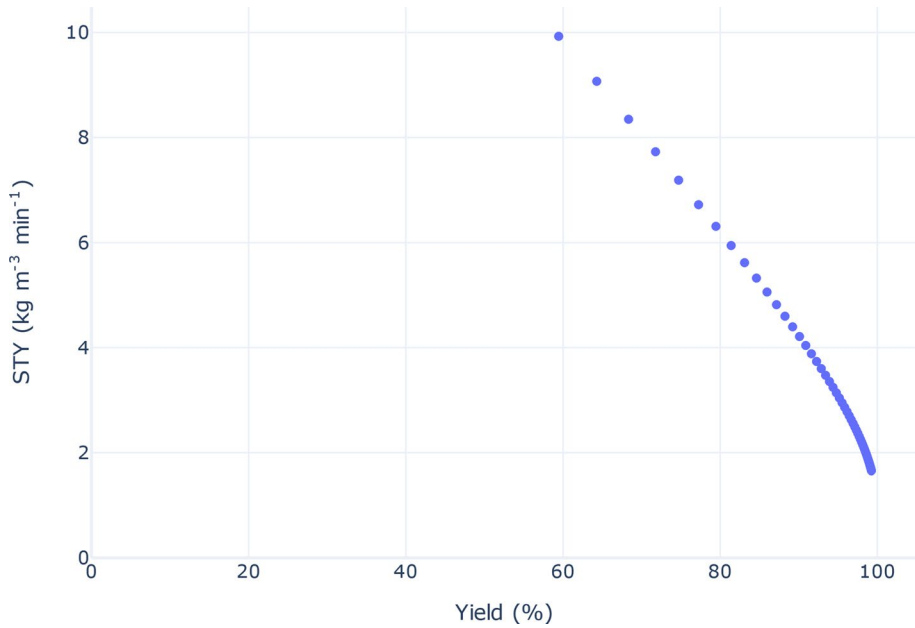


Fig. 4 Pareto front for catalytic ODE example. Pareto set at the following conditions: $C_{cat} = 4.175$ mM, $T = 80$ °C, $t_{res} \in [1 \text{ min}, 10 \text{ min}]$ and $d = 1$. Link to interactive plot: <https://plotly.com/~jmanson377/193/>

As the Pareto front in this test problem is comprised solely of a single discrete variable level, this allows for comparison with a continuous multi-objective optimisation algorithm. In this instance comparison versus the Thompson Sampling Efficient Multi-Objective Optimisation (TSEMO) [14] algorithm was performed.

2.9 Fuel injector design

This problem is comprised of a mixed integer fuel injector design problem proposed by Burachik et al. [38]. The authors adjusted the problem to impose an integer constraint on one of the design variables. The problem is described as follows (Fig. 5):

$$\begin{aligned}
& \min (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x})) \\
& x_{2-4} \in [-2, 2] \bar{x}_1 \in \{0, 1, 2, 3\} \\
& x_1 = 0.2\bar{x}_1 \\
& f_1(\mathbf{x}) = 0.692 + 0.4771x_1 - 0.687x_4 - 0.08x_3 - 0.065x_2 - 0.167x_1^2 \\
& \quad - 0.0129x_1x_4 + 0.0796x_4^2 - 0.0634x_1x_3 - 0.0257x_3x_4 \\
& \quad + 0.0877x_3^2 - 0.0521x_1x_2 + 0.00156x_2x_4 + 0.00198x_2x_3 \\
& \quad + 0.0184x_2^2 \\
& f_2(\mathbf{x}) = 0.37 - 0.205x_1 + 0.0307x_4 + 0.108x_3 + 1.019x_2 - 0.135x_1^2 \\
& \quad + 0.0141x_1x_4 + 0.0998x_4^2 + 0.208x_1x_3 - 0.0301x_3x_4 \\
& \quad - 0.226x_3^2 + 0.353x_1x_2 - 0.0497x_2x_3 - 0.423x_2^2 \\
& \quad + 0.202x_1^2x_4 - 0.281x_1^2x_3 - 0.342x_1x_4^2 - 0.245x_3x_4^2 \\
& \quad + 0.281x_3^2x_4 - 0.184x_1x_2^2 + 0.281x_1x_3x_4 \\
& f_3(\mathbf{x}) = 0.153 - 0.322x_1 + 0.396x_4 + 0.424x_3 + 0.0226x_2 + 0.175x_1^2 \\
& \quad + 0.0185x_1x_4 - 0.0701x_4^2 - 0.251x_1x_3 + 0.179x_3x_4 \\
& \quad + 0.015x_3^2 + 0.0134x_1x_2 + 0.0296x_2x_4 + 0.0752x_2x_3 \\
& \quad + 0.0192x_2^2 \\
& f_4(\mathbf{x}) = 0.758 + 0.358x_1 - 0.807x_4 + 0.0925x_3 - 0.0468x_2 - 0.172x_1^2 \\
& \quad + 0.0106x_1x_4 + 0.0697x_4^2 - 0.146x_1x_3 - 0.0416x_3x_4 \\
& \quad + 0.102x_3^2 - 0.0694x_1x_2 - 0.00503x_2x_4 + 0.0151x_2x_3 \\
& \quad + 0.0173x_2^2
\end{aligned} \tag{14}$$

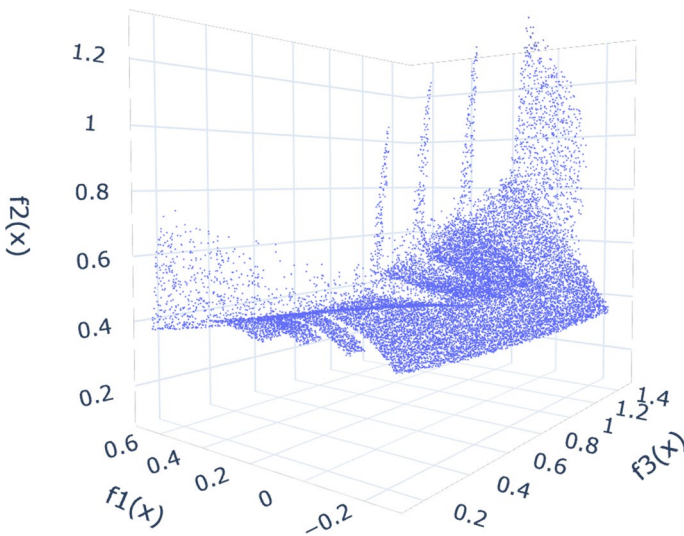


Fig. 5 Fuel Injector Pareto front displayed for the first three objective functions. Link to interactive plot: <https://chart-studio.plotly.com/~jmanson377/195>

As with the discrete VLMOP2 example, the Pareto front is made up of multiple discrete variable levels so should provide a good test problem to determine how effectively the algorithm can switch between different discrete variable combinations in a larger problem.

3 Results

For the three test cases presented the algorithm was compared against random sampling, a split Latin hypercube (LHC) design and the mixed variable NSGA-II.

3.1 Performance indicators

There exists a wide plethora of performance metrics that have been developed over the last few decades for the comparison of different multi-objective optimisation algorithms. Riquelme et al. [40] provide a comprehensive review of many performance metrics highlighting their usage and the associated advantages/disadvantages. To evaluate the performance of the newly proposed algorithm we have utilised three performance metrics. A summary for each of the chosen metrics is provided below.

3.2 Hypervolume

The Hypervolume performance metric measures the volume of the objective space dominated by the calculated Pareto front given a specified reference point. The indicator provides a measure for both convergence and diversity and has been widely used as a comparison tool for multi-objective algorithms [41].

3.3 Modified inverted generational distance (IGD+)

The modified Inverted Generational Distance (IGD+) performance metric was first proposed by Ishibuchi et al. [42]. Modification of the IGD performance metric [43] was performed to achieve a weakly Pareto compliant version. The metric provides a measure on the convergence towards the true Pareto front as well as the diversity of the calculated front, with a smaller value suggesting the calculated Pareto front is close to that of the true Pareto front. IGD+ is calculated as follows:

$$IGD^+(A) = \frac{1}{|Z|} \left(\sum_{i=1}^{|Z|} d_i^{+2} \right)^{\frac{1}{2}} \quad (15)$$

where $d_i^+ = \max \{a_i - z_i, 0\}$, with a_i being a point from the calculated Pareto front A and z_i being a point from the true Pareto front Z .

3.4 Worst Attainment Surface

The Worst Attainment surfaces are defined by a set of Pareto points for a specific algorithm and test function that are dominated by all other points for the specific algorithm

on a specific test function. They highlight the divide between dominated and non-dominated points for the given set [44]. This provides insight as to where the different algorithms perform differently and to what extent and thus can highlight a specific algorithm's weakness.

3.5 Discrete VLMOP2

Firstly, we will consider the results of the optimisation of the discrete VLMOP2 test problem. Summary plots for all performance metrics, as well as iterative summary plots, are shown in Fig. 6. For the hypervolume performance indicator a reference point of $R = [1.0, 1.25]$ was used.

Overall, the MVMOO algorithm performs comparatively well compared to the mixed version of the NSGA-II algorithm with the novel MVMOO algorithm exceeding the performance in terms of both Hypervolume and IGD+, with a 21-fold reduction in iterations for Hypervolume and a 36-fold reduction for IGD+ for comparative performance to NSGA-II.

The iterative results for both Hypervolume and IGD+ (Fig. 6a, b) appear to indicate premature termination of both MVMOO and NSGA-II with MVMOO plateauing to a lesser extent than NSGA-II. The budget for this test problem was kept deliberately small due to only two variables being optimised and the desire for efficient multi-objective optimisation.

Figure 6(e) illustrates well how in the MVMOO algorithm was able to achieve points at the extremes of the Pareto front and correlate well for the two outlying points for both Hypervolume and IGD+. Here the algorithm has failed to correctly obtain the mixed central aspect of the Pareto front (see Fig. 6(e)) suggesting the algorithm may at times struggle to switch between discrete variables where the effects of the continuous variables are similar, this behaviour is further explored in the fuel injector design test problem. Again, it is likely this effect, combined with the limited evaluation budget, has likely led to the two outlying points in both IGD+ and Hypervolume. In this case where the evaluation budget is limited the initial training data collection plays a key role in the efficacy of the algorithm.

3.6 ODE Catalytic System

Summary plots for all performance metrics, as well as iterative summary plots for the ODE catalytic system test problem are shown in Fig. 7. For the hypervolume performance indicator a reference point of $R = [0, 0]$ was used.

Comparative performance can be observed in Fig. 7(a–d) for the MVMOO algorithm compared with a solely continuous multi-objective optimisation counterpart. Figure 7(e) details how the algorithm in the worse instance struggles to achieve the points at the extremes of the Pareto front, possibly accounting for the lower values of Hypervolume observed. NSGA-II was able to achieve superior performance across all metrics with extremely good approximation of the Pareto front in the worst attainment summary surface plot, Fig. 7(e).

Considering both Fig. 7(a, b), we can see there is a significant stall in the iterative progress for both performance metrics. This behaviour is linked to the current initial data gathering procedure for the MVMOO algorithm with a 5 sample Latin hypercube taken for each level of the discrete variable. As the system only has a single optimum discrete variable level this leads to this stalling behaviour. Further work investigating improved techniques

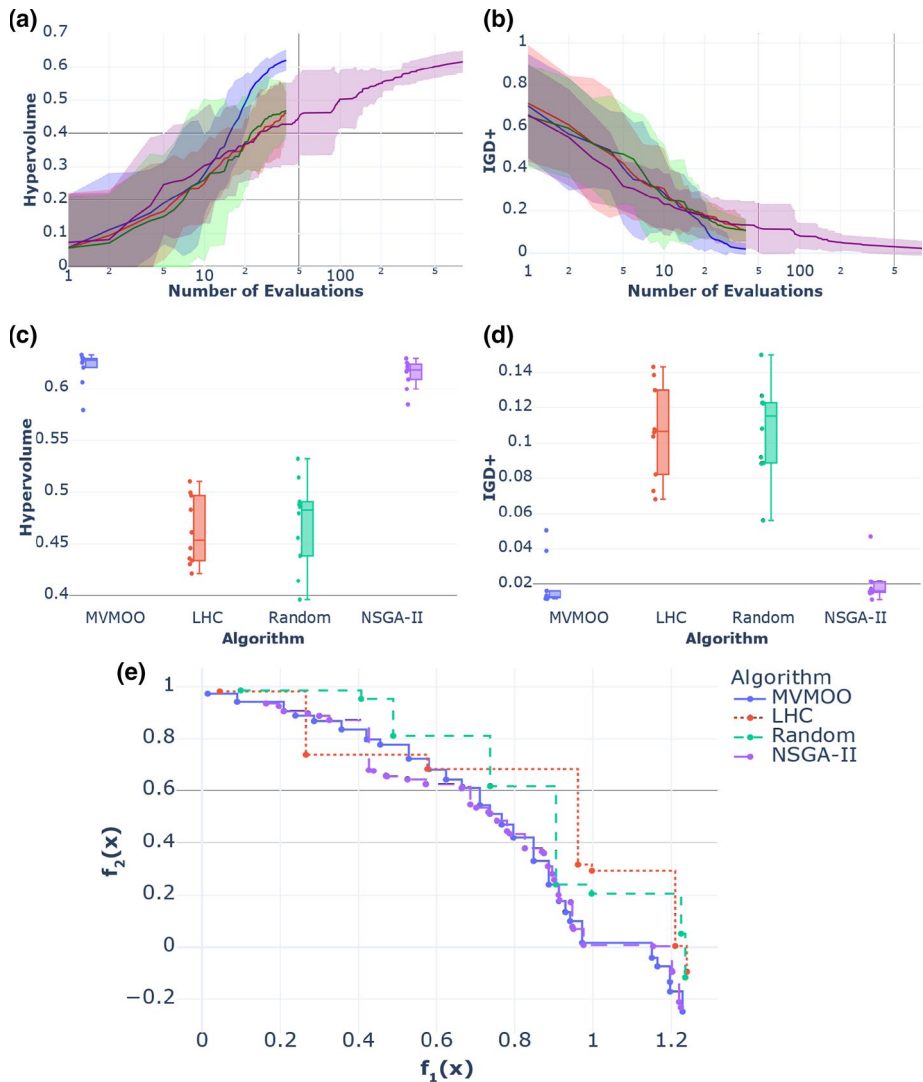


Fig. 6 Optimisation results for discrete VLMOP2 test problem for 10 runs. Algorithms are shown as follows: — MVMOO, — LHC, — Random and — NSGA-II. For **a** and **b** 95% confidence interval (CI) for each algorithm indicated by the shaded areas. For **a**, **b** NSGA-II was afforded a total of 800 function evaluations per run (20 generations with a population size of 40). Interactive plots for all subfigures are given as follows: **a** <https://plotly.com/~jmanon377/207/>, **b** <https://plotly.com/~jmanon377/218/>, **c** <https://plotly.com/~jmanon377/139/>, **d** <https://plotly.com/~jmanon377/135/> and **e** <https://plotly.com/~jmanon377/137/>. (Color figure online)

for initial data collection may prove beneficial for systems when there is a single optimal discrete variable level. The steepness of the gradient of improvement for Hypervolume in Fig. 7(a) indicates how the MVMOO algorithm efficiently works towards the Pareto front for the system with final results comparable to that of the continuous algorithm which was afforded a larger algorithm iteration budget. This highlights the efficiency of the algorithm

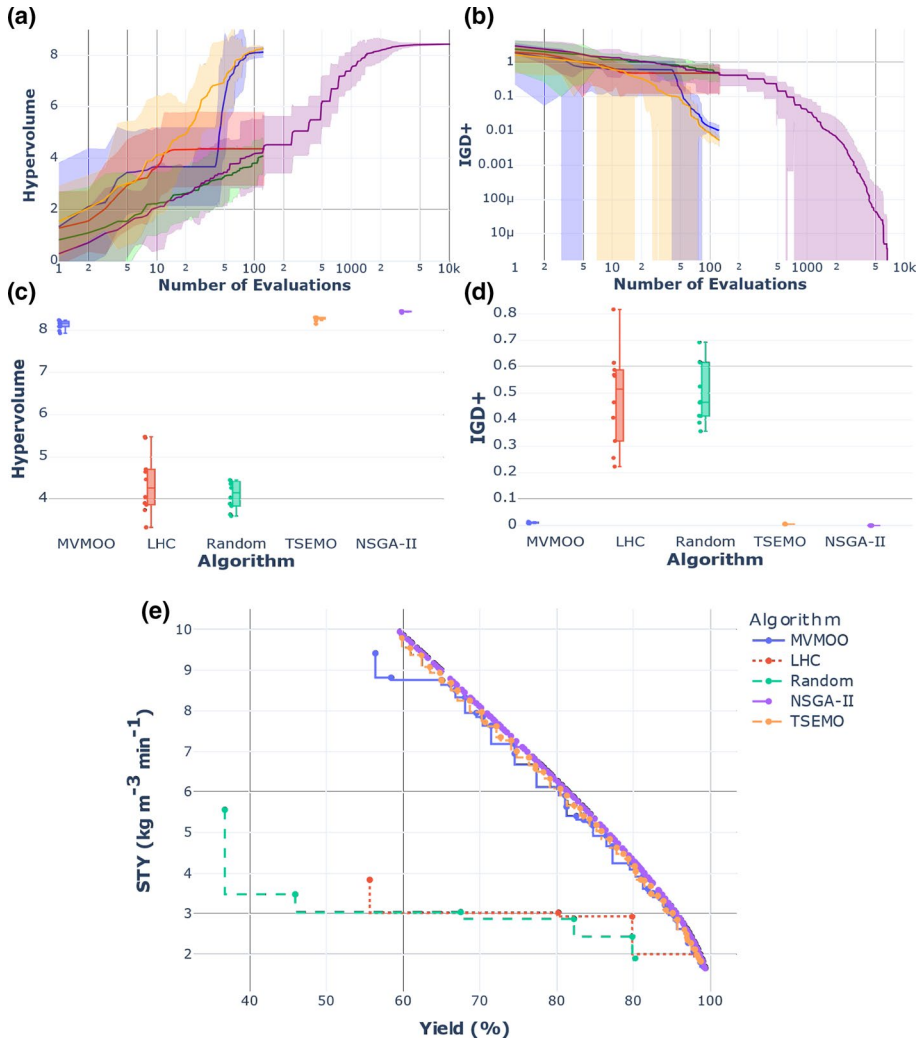


Fig. 7 Optimisation results for ODE catalytic system test problem for 10 runs. Algorithms are shown as follows: — MVMOO, — LHC, — Random, — TSEMO and — NSGA-II. For **a** and **b** 95% confidence interval (CI) for each algorithm indicated by the shaded areas. NSGA-II was afforded a total of 10,200 function evaluations per run (85 generations with a population size of 120). Interactive plots for all subfigures are given as follows: **a** <https://plotly.com/~jmanson377/224/>, **b** <https://plotly.com/~jmanson377/226/>, **c** <https://plotly.com/~jmanson377/146/>, **d** <https://plotly.com/~jmanson377/148/> and **e** <https://plotly.com/~jmanson377/153/>. N.B. The y axis for **b** is plotted using a log scale to allow discernment between algorithms. The large jumps in CI are where the CI tends to zero. (Color figure online)

particularly given the search domain was 8 times larger than that of the continuous algorithm when you consider the discrete variable levels. Comparing performance with regards to Hypervolume and IGD+ against NSGA-II the algorithm achieves the same level of performance in approximately 7% and 8% of the iterations to that required by NSGA-II.

3.7 Fuel injector design

Summary plots for all performance metrics, as well as iterative summary plots for the fuel injector design test problem are shown in Figs. 8, 9 and 10. For the hypervolume performance indicator a reference point of $R = [0.8, 1.4, 1.7, 1.0]$ was used.

Examining Fig. 8(a–d) the results for both performance metrics indicate a significant improvement when using MVMOO over NSGA-II, with the algorithm achieving on par performance to the final mean hypervolume of NSGA-II after 55 iterations and after 42 iterations for mean IGD+ representing a 145-fold and a 190-fold improvement in efficiency with respect to these performance indicators.

A notable difference between the worst attainment surfaces of MVMOO and NSGA-II is the approximation of the bulk body of the Pareto front, with MVMOO providing a more uniform spread of points with some clustering present in NSGA-II. As with the ODE catalytic system example the MVMOO struggles to achieve the extremes of the Pareto front. This, however, may be rectified if the algorithm was allowed to continue with a lack of plateauing in performance metrics evident from the iterative Hypervolume plot (Fig. 8a) suggesting an improvement may be achieved through increasing the budget, this behaviour is echoed to a lesser extent on the iterative plot for IGD+ (Fig. 8b). The greater degree in plateau evident on the IGD+ could be linked to the algorithm struggling to achieve points at the edge of the Pareto front. In both the Hypervolume and IGD+ iterative plots there is clear evidence that the NSGA-II algorithm

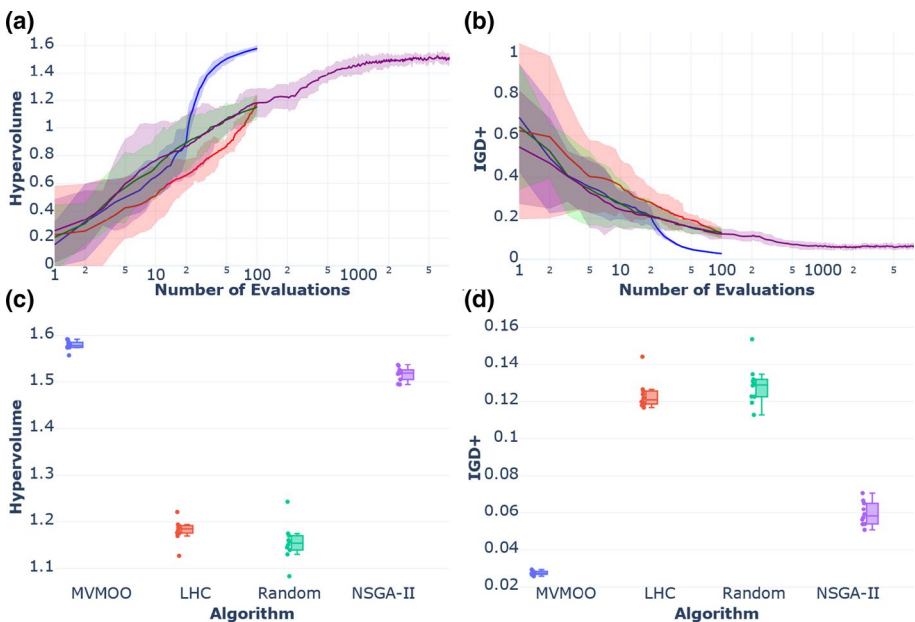


Fig. 8 Optimisation results for fuel injector design test problem for 10 runs. Algorithms are shown as follows: — MVMOO, — LHC, — Random and — NSGA-II. For **a** and **b** 95% confidence interval (CI) for each algorithm indicated by the shaded areas. NSGA-II was afforded a total of 8000 function evaluations per run (80 generations with a population size of 100). Interactive plots for all subfigures are given as follows: **a** <https://plotly.com/~jmanson377/234/>, **b** <https://plotly.com/~jmanson377/237/>, **c** <https://plotly.com/~jmanson377/165/>, **d** <https://plotly.com/~jmanson377/167/>. (Color figure online)

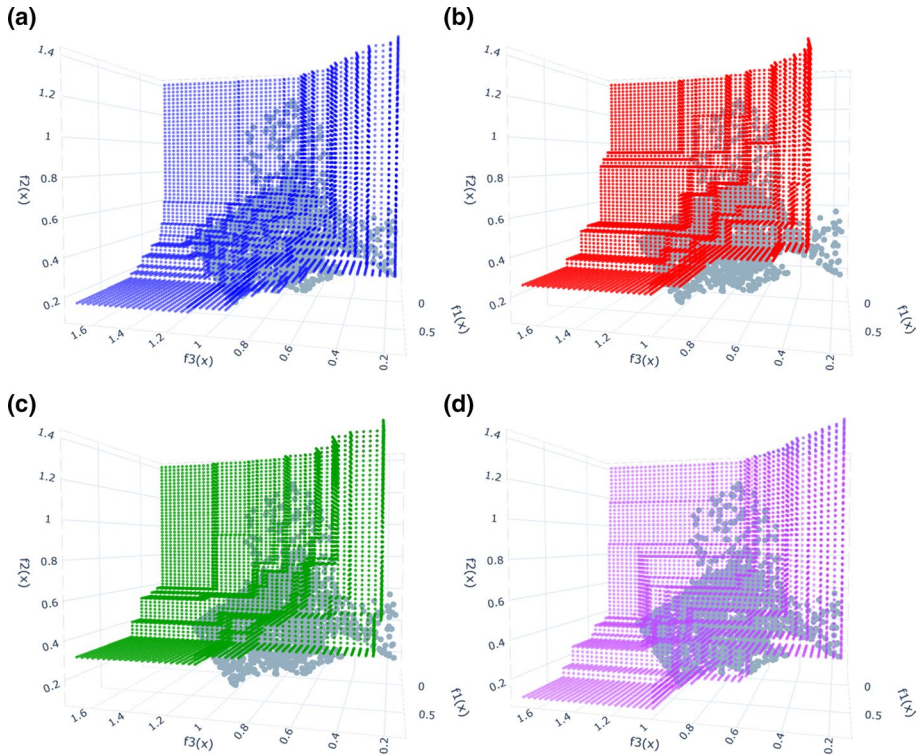


Fig. 9 Worst attainment summary surface for fuel injector design test problem. Reference surface shown as grey points, with the plots corresponding to the following algorithms: **a** MVMOO, **b** LHC, **c** Random and **d** NSGA-II. Link to interactive plot: <https://chart-studio.plotly.com/~jmanson377/201>

struggles with this 4-objective system with little improvement apparent for the latter iterations. Observing Figs. 9 and 10, this marked difference in both Hypervolume and IGD+ for NSGA-II could be attributed to the lack of points around the bulk sections of the Pareto front. Additional iterations would likely rectify this; however, this highlights the drastically improved efficiency in attaining near Pareto optimal points presented by the MVMOO algorithm.

Overall, the MVMOO displays competitive performance in term of both efficiency and final performance across all three test problems presented. On all examples the algorithm displays excellent repeatability evident through the narrow quartile ranges on all test problem box plots. The algorithm has been shown to be effective at switching between discrete variable levels should this be required and produces a uniform distribution of points across the front efficiently describing the front for a minimal budget.

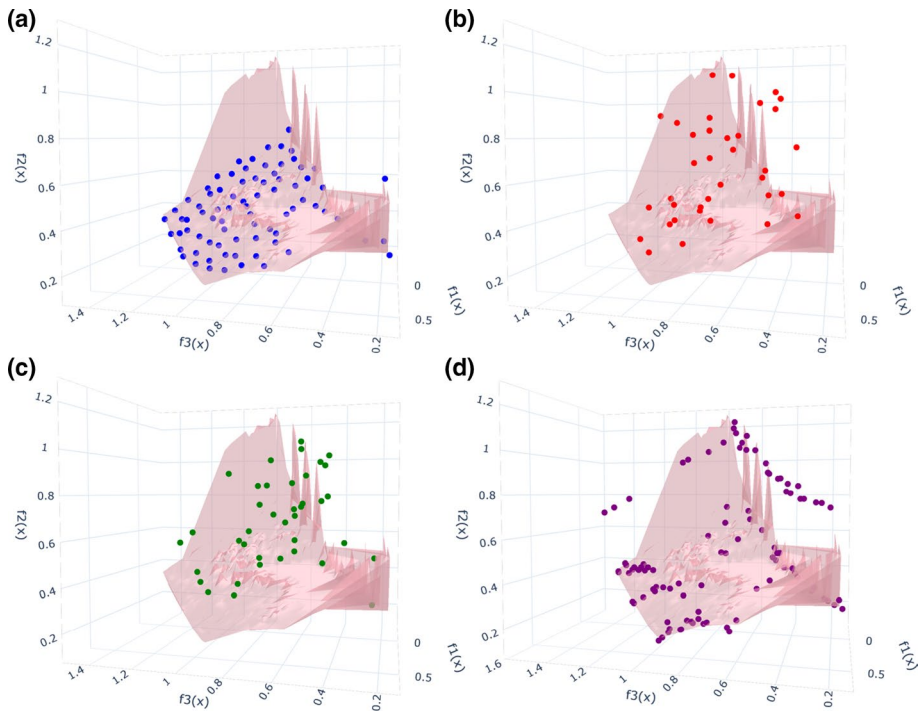


Fig. 10 Worst attainment scatter plot for fuel injector design test problem. Reference surface shown as pink mesh, with the plots corresponding to the following algorithms: **a** MVMOO, **b** LHC, **c** Random and **d** NSGA-II. Link to interactive plot: <https://plotly.com/~jmanson377/163/>

4 Conclusions

In this work we present a novel Bayesian multi-objective algorithm (MVMOO) capable of simultaneously optimising both discrete and continuous input variables. The algorithm was applied to three test cases with comparison to random and genetic algorithm-based alternatives. MVMOO was compared to NSGA-II as well as two random sampling techniques for 3 test functions where it was able to outperform NSGA-II for 2 out of the 3 test problems with a significantly reduced function evaluation budget. For the catalytic ODE test problem, MVMOO compared comparatively well with NSGA-II and a continuous variable multi-objective algorithm, TSEMO, which were able to outperform the new algorithm in terms of IGD+ and hypervolume. Overall, MVMOO was able to perform competitively when compared to NSGA-II with a substantially reduced experimental budget, providing a viable, efficient option when optimising expensive mixed variable multi-objective optimisation problems without the need of a priori knowledge of the objective function.

Acknowledgements JAM thanks the EPSRC (EP/R513258/1) and University of Leeds for funding. This work was funded, in part, by the EPSRC project “Cognitive Chemical Manufacturing” EP/R032807/1. RAB was supported by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme (RCSR/F1920/9\38).

Funding JAM thanks the EPSRC and University of Leeds for funding. This work was funded, in part, by the EPSRC project “Cognitive Chemical Manufacturing” EP/R032807/1.

Data Availability Links have been provided to interactive plots for all figures in text.

Code Availability The source code of the algorithm is available at: <https://github.com/jmanson377/MVMOO>.

Declaration

Conflict of interest All authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Deb, K.: Multi-objective optimisation using evolutionary algorithms: an introduction. In: Wang, L., Ng, A.H.C., Deb, K. (eds.) *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pp. 3–34. Springer, London (2011)
2. Gunantara, N.: A review of multi-objective optimization: methods and its applications. *Cogent Eng.* **5**, 1–16 (2018). <https://doi.org/10.1080/23311916.2018.1502242>
3. Emmerich, M.T.M., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat. Comput.* **17**, 585–609 (2018). <https://doi.org/10.1007/s11047-018-9685-y>
4. Coello, C.A.C.: Multi-objective optimization. In: Martí, R., Panos, P., Resende, M.G.C. (eds.) *Handbook of Heuristics*, pp. 1–28. Springer International Publishing, Cham (2018)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002). <https://doi.org/10.1109/4235.996017>
6. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm Evol. Comput.* **1**, 32–49 (2011). <https://doi.org/10.1016/j.swevo.2011.03.001>
7. Cui, Y., Geng, Z., Zhu, Q., Han, Y.: Review: Multi-objective optimization methods and application in energy saving. *Energy* **125**, 681–704 (2017). <https://doi.org/10.1016/j.energy.2017.02.174>
8. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: A survey on handling computationally expensive multi-objective optimization problems with evolutionary algorithms. *Soft Comput.* **23**, 3137–3166 (2019). <https://doi.org/10.1007/s00500-017-2965-0>
9. Allmendinger, R., Emmerich, M.T.M., Hakanen, J., Jin, Y., Rigoni, E.: Surrogate-assisted multicriteria optimization: complexities, prospective solutions, and business case. *J. Multi-Criteria Decis. Anal.* **24**, 5–24 (2017). <https://doi.org/10.1002/mcda.1605>
10. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: a review of Bayesian optimization. *Proc. IEEE*. **104**, 148–175 (2016). <https://doi.org/10.1109/JPROC.2015.2494218>
11. Emmerich, M.T.M., Giannakoglou, K.C., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Trans. Evol. Comput.* **10**, 421–439 (2006). <https://doi.org/10.1109/TEVC.2005.859463>
12. Emmerich, M.T.M., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: monotonicity properties and exact computation. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2147–2154 (2011)
13. Yang, K., Emmerich, M., Deutz, A., Fonseca, C.M.: Computing 3-D expected hypervolume improvement and related integrals in asymptotically optimal time. In: Trautmann, H., Rudolph, G., Klamroth, K.,

- Schütze, O., Wiecek, M., Jin, Y., Grimme, C. (eds.) *Evolutionary Multi-Criterion Optimization*, pp. 685–700. Springer International Publishing, Cham (2017)
14. Bradford, E., Schweidtmann, A.M., Lapkin, A.A.: Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *J. Glob. Optim.* **71**, 407–438 (2018). <https://doi.org/10.1007/s10898-018-0609-2>
 15. Brownlee, A.E.I., Wright, J.A.: Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation. *Appl. Soft Comput.* **33**, 114–126 (2015). <https://doi.org/10.1016/j.asoc.2015.04.010>
 16. Zhou, T., Zhou, Y., Sundmacher, K.: A hybrid stochastic–deterministic optimization approach for integrated solvent and process design. *Chem. Eng. Sci.* **159**, 207–216 (2017). <https://doi.org/10.1016/J.CES.2016.03.011>
 17. Tamilselvi, S., Baskar, S.: Modified parameter optimization of distribution transformer design using covariance matrix adaptation evolution strategy. *Int. J. Electr. Power Energy Syst.* **61**, 208–218 (2014). <https://doi.org/10.1016/J.IJEPES.2014.03.039>
 18. Ahmadi, M., Arabi, M., Hoag, D.L., Engel, B.A.: A mixed discrete-continuous variable multiobjective genetic algorithm for targeted implementation of nonpoint source pollution control practices. *Water Resour. Res.* **49**, 8344–8356 (2013). <https://doi.org/10.1002/2013WR013656>
 19. Fazlollahi, S., Mandel, P., Becker, G., Maréchal, F.: Methods for multi-objective investment and operating optimization of complex energy systems. *Energy* **45**, 12–22 (2012). <https://doi.org/10.1016/j.energy.2012.02.046>
 20. Lin, M., Chin, K.S., Ma, L., Tsui, K.L.: A comprehensive multi-objective mixed integer nonlinear programming model for an integrated elderly care service districting problem. *Ann. Oper. Res.* **291**, 499–529 (2018). <https://doi.org/10.1007/s10479-018-3078-6>
 21. Qian, P.Z.G., Wu, H., Wu, C.F.J.: Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**, 383–396 (2008). <https://doi.org/10.1198/004017008000000262>
 22. Zhou, Q., Qian, P.Z.G., Zhou, S.: A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics* **53**, 266–273 (2011). <https://doi.org/10.1198/TECH.2011.10025>
 23. Zhang, Y., Notz, W.I.: Computer experiments with qualitative and quantitative variables: a review and reexamination. *Qual. Eng.* **27**, 2–13 (2015). <https://doi.org/10.1080/08982112.2015.968039>
 24. Deng, X., Lin, C.D., Liu, K.W., Rowe, R.K.: Additive Gaussian process for computer models with qualitative and quantitative factors. *Technometrics* **59**, 283–292 (2017). <https://doi.org/10.1080/00401706.2016.1211554>
 25. Zhang, Y., Apley, D.W., Chen, W.: Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Sci. Rep.* **10**, 1–13 (2020). <https://doi.org/10.1038/s41598-020-60652-9>
 26. Benítez-Hidalgo, A., Nebro, A.J., García-Nieto, J., Oregi, I., Del Ser, J.: jMetalPy: A python framework for multi-objective optimization with metaheuristics. *Swarm Evol. Comput.* **51**, 100598 (2019). <https://doi.org/10.1016/j.swevo.2019.100598>
 27. Zhan, D., Xing, H.: Expected improvement for expensive optimization: a review. *J. Glob. Optim.* **78**, 507–544 (2020). <https://doi.org/10.1007/s10898-020-00923-x>
 28. Zhan, D., Cheng, Y., Liu, J.: Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Trans. Evol. Comput.* **21**, 956–975 (2017). <https://doi.org/10.1109/TEVC.2017.2697503>
 29. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
 30. Halstrup, M.: *Black-Box Optimization of Mixed Discrete-Continuous Optimization Problems*. Technical University Dortmund (2016). <https://doi.org/10.17877/DE290R-17800>
 31. Gower, J.C.: A general coefficient of similarity and some of its properties. *Biometrics* **27**, 857–871 (1971)
 32. Pelamatti, J., Brevault, L., Balesdent, M., Talbi, E.-G., Guerin, Y.: Efficient global optimization of constrained mixed variable problems. *J. Glob. Optim.* **73**, 583–613 (2019). <https://doi.org/10.1007/s10898-018-0715-1>
 33. de Mathews, A.G.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrà, P., Ghahramani, Z., Hensman, J.: GPflow: a Gaussian process library using TensorFlow. *J. Mach. Learn. Res.* **18**, 1–6 (2017)
 34. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva,

- F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda-Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y.: SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*. **17**, 261–272 (2020). <https://doi.org/10.1038/s41592-019-0686-2>
35. Kraft, D.: A software package for sequential quadratic programming. DFVLR-FB. 88–28 (1988)
 36. van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithm test suites. In: Proceedings of the 1999 ACM Symposium on Applied Computing—SAC '99. pp. 351–357. ACM Press, New York (1999)
 37. Baumgartner, L.M., Coley, C.W., Reizman, B.J., Gao, K.W., Jensen, K.F.: Optimum catalyst selection over continuous and discrete process variables with a single droplet microfluidic reaction platform. *React. Chem. Eng.* **3**, 301–311 (2018). <https://doi.org/10.1039/C8RE00032H>
 38. Burachik, R.S., Kaya, C.Y., Rizvi, M.M.: Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. [arXiv:1903.07041v1](https://arxiv.org/abs/1903.07041v1) (2019)
 39. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6**, 19–26 (1980). [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3)
 40. Riquelme, N., Von Lucken, C., Baran, B.: Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI). pp. 1–11. IEEE (2015)
 41. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**, 117–132 (2003). <https://doi.org/10.1109/tevc.2003.810758>
 42. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Modified distance calculation in generational distance and inverted generational distance. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 110–125 (2015)
 43. Coello Coello, C.A., Reyes Sierra, M.: A Study Of The Parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). pp. 688–697 (2004)
 44. Fonseca, C.M., Fleming, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (1996)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.