Empirical Investigation of MOEAs for Multi-objective Design of Experiments

Alexander Evans and Tinkle Chugh[0000−0001−5123−8148]

Department of Computer Science, University of Exeter, Exeter, UK
aje220@exeter.ac.uk, t.chugh@exeter.ac.uk

Abstract. Many machine learning algorithms require the use of good quality experimental designs to maximise the information available to the model. Various methods to create experimental designs exist, but the solutions can be sub-optimal or computationally inefficient. Multi-objective evolutionary algorithms (MOEAs), with their advantages of being able to solve a variety of problems, are a good method of creating designs. However, with such a variety of MOEAs available, it is important to know which MOEA performs best at optimising experimental designs. In this paper, we formulate experimental design creation as a multi-objective optimisation problem. We compare the performance of different MOEAs on a variety of experimental design optimisation problems, including a real-world case study. Our results show that NSGA-II can often perform better than NSGA-III in many-objective optimisation problems; RVEA performs very well; results suggest that using more objectives can create better quality designs. This knowledge allows us to make more informed decisions about how to use MOEAs when creating metamodels.

Keywords: Pareto optimality · Metamodelling · Evolutionary Computation

1 Introduction

Computer simulations are widely used in many scientific fields to understand systems that are complex or difficult to measure in the real world. Problems arise when simulations become computationally expensive. If one wants to understand the landscape, a small set of samples can be used to construct a metamodel. A metamodel is a regression model representative of a simulator. This allows the prediction of unsimulated areas of the landscape without expensive simulator runs. The problem of metamodeling and experimental designs is to determine what values to run the true simulator so that the metamodel regression is as accurate as possible [8]. Intuitively, it is best to uniformly spread the sample points across the domain, to maximise the information available for the regression metamodel. Uniform spread, or space filling, is the main concern of creating experimental designs; how do we position the sample points used for the metamodel across the domain space?

There are various ways to create experimental designs, the most simple method is random sampling/Monte Carlo sampling. This is very limited in its use for metamodeling, as many samples are required to fill space effectively [8]. Latin hypercube sampling (LHS) improves on random sampling by considering a one-dimensional projection property for all sample points. LHS, when combined with space filling criteria, can create effective space filling designs; however, maintaining the one-dimensional projection property is difficult, as it is a strict constraint. Methods to obtain optimised LHS are computationally expensive and for some design parameters become infeasible [15]. Single objective methods return single solutions; no alternatives are given.

By employing multi-objective optimisation (MOO) in the creation of design of experiments (DOE), we can overcome these issues and give the decision maker (DM) greater control over the optimisation process. Multiple desirable properties of experimental designs can be chosen by the DM and constraints upon solutions can be applied [6]. For example, we could set a constraint that requires solutions to be Latin hypercubes/maintain single-dimensional projection. Alternatively, the single dimension projection ability of a design can be measured as an objective that is optimised in conjunction with other objectives; this may not give exact LHSs, but it can produce families of designs close to pure LHSs in a fraction of the time.

In addition to speed and customisability, MOO facilitates the creation of a set of optimal solutions, which provides many alternatives with different evaluation values [6]; the DM can select a design that fits his/her requirements. For experimental designs specifically, the presence of alternatives is especially powerful due to the multi-modality of the problem. MOO of DoE is a multi-modal multi-objective optimisation problem (MMOP). As such, experimental designs with similar evaluation values can have vastly different sample point locations. This gives the DM an even greater choice [18]. If, for example, a chosen design produces a substandard metamodel, the decision maker has not to change his requirements; s(he) can select another experimental design that is similar within the objective space and distant in the solution space [18]. This new design still meets the decision makers requirements however may produce a far better metamodel.

With the advantages of customisability and easy access to alternatives, the use of MOO for the creation of experimental designs is considered appropriate and should be explored. MOO is frequently done with the use of multi-objective evolutionary algorithms (MOEAs); these algorithms have various strengths and weaknesses. MOEAs can solve many types of problems; they can solve non-convex problems and without derivatives [6]. They are a good choice for solving the problem of design of experiments; however, they must be prepared to overcome the unique problems presented by multi-objective design of experiments. These problems include:

1. Large Gene Count: Due to the encoding methods, each potential solution in a modest DOE optimisation problem can contain hundreds of genes. As the

number of genes becomes very large the search space increases and algorithm performance deteriorates [19].
2. Multi-Modality: Although multi-modality can be advantageous, it comes with some drawbacks. For multi-modal problems, diversity management subroutines in MOEAs can inadvertently reduce diversity in the population and therefore the solution set [18].
3. Many-Objectives: A multi-objective problem with more than three objectives is called a many-objective problem. When the number of objectives increases, the effect of evolutionary operators on the population deteriorates and algorithms can struggle to converge on the optimum [12]. In our experiments, we are executing multi and many objective problems; the algorithms must be equipped to handle both.

To understand how to best use MOO for the creation of experimental designs, we will evaluate the performance of different MOEAs in their creation. By comparing performance, we can in the future select the correct algorithms to overcome the challenges of MOO of DOE, and fully reap its benefits. Furthermore, research into how the number of objectives affects design quality has not been explored. By performing experiments on different numbers of objectives, we can understand how adding more objectives affects the quality of the designs.

The rest of the article is structured as follows. In Section 2, we provide a background of MOO and DOE. In Section 3, we formulate the DOE as a multi-objective optimisation problem. In Sections 4 and 5, we provide results for several benchmark and real-world problems by using different MOEAs. Finally, we conclude and mention the future research directions in Section 6.

2 Background

Criteria for space filling are widely researched in the experimental design field. They can be defined via distance based criteria, for example, minimax, maximin [13], potential energy [2]; or uniformity based criteria, where deviation from a uniform distribution is measured. More obscure criteria include correlation based and collapsibility criteria. Often, a single criterion is selected to optimise the sample points in an experimental design. We can remove this consideration and consider multiple objectives to create designs via multi-objective optimisation. We consider MOPs of the following form :

$$\text{minimize } \mathbf{f} = \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \text{ subject to } \mathbf{x} \in S, \tag{1}$$

with k (≥ 2) objective functions and the feasible set S is a subset of the decision space $\Re^D$. A solution $\mathbf{x^1}$ dominates another solution $\mathbf{x}^2$ if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i = 1, \ldots, k$ and $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ for at least one $i = 1, \ldots, k$. If a solution is not dominated by any of the possible solutions, it is called non-dominated. The set of such solutions is called the Pareto set. The aim of solving MOP is to find an approximated set of Pareto optimal solutions.

There are various methods of multi-objective optimisation: weighted sum, lexicographic ordering, and multi-objective evolutionary algorithms. All have

been used to optimise experimental designs with promising results. In [14], multi-objective designs were created by combining the maximin and linear correlation criteria. Their designs were good, however, their use of weighted sums makes their results weaker, as weighted sum requires strong consideration of user preference and leaves results open to human error. Moreover, the weighted sum approach is not suitable for non-convex problems [16]. Abdellatif et al. [1] use lexicographical ordering to create hybrid Latin hypercube designs that optimise both the maximin criterion and the orthogonality criterion. Although they considered the proper order of optimisation, lexicographical ordering has weaknesses concerning the limitation of the search space. Gunpinar [9] used a multi-objective approach to create a genetic algorithm selection technique for computer-assisted design. Li et al. [15] created designs using the potential energy and maximin criteria to optimise designs via a modified NSGA-III. They did not consider the use of other algorithms. We will build upon their work by investigating which MOEAs are best for optimising experimental designs.

MOEAs attempt to find a evenly distributed approximation of the Pareto-optimal set of solutions. They use evolutionary operators like crossover, mutation, and selection to converge on a global optimum. In lower dimensional spaces where the Pareto set is one or two dimensions finding the optimal set is simple. Algorithms like NSGA-II [6] can perform very well at these tasks; however, as the number of objectives increases, selection pressure falls and convergence upon the optimum is weakened [12].

Work has been done to combine decomposition with Pareto-based approaches. NSGA-III uses predefined reference points. Reference points help select solutions from the non-dominated set, maintain diversity, and enhance convergence. These reference points must be chosen by the user although typically are uniformly distributed. NSGA-III selects members that are non-dominated and close to the given reference points. Proposed by Deb and Jain [7], they showed that NSGA-III produces good results for problems of up to fifteen objectives.

RVEA [4] also uses reference points to guide selection. Like NSGA-III, RVEA partitions the objective space, and selection is performed individually inside each partition. This helps balance diversity and convergence. The authors of RVEA showed that RVEA is a competitive algorithm when compared to NSGA-III; in some test problems it outperformed.

Indicator-based approaches, like Indicator Based EA (IBEA) [20], don't use dominance as selection measure but a user specified indicator. Indicators include hypervolume or eta indicators. Therefore, indicator-based approaches do not suffer the issues of dominance-based evolutionary algorithms. They can be prohibitively expensive when the number of objectives is too large [4]. NSGA-II, NSGA-III, RVEA, and IBEA are the algorithms that we shall use for the construction of designs. These algorithms have been chosen because they are commonly used and cover various paradigms of algorithm design.

3 Multi-objective Design of Experiments

In this section, we define the objective functions and formulate the design of experiment as a multi-objective optimisation problem.

3.1 Objective functions

We have chosen four criteria that are appropriate for a design. All four are to be minimised. Having a selection of four different criteria allows evaluation of performance for different numbers of objectives. We can test the performance of each algorithm by constructing designs via two, three, and four objectives.

Potential Energy (AE) A popular space filling criterion, the Audze-Eiglais criterion [2] (also known as the potential energy criterion) fills space by treating each design point as a charged particle that repels all other particles. The total potential energy between the particles is used to evaluate their space filling. A design with low potential energy suggests the particles are spread uniformly across the domain. We chose this criterion for its excellent space filling properties. The potential energy criterion, for a design X_N, where N is the total number of samples, is denoted as:

$$PE(X_N) = \sum_{n=1}^{N-1} \sum_{j=n+1}^{N} \frac{1}{dis(\mathbf{x}^n, \mathbf{x}^j)},$$

where $dis(\mathbf{x}^n, \mathbf{x}^j)$ is the Euclidean distance between $\mathbf{x}^n$ and $\mathbf{x}^j$.

L2 Derived by Hickernell [10], the centred L2 discrepancy criterion assesses space filling by quantifying the distance between the continuous distribution of the design points and a discrete uniform distribution. We chose this criterion because it is also an effective space filling criterion that optimises from a different perspective to potential energy. For design X_N^D; where N is the number of sample points, D is the number of dimensions, and x_d^n is the nth sample in dimension d, the metric can be denoted as:

$$L_2(X_N^D) = \left(\frac{13}{12}\right)^D - \frac{2}{N} \sum_{n=1}^{N} \prod_{d=1}^{D} \left(1 + \frac{1}{2}\left|x_d^n - 0.5\right| - \frac{1}{2}\left|x_d^n - 0.5\right|^2\right)$$
$$+ \frac{1}{N^2} \sum_{j,n=1,j\neq n}^{N} \prod_{d=1}^{D} \left(1 + \frac{1}{2}\left|x_d^n - 0.5\right| + \frac{1}{2}\left|x_d^j - 0.5\right| - \frac{1}{2}\left|x_d^n - x_d^j\right|\right)$$

Collapsibility (Coll) The non-collapsibility of a Latin hypercube is advantageous for an experimental design. When two points do not have a mutual coordinate they are said to be non-collapsible. A design is non-collapsible when

no two points lie along the same one-dimensional slice; no two points share the same coordinate. Having non-collapsible points can save resources and provide more information per simulation run. Suppose that two points are collapsible along a single coordinate/variable; that is, they have the same or very similar value. If another variable/coordinate value has very little impact on the output of the simulator, those two design points will give similar outputs with no further information gained. Therefore, minimising the collapsibility of a design is important for its effectiveness; we have chosen to use a collapsibility criterion for the optimisation.

Collapsibility does not guarantee an effective space filling design; using this criterion in conjunction with other space filling criteria will allow its advantages to be fully utilised. Using the formula below can only be done using a multi-objective technique; by itself it is useless for space filling. Bates et al [3] discussed this penalisation method that allows me to assess collapsibility.

We can assess collapsibility by evaluating each one dimensional projection of the sample points. If we take the d^{th}-coordinate of all sample points in a design and sort them from smallest to largest we get the set $M_d = \{m_{d1}, m_{d2}...m_{dn}\}$. We can then create a set of equally spaced intervals that each point in M_d should lie appropriately within, $L = \{l_1...l_x\}$; where $x = N + 1$, l_1 is minimum of the sample space, and l_x is the maximum of the sample space. For a design to be a true Latin hypercube, each m_{dn} should lie within the interval $l_n \leq m_{dn} \leq l_{n+1}$. We check this equality across every m_{dn}, if any conflicts occur, we penalise the design. For a design, we sum the number of conflicts across all dimensions. A design with no conflicts is a Latin hypercube and the function would return zero. The function treats collapsibility as a minimisation problem. For a design X_N^D we can write the function as:

$$Coll(X_N^D) = \sum_{d=1}^{D} \sum_{n=1}^{N} A(M_{dn}), \;\; A(M) = \begin{cases} 0, & \text{if } l_n \leq M \leq l_{n+1} \\ 1, & \text{otherwise} \end{cases}$$

Correlation (Corr) A design that has a strong correlation between its points will have areas of the domain space unexplored, which is undesirable. However, a design that has a low correlation is not guaranteed to be space filling. Using the correlation criterion in conjunction with space filling criteria ensures that the design is non-correlated and also space filling. By including this criterion the quality of the designs should increase. In our work we shall be using the Pearson coefficient; we try to minimise the largest pairwise correlation found across the design points. If R_X is the Pearson correlation matrix of each point in design X and I is an identity matrix of the same size, we can evaluate correlation in a single value denoted as:

$$Corr(X) = \max |R_X - I|$$

3.2 Encoding

For evolutionary algorithms encoding must be considered. If we consider an experimental design to be a system of N coordinates in an D dimensional hy-

percube we can represent a design as a N by D array. Most MOEAs do not support manipulating multi-dimensional arrays within their evolutionary operators; therefore, conversion is required. When we perform evolutionary operations upon each individual we flatten the multidimensional array into a single one dimensional array. When we evaluate the performance of each solution/design we reshape the one dimensional array into its true N by D array.

Each solution is represented as an array of length ND, where every component of each coordinate is a gene that can be operated against. Each gene is a real number between 0 and 1; this is done for ease of optimisation. For example, selecting 10 samples for a 5 dimensional simulator will grant me 50 genes per potential solution. The magnitude of samples can increase quite dramatically, for 200 samples in 5 dimensions the number of genes is 1000 per solution.

Bates et al [3] compared our encoding solution to an alternative, where each sample point is represented as a single node number in the design space. The design contains a finite number of nodes each represented by an integer. A design can be represented by a sequence of integers each representing the nodes at which each sample is placed. We will not be using this encoding system as Bates et al explains; the coordinates based encoding system requires less bits and therefore has a lower risk of encountering numerical errors.

4 Numerical Experiments

In this section, we compare different MOEAs with different combinations of objectives defined in the previous section.

4.1 Problem Specifications and mumerical settings

To test the limits of the MOEAs, several experiments with different parameters shall be executed - each building on the previous. The table below describes the specifications of each problem.

Experiment	Samples	Dimensions	Genes	Objectives
DOE 5.2	25	5	125	AE, Coll
DOE 5.3	25	5	125	AE, Coll, L2
DOE 5.4	25	5	125	AE, Coll, L2, Corr
DOE 10.2	50	10	500	AE, Coll
DOE 10.3	50	10	500	AE, Coll, L2
DOE 10.4	50	10	500	AE, Coll, L2, Corr
DOE 25.2	40	25	1000	AE, Coll
DOE 25.3	40	25	1000	AE, Coll, L2
DOE 25.4	40	25	1000	AE, Coll, L2, Corr

The experiment names are based on the parameters; a suffix of “5.2” refers to a 5 dimensional design optimised by 2 objectives.

Hypervolume shall be used as a performance measure upon the algorithms NSGA-II, NSGA-III, IBEA, RVEA. The reference point is constant across problems with a mutual number of objectives; for two objectives it is 1500, 1000; for three objectives it is 1500, 1000, 100; for four objectives it is 1500, 1000, 100, 2. RVEA parameters include an adaptation frequency of 0.2 and a rate of change of penalty of 2. IBEA used a kappa value of 0.05. Simulated binary crossover and polynomial mutation were used, both with a distribution index of 20 and a probability of 1. Initial population size of 200; the initial population is identical across problems with mutual levels of dimensionality. Termination occurs after 100,000 function evaluations.

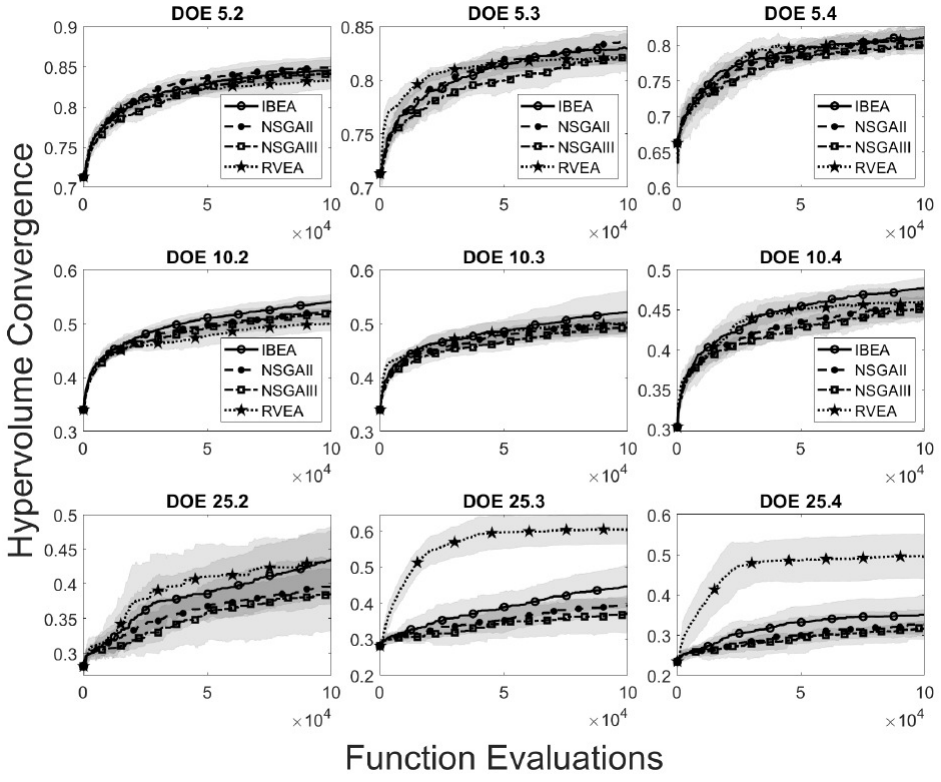

Fig. 1: PlatEMO hyper-volume performance across all nine DoE problems. The legend is the same for all subplots.

4.2 Results and discussion

The results for hypervolume convergence can be seen in Figure 1. NSGA-III is designed for many objective problems therefore it is expected to perform bet-

ter than NSGA-II in 4 objectives [7], however the results suggest otherwise. Ishibuchi et al. [11] showed research that suggests that the choice of problem has a larger effect on performance comparisons than the number of objectives. In their algorithm evaluation of a 500 item knapsack problem they showed that NSGA-II performs consistently better up to 10 objectives. DOE problems and knapsack problems are similar in that each individual is represented by a large number of genes. The performance of NSGA-II over NSGA-III remains constant across all hypervolume convergence graphs in figure 1; the large number of genes in DoE problems could be a factor in explaining the results. Ishibushi et al. also showed that NSGA-II performs better than NSGA-III when the Pareto front is very large compared to the spread of the initial solutions. For these problems, strong diversification is needed [11]. Figure 2 shows the initial population and the final population for NSGA-II and NSGA-III, we can see that the difference in spread between the final and initial populations is large and that NSGA-II produces a more diverse final population. NSGA-II's crowding distance diversity measure seems to perform better on this class of problem, as it does with knapsack problems.

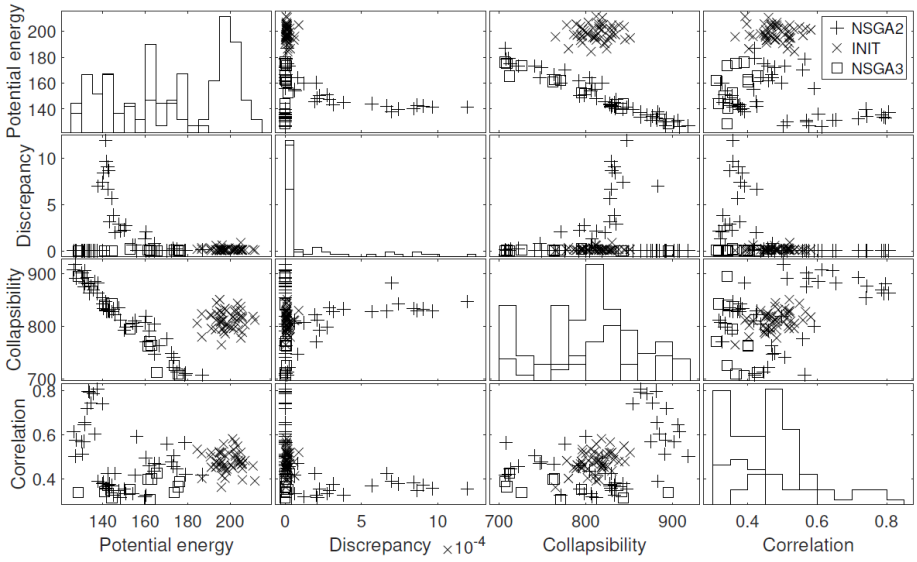

Fig. 2: DOE 25.4: NSGA-II and, NSGA-III final and initial population

As both algorithms had the same initial population, it would be worth confirming performance comparisons by re-running the experiment with a different initial population. Different methods of initial population generation should be considered also; in these experiments initial populations were random. Perhaps an initial population of Latin hypercubes would produce better results as the

collapsibility criterion is attempting to achieve Latin hypercube qualities. A non-optimal Latin hypercube initial population would help the MOEAs produce good designs with less work.

RVEA performed better than other algorithms for problems with more than two objectives. Cheng et al. compared RVEA with other popular MOEAs and showed their performance was better than other many-objective evolutionary algorithms. RVEA's strengths in benchmark problems have been replicated in MOO of DoE. This high performance is likely due to the unique scalarisation approach employed by RVEA. IBEA has consistently good performance across all test problems, this suggests it is good as a general use algorithm for MOO of DoE.

Cheng et al. [5] showed that in many objective problems with high gene count, IBEA and RVEA performed better in approximating the Pareto front than NSGA-III. DoE MOO's high gene count has replicated these results as both IBEA and RVEA perform better than NSGA-III across all problems. However, in Cheng et al.'s work neither IBEA nor RVEA perform best overall, which is also confirmed by our hypervolume results.

Design of experiments MOO is a multi-modal multi-objective optimisation problem, two solutions that may be distant in the decision space may be close or overlapping within the objective space. A consequence of multi-modality is that conventional MOEAs struggle to maintain diversity within the decision space. MOEAs will remove solutions that are crowded in the objective space when they may be distant in the decision space. Removal of distant individuals reduces diversity in the decision space. This process - along with genetic drift and the consequence of crossover and mutation not producing diverse offspring effectively - reduces diversity in the objective space as the population's decision variables are somewhat homogeneous [18]. Consequences of multi-modality may explain the irregular, disconnected final populations found by IBEA, NSGA-II, and NSGA-III; the objective space can be seen in figure 3. Multi-modality has reduced diversity in the decision space and, therefore, reduced diversity in the objective space that can be seen as disconnected, unexplored regions.

Disconnection is not seen in RVEA; RVEA's unique angle penalised distance (APD) scalarisation function gives it the ability to maintain uniformity across the population. Cheng [4] et al. showed that RVEA produces better quality Pareto front approximations than NSGA-III in multi-modal MOO problems, as it does in our results.

5 Case Study

The ultimate goal of experimental design is to create effective metamodels; therefore, the quality of metamodels should be verified as a means of determining optimisation success. We used our designs to explore the landscape of the ratio between time and molecular weight produced in the batch creation of branched polymers. Parameters for this simulation include $Time$, the duration of each batch production, M, monomer concentration, I, initiator, and T the tempera-

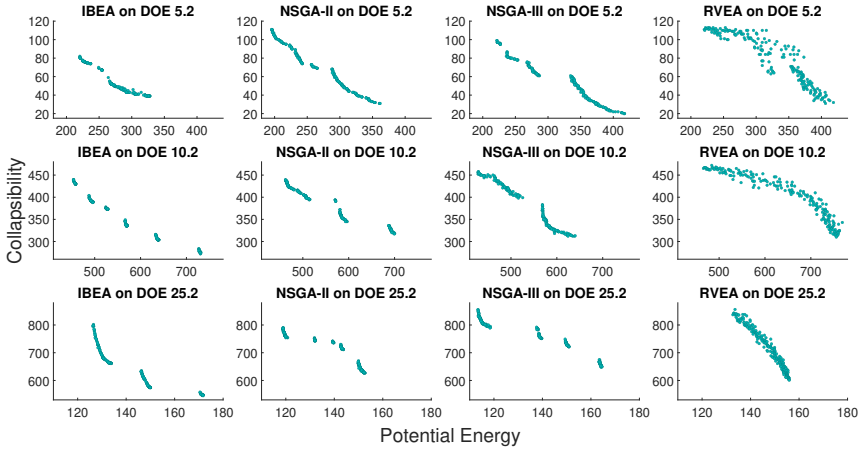

Fig. 3: Final populations for each algorithm, PlatEMO, two objective problems

ture of the batch production vessel. For more details about the problem, see [17]. Bounds for each parameter can be seen in Table 1.

Parameter	Range	Unit
Time	30 to 10,000	seconds
Monomer concentration	10 to 14	Meters cubed per second
Initiator	3E-5 to 1.5E-4	Meters cubed per second
Temperature	60 to 80	degrees centigrade

Table 1: Branched polymer input specifications.

5.1 Multi-criteria Decision making

Once optimisation is complete the DM can select a representative from the approximated Pareto set. In order to validate the success of the optimisation we select a design from the final population to use as an experimental design in the polymerisation problem. We used decomposition to select a choice. The weights for the four objectives are $[0.1, 0.1, 0.7, 0.1]$ (potential, discrepancy, collapsibility, correlation). For three objectives, $[0.3, 0.1, 0.6]$ (potential, discrepancy, collapsibility). In two objectives, [0.4, 0.6] (potential and collapsibility). We considered collapsibility to be a very important property when exploring the landscape therefore a high weight was given. The weights suffer the disadvantage of human error, we cannot see all possibilities and must make assumptions. Investigations with other weights are not within the scope of this paper.

5.2 Results

The experiment was carried out with varying numbers of objectives on the four different algorithms. A Gaussian process (GP) was chosen to explore the outputs of the function because it is non-parameterised and the confidence intervals provide a good performance measure. A GP was fitted according to the various sizes of experimental design. Samples were then taken from these GPs and their predictions were compared with the true function value. All GPs were built using a matern32 kernel. The design's dimensions were scaled to fit the bounds of the input variables.

After the GPs have been created, we evaluate their performance using the averages of the confidence intervals across the entire landscape. We created a Cartesian product across the landscape; four evenly spaced intervals for four variables produce a Cartesian product of 256 members/points in a grid across the domain space. If the experimental design has accurately mapped the landscape, then the confidence intervals of these GPs should be small and uniform.

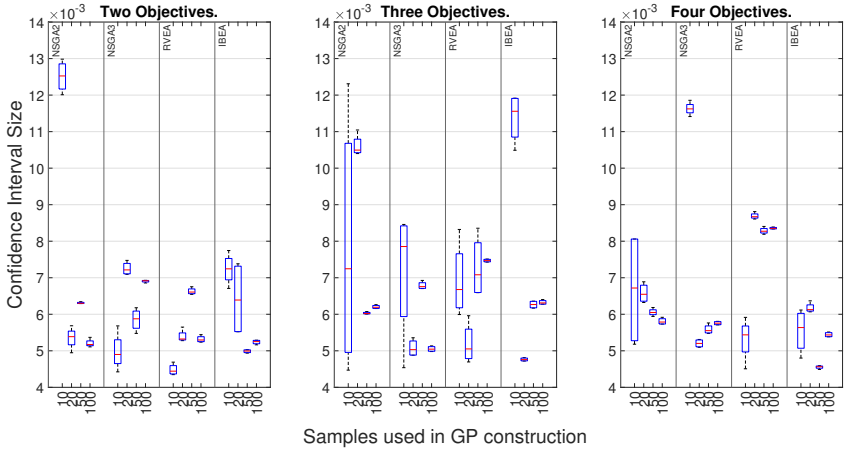

Fig. 4: Confidence intervals for the Cartesian evaluation, PlatEMO

Figure 4 shows box-plots of the confidence intervals using the data from PlatEMO. No algorithm stands out as superior; however, the interquartile ranges (IQR) generally fall as the number of samples increases. In four objectives, many of the IQRs are low and thin. This suggests that the inclusion of more objectives produces more desirable qualities; treating DOE optimisation as a many-objective optimisation problem will produce better metamodels. In four objectives a correlation criterion is included, despite it not conflicting with other objectives the quality of the designs seems to improve. Suggesting that adding objectives that define good qualities, but from a different perspective, can add to the overall quality of the design. Three objectives has noticeably wider IQRs

than two and four objectives. Work should be done to investigate whether it is the choice of criterions or the number of them that most effects performance; this result gives merit to both theories.

6 Conclusions

In this paper, we have explored the use of MOO as a tool for design of experiments. We successfully formulated the problem of DOE into a MOO problem. Designs have been successfully optimised and used to investigate a real-world problem where their success was verified. Various objective functions were chosen, potential energy, discrepancy, collapsibility, and correlation; these cover various desirable qualities. We have successfully investigated the performance of different MOEAs when optimising experimental designs. We chose four different algorithms to compare performance. Dominance-based approaches (NSGA-II), reference-based algorithms (RVEA and NSGA-III), and indicator-based algorithms (IBEA) were used. The high gene count and distance between initial and final population lead NSGA-II to converge better than NSGA-III in higher objectives. RVEA's unique scalarisation approach led it to perform well with high gene counts; IBEA performed well as a general use algorithm. We were successful in using optimised designs in the construction of metamodels. In the branch polymer metamodels, two and four objectives had low and narrow IQRs. Suggesting there is an optimal choice of criteria.

To further improve the knowledge of how best to utilise MOO in experimental design, further experiments should be conducted. Algorithms designed to tackle problems with large gene counts should be explored. Exploration of the use of more objectives/different objectives is needed to confirm how the choice of objectives effects the performance. More study into the performance of these designs in real life problems should be done. Ishibuchi et al [12] discussed how for many-objective problems the number of solutions needed to best approximate the Pareto set becomes exponentially large; a bigger population is needed. We chose 200 individuals for our work however exploration of optimisations with higher population could be explored. Working on using different reproduction operators is also one of the future works. It is important to consider decision-maker's preferences before or after the optimisation process. This work finds a set of nondominated solutions and does not utilise preferences. Therefore, getting one solution based on the preferences (e.g., weights, desirable objective function values) will be in our future research.

References

1. Abdellatif, A.S., Abdelhalim, M.B., El Rouby, A.B., Khalil, A.H.: Hybrid latin hypercube designs. In: 2010 The 7th International Conference on Informatics and Systems (INFOS). pp. 1–5 (2010)
2. Audze, P., Eglājs, V.: New approach to the design of multifactor experiments, problems of dynamics and strengths 35. Zinatne Publishing House pp. 104–107 (1977)

3. Bates, S., Sienz, J., Langley, D.: Formulation of the audze–eglais uniform latin hypercube design of experiments. Advances in Engineering Software **34**(8), 493–506 (2003)
4. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(5), 773–791 (2016)
5. Cheng, R., Jin, Y., Olhofer, M., sendhoff, B.: Test problems for large-scale multiobjective and many-objective optimization. IEEE Transactions on Cybernetics **47**(12), 4108–4121 (2017)
6. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
7. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
8. Garud, S., Karimi, I., Kraft, M.: Design of computer experiments: A review. Computers & Chemical Engineering **106**, 71–95 (2017)
9. Gunpinar, E., Khan, S.: A multi-criteria based selection method using non-dominated sorting for genetic algorithm based design. Optimization and Engineering **21** (12 2020)
10. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Mathematics of Computation **67**(221), 299–322 (1998)
11. Ishibuchi, H., Imada, R., Setoguchi, Y., Nojima, Y.: Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In: 2016 IEEE Congress on Evolutionary Computation (CEC). pp. 3045–3052 (2016)
12. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: IEEE World Congress on Computational Intelligence. pp. 2419–2426 (2008)
13. Johnson, M., Moore, L., Ylvisaker, D.: Minimax and maximin distance designs. Journal of Statistical Planning and Inference **26**(2), 131–148 (1990)
14. Joseph, V.R., Hung, Y.: Orthogonal-maximin latin hypercube designs. Statistica Sinica **18**(1), 171–186 (2008)
15. Li, Y., Li, N., Gong, G., Yan, J.: A novel design of experiment algorithm using improved evolutionary multi-objective optimization strategy. Engineering Applications of Artificial Intelligence **102**, 104283 (2021)
16. Miettinen, K.: Nonlinear multiobjective optimization. Kluwer, Boston, MA (1999)
17. Mogilicharla, A., Chugh, T., Majumder, S., Mitra, K.: Multi-objective optimization of bulk vinyl acetate polymerization with branching. Materials and Manufacturing Processes **29**, 210–217 (2014)
18. Peng, Y., Ishibuchi, H., Shang, K.: Multi-modal multi-objective optimization: Problem analysis and case studies. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1865–1872 (2019)
19. Zille, H., Mostaghim, S.: Comparison study of large-scale optimisation techniques on the lsmop benchmark functions. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8 (2017)
20. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature - PPSN VIII. pp. 832–842. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)