# Identifying Solutions of Interest for Practical Many-objective Problems using Recursive Expected Marginal Utility

Hemant Kumar Singh
The University of New South Wales
Canberra, ACT 2600, Australia
h.singh@adfa.edu.au

Tapabrata Ray
The University of New South Wales
Canberra, ACT 2600, Australia
t.ray@adfa.edu.au

Tobias Rodemann
Honda Research Institute
63073 Offenbach/Main, Germany
Tobias.Rodemann@honda-ri.de

Markus Olhofer
Honda Research Institute
63073 Offenbach/Main, Germany
Markus.Olhofer@honda-ri.de

## ABSTRACT

Real-world problems often involve optimization of multiple conflicting objectives. Significant research has been directed recently towards development of multi-objective evolutionary algorithms that are scalable, i.e., able to deal with problems involving more than 3 objectives, commonly referred to as *many*-objective optimization problems. This has led to the emergence several new techniques that can deliver a set of trade-off solutions to approximate the Pareto optimal front of the problem. However, means to select solution(s) from this large trade-off set for final implementation/decision making has received relatively scarce attention. This paper aims to study and demonstrate the performance of recursive expected marginal utility ($\text{EMU}^r$) approach for informed decision-making. Towards this goal, we apply the $\text{EMU}^r$ approach to identify solutions of interest for two practical examples and analyze the obtained set of solutions. The study highlights the desirable trade-off characteristics that the chosen solutions have over the rest of the trade-off set, highlighting its potential as a decision-making tool, especially in cases where other preference information or domain knowledge is unavailable.

## CCS CONCEPTS

• **Applied computing** → **Multi-criterion optimization and decision-making**;

## KEYWORDS

Multi-objective optimization, solutions of interest, decision-making

**ACM Reference Format:**
Hemant Kumar Singh, Tapabrata Ray, Tobias Rodemann, and Markus Olhofer. 2019. Identifying Solutions of Interest for Practical Many-objective Problems using Recursive Expected Marginal Utility. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19)*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn


*GECCO '19, July 13–17, 2019, Prague, Czech Republic*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION AND BACKGROUND

It is common to encounter situations in real-world problems where multiple conflicting performance criteria need to be optimized simultaneously. Such problems are commonly referred to as multi-objective optimization problems (MOP). Empirical studies have shown that solving such problems becomes increasingly difficult for traditional Pareto-dominance based methods beyond three objectives [15, 17], and therefore such problems are now further differentiated as *many-objective* optimization problems (MaOP). The difficulties lie not only in obtaining well converged and diverse set of solutions on the Pareto-optimal front (PF), but also in the subsequent visualization and selection of solutions to aid decision-making. Evidently, as the number of objectives increases, it becomes difficult to cognitively interpret the trade-off between them. This challenge is further aggravated by the fact that the number of solutions to cover the PF of MaOPs are usually very high.

The challenges inherent in MaOPs have been recognized in the field of evolutionary multi-objective optimization (EMO) for more than fifteen years; and some of the early papers exploring the topic include [12, 15, 21]. Initial attempts to solve such problems involved use of modified dominance relations [25], secondary ranking [18], dimensionality reduction [28], etc. Subsequently, the use of *decomposition* of objective space was observed to be beneficial in overcoming the loss of selection pressure. Following the publication of MOEA/D [31], the idea attracted significant attention for solving MaOPs. Many subsequent improvised algorithms have been since proposed that use this core idea, often in conjunction with other beneficial components such as non-dominance or indicators. A recent survey of objective space decomposition-based techniques can be found in [29].

With the development of the above tools, a number of applications have also been modeled and solved as MaOPs to obtain their approximate PF. Some examples include 9-objective radar waveform optimization [14], 7-objective hybrid car controller [24], 3-6-objective space trajectory design [16, 26], 4-objective wind-farm layout optimization [4], 4-objective energy management problem [22], 5-objective water resource management [1], 10-objective general aviation aircraft design [13], etc.

Although development of MaOPs has been of significant interest, relatively scarce attention has been paid towards informed *a posteriori* decision-making. In most of the above works (including practical problems), the study concludes after an approximate non-dominated PF set is identified and assessed using metrics such as inverted generational distance (IGD) and/or hypervolume (HV). In practice, not all solutions obtained can be implemented or even duly examined. The target of informed decision-making exercise is therefore to look at a set of (potentially large) non-dominated solutions obtained during the search process and select a few (typically less than 10 [5]) that exhibit desirable trade-offs and can be presented to a decision maker for further consideration. If some preference structure is available from a decision-maker (DM)/user, it can be incorporated in the selection; however if no such information is available, the task becomes even more challenging. In either case, the identification of such solutions can also help the search itself in an online/interactive mode, where the search aims to focus on obtaining certain good solutions instead of the entire PF.

In absence of prior information, one of the approaches is to look for certain key solutions referred to as *knee* [6] solutions or more generally *solutions of interest* (SOI) [5]. For a solution to be an SOI it is desirable that improvement in any objective along the PF should come at a significant sacrifice of at least one another objective. This makes the choice of other neighboring solutions less preferred, thus narrowing down the set. To enable this approach, some quantitative measures have been proposed to identify SOIs from the PF approximation. Some of these are specific to two objectives (or their extension to higher objectives is not straightforward), such as bend angle, reflex angle, and trade-off approach[11]. The bend/reflex angle techniques attempt to quantify the knee regions for bi-objective problems by considering the angle formed by a solution under consideration with a defined set of points on its either side. The trade-off approach directly defines how much of sacrifice is acceptable for a unit gain on the objectives. Other methods, that can be applied to MaOPs, include maximum convex bulge [9], hypervolume contribution [32], local curvature [3], expected marginal utility (EMU) [6] and solution density [30].

Of the above-mentioned approaches, EMU is most promising as it is not computationally expensive (unlike hypervolume), doesn't forgo trade-off information (unlike convex bulge, solution density), and doesn't assume symmetric PF (unlike local curvature). In EMU, the linear utility $U(\mathbf{x}, \lambda)$ of a solution is calculated using an expression of the form $U(\mathbf{x}, \lambda) = \lambda f_1 + (1 - \lambda) f_2$ (for two-objective case), where $\lambda \in [0, 1]$. For a given preference direction $\lambda'$, the marginal utility $U'(\mathbf{x_i}, \lambda')$ of a solution $\mathbf{x_i}$ is defined as the additional cost the decision maker would have to incur if the second best individual is chosen instead of the individual with the highest utility (Eq. 1). If several different uniformly distributed weight vectors are constructed to scalarize the problem, a solution with higher EMU than another will have a better value in a larger number of such scalarized problems (see [6, 11] for more details).

$$U'(\mathbf{x_i}, \lambda') = \begin{cases} \min_{j \neq i} U(\mathbf{x_j}, \lambda') - U(\mathbf{x_i}, \lambda'), & \text{if } i = \arg\min U(\mathbf{x_j}, \lambda'), \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

A limitation of EMU is that as the number of objectives and solutions increases, a very small number of solutions are actually assigned a unique non-zero value, making it difficult to influence the ranking (for online use) as well as to select required number of solutions due to incomplete ordering. A more detailed discussion on the above measures can be found in [5]. To address the above research gaps, a recursive EMU approach (EMU$^r$) was suggested in [5] with an aim to (a) select a set of $K$ preferred solutions from a given non-dominated front considering trade-off behavior and diversity, (b) prioritize the selected solutions among themselves, (c) characterize the selected solutions as internal/peripheral and (d) deal with problems involving many objectives with a large trade-off set of solutions.

In this study, we attempt to bring the above two aspects (practical application and decision-making) together. Towards this end, we investigate two practical optimization problems from the point of view of identifying the required (small) number of SOIs. The studies are conducted by application of EMU$^r$ on the non-dominated solution datasets obtained for the two practical problems: hybrid car controller optimization (HEV) [24], and building energy management optimization (BEM) [22]. The data has been generated through application of evolutionary multi-objective approaches. While the previous studies have been aimed at obtaining the good PF approximation, in this study we intend to obtain a few solutions of interest from the previously obtained sets in order to aid informed decision-making. Within the scope of this paper, the interactions with the domain experts is not considered; but the quality of selected solutions are analyzed through visualization and proximity graphs to establish their desired properties.

A brief introduction to the approach is presented in the Section 2, followed by illustrations on a test problem in Section 3. Results and analysis on the practical problems are given in Section 4, with concluding remarks in Section 5.

## 2 APPROACH

The approach studied herein operates by calculating EMU recursively and is referred to as EMU$^r$. The approach identifies $K$ unique, sparse solutions of interest following the steps outlined in Algo. 1. The approach was developed and benchmarked with some of the aforementioned techniques in a previous publication [5]. The key steps of the approach are discussed in the following sub-sections for completeness. The approach takes as its input a set of non-dominated solutions ($NDF$), and the number of solutions ($K$) required by the user/decision-maker.

### 2.1 Generate reference vectors/directions

A uniformly distributed set $W$ of reference points ($|W| \geq K$) is generated spanning a hyperplane with unit intercepts in each objective axis using Das and Dennis's normal boundary intersection (NBI) method [10]. The approach generates $|W|$ reference directions (also sometimes referred to as weight vectors) by joining reference points to the origin. A spacing parameter $p$ is a user input which determines the density of points on the hyperplane. Since the number of points generated using the NBI method grows exponentially with an increase in objectives, $p$ needs to be adjusted so as to generate a set of a size that is not computationally overwhelming to store and evaluate.

**Algorithm 1** Identification of SOI using $\text{EMU}^{\text{r}}$

**Input:** $NDF$ (objective values of all unique non-dominated solutions ($P$) for an $m$ objective problem), $K$ (number of SOI required)
**Output:** $\text{SOI}_{\text{F}}$ (Final $K$ solutions selected)

**Generate** $|W|$ ($|W| \geq K$) reference directions ($W$).
$\text{SOI} = \emptyset$, $\text{SOI}_{\text{A}} = P$.
**while** $|\text{SOI}| < K$ **do**
  {Start *Outer Loop*}
  **while** $|\text{SOI}_{\text{A}}| > 0$ **do**
    {Start *Inner Loop*}
    **Compute** $\text{EMU}^{\text{r}}$ of each solution in $\text{SOI}_{\text{A}}$ using $W$.
    **Associate** solutions to their closest reference directions based on angle.
    **Identify** solutions with highest $\text{EMU}^{\text{r}}$ values along each non-empty reference directions.
    **Select** a set of solutions ($S$, $S \subseteq \text{SOI}_{\text{A}}$) with higher $\text{EMU}^{\text{r}}$ values than the neighboring solutions (based on neighboring reference directions).
    **if** $(|S| \leq K) \vee (|S| = |\text{SOI}_{\text{A}}|)$ **then**
      **break**.
    **else**
      $\text{SOI}_{\text{A}} = S$.
    **end if**
    {End *Inner Loop*}
  **end while**
  $\text{SOI} = \text{SOI} \cup \text{SOI}_{\text{A}}$.
  $\text{SOI}_{\text{A}} = P \setminus \text{SOI}$.
  {End *Outer Loop*}
**end while**
Order solutions in SOI based on $\text{EMU}^{\text{r}}$ values.
**if** $|\text{SOI}| > K$ **then**
  Classify solutions of the SOI into three sets: $Internal$, $Peripheral$ and $Peripheral_E$.
  **if** $|Internal| > K$ **then**
    Pick $K$ diverse $Internal$ solutions to construct $\text{SOI}_{\text{F}}$.
  **else if** $|Internal| < K$ **then**
    Combine $Internal$ with $K - |Internal|$ diverse solutions from $Peripheral \cup Peripheral_E$ to construct $\text{SOI}_{\text{F}}$.
  **end if**
**else if** $|\text{SOI}| = K$ **then**
  $\text{SOI}_{\text{F}} = \text{SOI}$.
**end if**

## 2.2 Compute $\text{EMU}^{\text{r}}$

In the first stage, the objective values ($NDF$) of all the unique non-dominated solutions ($P$) are scaled using ideal and nadir vectors of the set. The EMU values of all solutions are computed using the reference directions [6] generated previously. The solutions with non-zero EMU values are ordered based on descending values (higher is better) of EMU and assigned to the first “front” (not to be confused with a non-dominated front). In the next stage, only the solutions having zero EMU values are considered and their EMU values are recomputed using the complete set of reference directions. The solutions with non-zero EMU values in this stage are ordered and assigned to the second front. This process continues until at most one solution has a zero EMU value. Thus, all the solutions can be ordered in each front. Thereafter, maximum EMU value of each front (starting from the last) is added to the EMU value of all members belonging to the next front. Thus all solutions in the trade-off set would have a non-zero $\text{EMU}^{\text{r}}$ value.

## 2.3 Associate the points to reference directions

In the normalized objective space, one can compute the angle between a solution and all reference vectors. A solution is assigned to the reference direction to which it has the smallest angle. Such a scheme of minimum angle based association is often used in the contemporary decomposition based algorithms to construct a sub-population focusing on the scalarized sub-problem along a given direction.

## 2.4 Identify best $\text{EMU}^{\text{r}}$ solutions

The previous step may result in having potentially multiple solutions associated with a single reference direction, while some reference directions might not have any. For each non-empty reference direction, the associated solution with the maximum $\text{EMU}^{\text{r}}$ value is selected.

## 2.5 Select the SOIs

The selection of the best $\text{EMU}^{\text{r}}$ solutions gives the first cut selection of SOIs, but the set could still be quite large, e.g., if one solution ends up being selected along each of the reference directions. To narrow down the set further, for any given solution along a non-empty reference direction, all solutions associated with its immediate neighboring reference directions are compared based on $\text{EMU}^{\text{r}}$ measure. ‘Neighboring directions’ here refers to the directions passing through the closest point(s) on the hyperplane to point that the current direction passes through. If the solution along the original non-empty reference direction is still the best following the neighborhood comparison, this solution is selected. Otherwise, no solution is selected as the best for this comparison. Please take note that $\text{EMU}^{\text{r}}$ values are not recomputed during the comparison and the order of comparison does not affect the selection since none of the solutions are deleted. The process is repeated for every non-empty reference direction to construct the most preferred set of solutions ($\text{SOI}_{\text{A}}$).

If the number of solutions in $\text{SOI}_{\text{A}}$ is more than $K$, the above processes (*Inner Loop*) continue until cardinality of the set $\text{SOI}_{\text{A}}$ does not change or is less than $K$. At the end of the *Inner Loop*, the members in $\text{SOI}_{\text{A}}$ are used to construct SOI.

The *Outer Loop* starts if SOI contains fewer than $K$ solutions. In this loop, $\text{SOI}_{\text{A}}$ is set to contain the remaining solutions in $P$, i.e, $P \setminus \text{SOI}$ and the *Inner Loop* continues until the cardinality of the set SOI reaches at least $K$.

At the end of both the loops, the reduction stage begins if SOI contains more than $K$ solutions. In this stage, solutions belonging to SOI are further divided into three classes: $Peripheral_E$, $Peripheral$ and $Internal$. Solutions having at least one objective at its minimum value among the non-dominated solutions belong to the $Peripheral_E$ class. Solutions closest to the reference directions passing through the edges of the hyperplane belong to the $Peripheral$ class. The remaining solutions belong to the $Internal$ class. Based on the number of solutions in SOI at this step, two cases can occur:

- If the set $|\text{SOI}| > K$, the number of solutions belonging to the class $Internal$ is observed. If the number exceeds $K$, then $K$ diverse solutions are picked from $Internal$ class using distance based subset selection (DSS) technique [27]. The DSS technique progressively selects points (one by one) such that the nearest neighbor distance from the currently selected set is maximized; resulting in a good diversity of selected points. In the event the $Internal$ class contains fewer than $K$ solutions, $K - |Internal|$ solutions are picked from other classes using DSS. Solutions belonging to the $Internal$ class

are preferred over the solutions belonging to $Peripheral$ and $Peripheral_E$ class since prior studies have indicated that the decision makers tend to prefer solutions from the middle section of the POF, to avoid extreme objective values [9]. The picked solutions are combined to form the final $\text{SOI}_\text{F}$.
- If SOI contains exactly $K$ points, then $\text{SOI}_\text{F}$ = SOI.

## 3 ILLUSTRATION ON A TEST PROBLEM

Before we move on to observing the results obtained by the proposed approach on the practical MaOPs under consideration, we demonstrate the technique for a 3-objective test problem in order to illustrate the experimental design as well as visualize some of the solutions that the proposed approach identifies. For this purpose, we consider the DEB3DK benchmark problem proposed in [6]. The shape of the POF of this problem has difficult features such as multiple knees and discontinuities. The problem contains four (internal) knee regions as shown in Fig 1(a), which form convex bulges towards the ideal point.

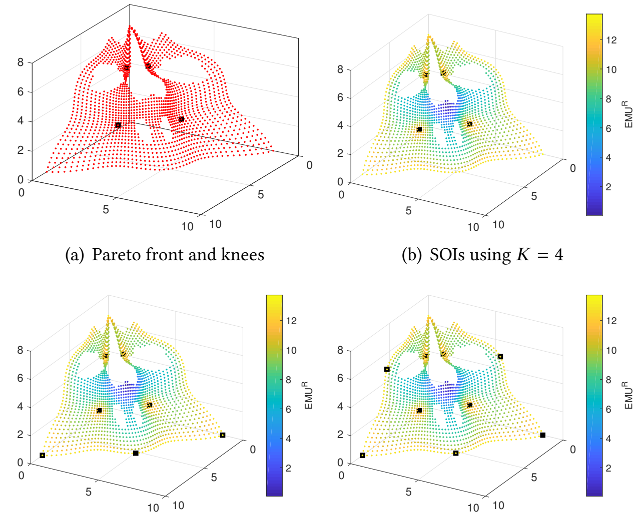
For this problem, we ran the proposed approach with different number of desired points, i.e., $K = 4, 7$, and 9. For the calculation of EMU, the spacing parameter is set to 23, resulting in $|W| = 300$ reference vectors. The dataset considered has originally 1611 points sampled on the PF, of which a small number of SOIs (4,7,9) are targeted.

### 3.1 Basic visualization of obtained SOIs

The obtained SOIs for the problem with $K = 4, 7, 9$ are shown in Figs 1(b), 1(c) and 1(d) respectively. In each of these figures, all the available data points are shown in color according to their $\text{EMU}^\text{r}$ values. It can be clearly seen from the color scale that the high $\text{EMU}^\text{r}$ values occur close to the true knee points of the problem, resulting in the selection of all the internal knees (only) when 4 points are requested by the decision maker. The other region which has high values of $\text{EMU}^\text{r}$ are distributed along the edges of the PF. Since there are large number of them, the approach identifies the sparsely located points within this set once the internal points have been selected. For example, when $K = 7$, after the selection of the 4 knees (which belong to *Internal* category), the rest of the 3 are selected from the solutions in the $Peripheral/Peripheral_E$ categories and can be seen covering the bottom edge of the front relatively uniformly. When $K = 9$, two further points are added close to the middle of the other two edges to maintain a good diversity among the SOIs.

### 3.2 Proximity plots

In order to verify that the selected SOIs exhibit favorable trade-offs in their neighborhoods, we present a proximity graph visualization of the obtained SOI in Fig. 2. The figure is shown for the case of $K = 7$, and the selected SOIs are shown in magenta color with their solution id (in the given dataset) indicated alongside. A directed connection is shown between the SOIs and the neighboring solutions, defined as the ones that have the best SOIs along the neighboring reference vectors to the one that the SOI is attached to. It is also easy to identify in this case that some of the SOIs have fewer neighboring solutions attached to them (e.g. 1562 and 1611), which are evidently the extreme solutions of this PF.



(a) Pareto front and knees (b) SOIs using $K = 4$

(c) SOIs using $K = 7$ (d) SOIs using $K = 9$

**Figure 1: Results for DEB3DK problem**

### 3.3 Effect of the dataset size

In addition to identifying the SOI for the given dataset, we also want to study the effect of the size of the dataset. More specifically, we want to examine if the selected SOIs *stabilize* as the density of the points in the dataset is increased. This can help in judging how reliable the SOIs are based on the sparsity of the current dataset under consideration; and consequently whether more data-points are required to come to a reliable (stable) set of SOIs.

In order to conduct this study, from the full available dataset, we progressively select subsets of well distributed data-points comprising 20%-100% of the points in the entire dataset using DSS. Assuming $K = 7$ SOIs are required by a DM, we select these 7 points from each of the reduced subsets and observe the change in distribution of the points. This is shown in Fig 3. We note for this problem that it took about 50% of the original dataset for the SOIs to roughly converge to the eventual SOIs that were shown previously in Fig. 1(c).

Apart from above visualization, the convergence can also be observed quantitatively, by using a similarity metric such as inverted generational distance (IGD). IGD is a commonly used unary metric in evolutionary multi-objective optimization (EMO) domain to quantitatively compare the quality of a PF approximation obtained by an algorithm. It does so by comparing the PF approximation to a *reference set*, which in the case of benchmark problems is a known set of well distributed Pareto optimal points. We can use the same concept in our study to quantify how far the obtained set of SOIs in the subset of the data is from the reference set comprising the SOIs obtained on the entire dataset. For a set $P$ whose quality needs to be evaluated w.r.t. a reference set $R$, the formal definition of IGD is given in Eq. 2. Note that because the IGD calculations involve Euclidean distance calculation, the objective values must be normalized so as to avoid any bias in distance calculations. Therefore, the SOI point sets are linearly normalized with respect to

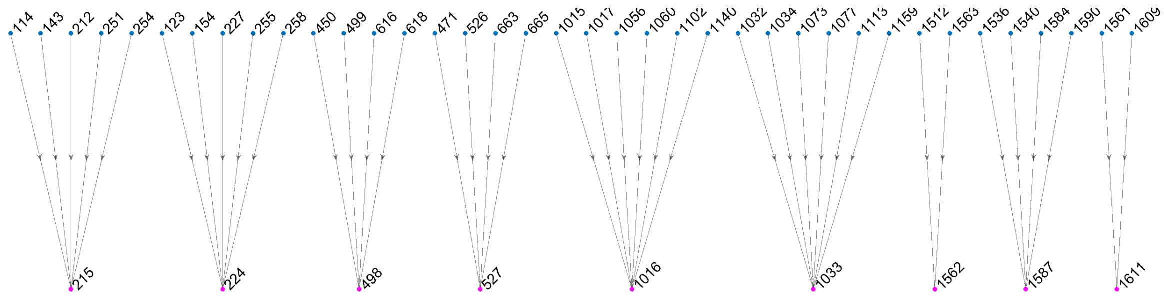
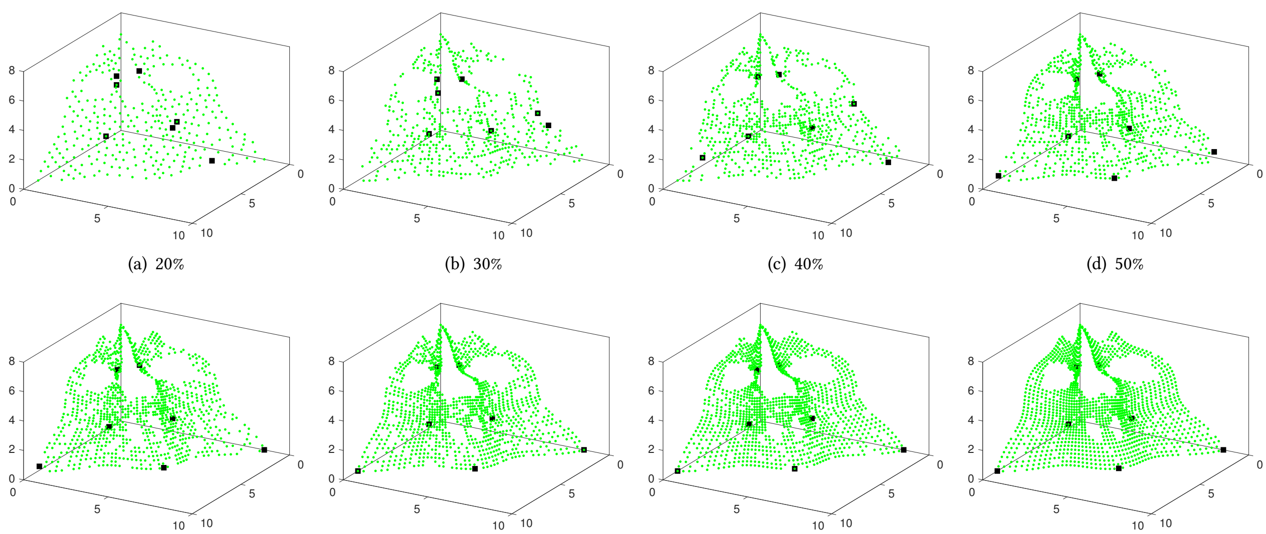



**Figure 2: Proximity plot of the identified SOIs (shown in magenta color) with $K = 7$ with neighborhood solutions (shown in blue color). The direction of arrows points towards the solution with better EMU$^{\mathbf{r}}$. All SOIs identified using the proposed approach are thus deemed to be better than the solutions in their neighborhood.**



(a) 20% (b) 30% (c) 40% (d) 50%

(e) 60% (f) 70% (g) 80% (h) 90%

**Figure 3: Observing the effect of datasize on SOIs (for the case $K = 7$) for DEB3DK problem**

the bounds of the dataset (lowest value mapped to 0 and highest value mapped to 1) for the IGD calculation. A lower value of IGD implies closer (similar) sets in the Euclidean space, with a value of 0 meaning the sets are identical.

$$\mathrm{IGD}(P, R) = \frac{1}{|R|} \sum_{i=1}^{|R|} \min_{\mathbf{f} \in P} d(\mathbf{p}_i, \mathbf{f}) \tag{2}$$

where $d(\mathbf{p}_i, \mathbf{f})$ is the Euclidean distance between the points $\mathbf{p}_i$ and $\mathbf{f} \in P$.

The progression of IGD value for the problem, as the subset size is increased from 20% to 100% is shown in Fig. 4. The values beyond 800 points (roughly 50%) are fairly low, indicating the convergence of SOIs to those found from the entire dataset. Simultaneously in Fig. 4 one can also see the number of common SOIs between intermediate sets and the final set. At around 60% of the original dataset size, there are only 3 common points, but the IGD is low; which implies points very close to the final SOIs are selected. From 70% onwards, all, or all but one points are common between the SOIs selected from the subset vs the final set.

# 4 RESULTS ON PRACTICAL PROBLEMS

In this section, we conduct the a posteriori SOI identification on two practical MaOPs of interest - hybrid car controller optimization [8, 19, 20, 24] and a building energy management optimization [22, 23].

## 4.1 Hybrid car controller optimization (HEV)

Hybrid cars have a combination of internal combustion engine and electric motor to in order to balance driving comfort and range with environmental considerations. An efficient controller is required to switch between these different sources of power during different driving scenarios so as to maximize the required performance objectives. The controller operates on a set of rules that define different driving conditions, and in the particular model under consideration

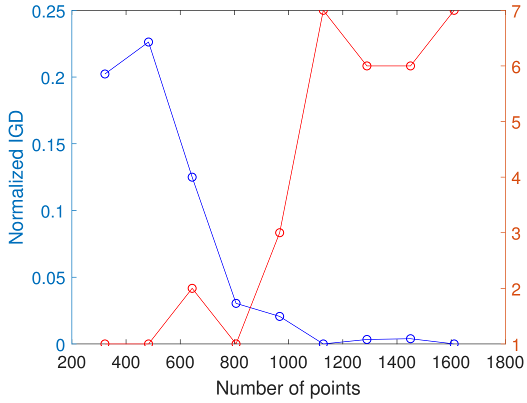



**Figure 4: The variation of IGD of the SOI with the change in datasize for DEB3DK problem**

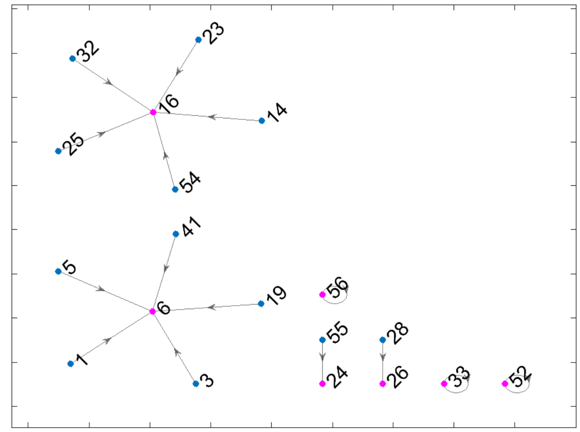


**Figure 5: The proximity graph for HEV problem**

comprises 11 tunable parameters. There are 7 objectives to be optimized (minimized), viz. fuel consumption, battery stress, operation changes, emissions, urban operation, noise, and battery state of health. A detailed discussion of the problem formulation can be found in [24].

The data for this study is obtained from a run of the reference vector guided evolutionary algorithm (RVEA) [7] on the HEV problem and contains 66 non-dominated solutions. The spacing parameter $p$ chosen for the problem is 7, resulting in 1716 weight vectors for $EMU^r$ calculations. The resulting SOIs identified by our approach are listed in Table 1. An interesting observation is that unlike the test problem considered previously which had some internal knees (by design), all the SOIs identified for the problem lie on the periphery of the PF approximation, with two of them being extremal, i.e., having at least one of the objectives at their best value.

**Table 1: All SOIs obtained for HEV problem** $p = 7, |W| = 1716$

| Id | $EMU^r$ | Category |
|---|---|---|
| 33 | 106.8122 | $\text{peripheral}_E$ |
| 24 | 73.0409 | peripheral |
| 6 | 75.2908 | peripheral |
| 26 | 0.5419 | peripheral |
| 16 | 73.6752 | peripheral |
| 52 | 85.5279 | $\text{peripheral}_E$ |
| 56 | 75.3269 | peripheral |

The corresponding proximity plot of the solutions in the objective space is shown in Fig 5, where the SOIs are shown in magenta color and the neighboring solutions in blue color. The directed arrow between any pair of solutions points towards the higher $EMU^r$ among them. It can be clearly seen from the plot that the identified SOIs (magenta dots) are either completely isolated (there are no solutions attached to the neighboring directions), or they always have a better $EMU^r$ value compared to the solutions attached to the neighboring non-empty reference directions.

The convergence of the IGD of selected SOIs for intermediate data subsets with respect to the SOIs selected from the final set is shown in Fig. 6. A clear difference can be observed in the trend compared to the case of the test problem in previous section in two aspects. Firstly, the IGD values are typically higher; which is reflective of the fact the points are relatively sparse given that it is a 7-objective space covered only by 66 points. Secondly, the SOIs selected from the intermediate subsets have only a small number of points in common (e.g. only 2 or 3 up to about 80% of the dataset) with the final SOI set. The SOIs get substantially similar only after 90% of the dataset is used. These trends indicate that a larger dataset would be helpful in obtaining a more stable set of SOIs for the problem; especially given the high-dimensionality of the problem. This could be enabled, for example, maintaining an archive of all solutions evaluated during the evolutionary search to construct a non-dominated set instead of considering only the final population.

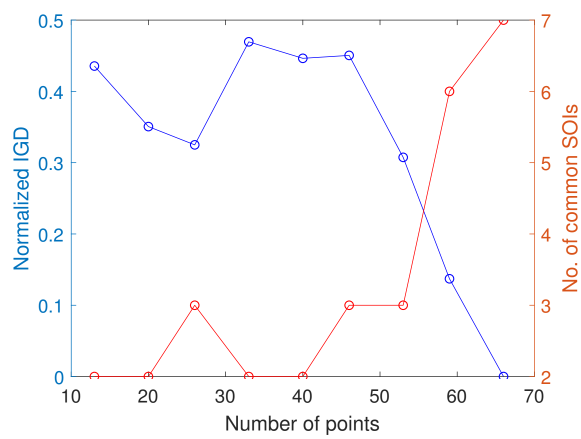



**Figure 6: The variation of IGD of the SOI with the change in datasize for HEV problem**

## 4.2 Building Energy Management (BEM) problem

Modern buildings and workplaces often consider in their design the integration of energy production, storage and consumption within their campuses. The decreasing costs of renewable energy generation (such as solar cells or wind energy) and storage (such as batteries) makes it an attractive option both from the economic investment and environmental perspective. The problem investigated herein is that of energy management in a research facility building. The problem has 8 design variables concerning the different parameters of the photovoltaic system, battery capacity, heat storage, etc. Four objectives are of interest: minimization of initial investment cost, annual operation cost, annual $CO_2$ emissions and maximization of resilience. The detailed formulation of the problem can be found in [22]. A extended version with 10 design variables and 5 objectives is published in [23].

The data for this study is obtained by running the $\mathcal{S}$-metric selection-Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [2] with 10,000 simulation calls and 100 individuals in the population. The final population of 100 non-dominated solutions is used for the a posteriori analysis. Since the number of objectives for this problem is lower than HEV, we can use higher value of the spacing parameter $p$ without resulting in excessively large reference vector set for $\text{EMU}^\text{r}$ calculations. For this study we consider $p = 15$, resulting in $|W| = 816$ weight vectors. The corresponding SOIs obtained are listed in Table 2. Four of the SOIs identified belong to the *Internal* category, with none of the objectives at their best value. The remaining three SOIs belong to $Peripheral_\text{E}$ category, exhibiting the best performance in at least one of the objectives.

**Table 2: All SOIs obtained for BEM problem ($p = 15, |W| = 816$)**

| Id | $\text{EMU}^\text{r}$ | Category |
|---|---|---|
| 12 | 54.3680 | internal |
| 2 | 54.2185 | internal |
| 9 | 39.6337 | internal |
| 78 | 55.4070 | internal |
| 1 | 156.5404 | $\text{peripheral}_\text{E}$ |
| 69 | 54.5904 | $\text{peripheral}_\text{E}$ |
| 13 | 55.0373 | $\text{peripheral}_\text{E}$ |

The corresponding proximity plots are presented in Fig. 7. Similar to the previous problem, it is observed that all the SOIs are either isolated in space (i.e., there are no solutions attached to their neighboring directions) or are strictly better then all neighboring solutions in terms of $\text{EMU}^\text{r}$.

Being a relatively low dimensional problem compared to HEV, as well as given that the non-dominated set considered has higher cardinality (hence covering the objective space more densely), the locations of the SOIs are observed to stabilize sooner. As shown in Fig. 8, the IGD values are in general lower than the HEV case; while at the same time the number of common solutions between the SOIs from subsets and the final set is in general higher. At about 80% and beyond, 6 of the identified SOIs are common, and the last remaining SOI is fairly close (given a low IGD value). Understandably, these values are not quite as low as that for the test problem (DEB3DK) considered earlier, given that a large number of data-points were available for the test problem. Once again, the use of archiving could be beneficial here to obtain more stable SOIs.

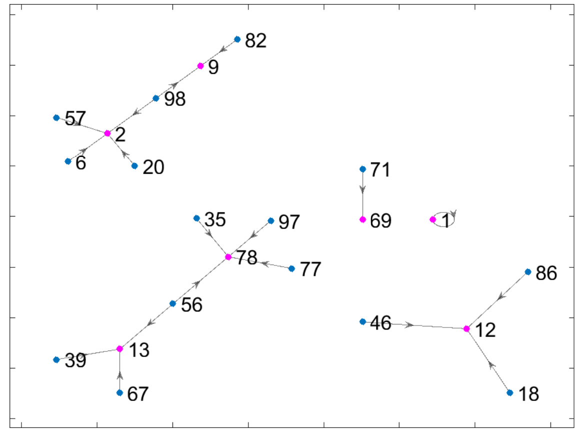



**Figure 7: Proximity graph for BEM problem**

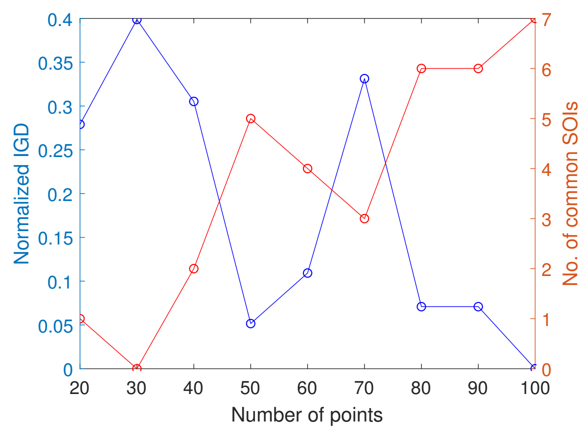



**Figure 8: The variation of IGD of the SOI with the change in datasize for building energy management problem**

## 5 CONCLUDING REMARKS AND FUTURE WORK

The research in many-objective optimization has gathered significant pace in the past few years. However, much of it is directed towards development of algorithms capable of delivering a set of converged and well distributed solutions. The next essential problem of selection of few designs out of this approximated set (which could often be large in size) has received relatively scarce attention. In this paper, we investigate the use of recursive expected marginal utility ($\text{EMU}^\text{r}$) to identify solutions of interest for informed decision making. The behavior of $\text{EMU}^\text{r}$ method is first demonstrated on a test problem with known knee solutions in order to establish its validity in identifying areas with desirable trade-offs. The method helps in assigning a complete order to the solutions, identifying given number of sparse SOI as well as classify them as internal, peripheral or extremal. This is illustrated through a number of different visualizations; for example the highest $\text{EMU}^\text{r}$ values on raw 3-objective data clearly show that the high values coincide with the knee regions. Through proximity plots, it is verified that the $\text{EMU}^\text{r}$ of any chosen SOI is higher than any of the solutions attached to its neighboring directions (or in some cases the SOIs cover isolated regions). Thereafter, the effect of datasize on the selection is investigated by taking progressively higher percentage of samples selected from the full dataset and observing the change in the selected solutions, using IGD as well as number of common solutions between the intermediate and final sets. The studies are then extended to two practical problems - a 7-objective hybrid electric vehicle controller and a 4-objective building energy management. The resulting SOIs and associated information and visualizations are presented in order to narrow down the design choices for a decision maker in a systematic manner.

A few different future directions are imminent following this work. First one is to enrich the information given to the SOI method in order to improve the possibilities of finding better SOIs. This can be done for example by providing all the non-dominated solutions instead of the final population. As it was seen, the chosen SOI tend to become less sensitive to the datasize if the density of the points in the space is sufficiently large (e.g. in the case of test problem). The second would be to consider the decision space in SOI selection; which would provide insights into the sensitivities of performance. When a large enough set (e.g., archive) is available, relatively accurate surrogate models can be built to conduct this analysis without excessive computational burden. Also, given that these studies were still in a proof of concept stage, explicit feedback from decision-makers was not considered in this study. When incorporated it could provide further insights and benchmarking of the method. Last but not the least, the method can be used interactively in order to focus on the knee regions of the PF instead of attempting to uncover solutions across the entire PF. These and other associated research directions will be investigated by the authors in the future work.

## ACKNOWLEDGMENT

The authors would like to acknowledge the financial support from Honda Research Institute Europe.

## REFERENCES


[1] Md Asafuddoula, Hemant K Singh, and Tapabrata Ray. 2015. Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 490–507.

[2] Nicola Beume, Boris Naujoks, and Michael Emmerich. 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (2007), 1653–1669.

[3] K. S. Bhattacharjee, H. K. Singh, and T. Ray. 2016. A study on performance metrics to identify solutions of interest from a trade-off set. In *Australasian Conference on Artificial Life and Computational Intelligence (ACALCI) (Lecture Notes in Computer Science)*, Vol. 9592. 66–77.

[4] Kalyan Shankar Bhattacharjee, Hemant Kumar Singh, and Tapabrata Ray. 2019. Optimum Wind Farm Layouts: A Many-Objective Perspective and Case Study. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 707–718.

[5] Kalyan Shankar Bhattacharjee, Hemant Kumar Singh, Michael Ryan, and Tapabrata Ray. 2017. Bridging the gap: Many-objective optimization and informed decision-making. *IEEE Transactions on Evolutionary Computation* 21, 5 (2017), 813–820.

[6] J. Branke, K. Deb, H. Dierolf, and M. Osswald. 2004. Finding knees in multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN) (Lecture Notes in Computer Science)*, Vol. 3242. Springer, 722–731.

[7] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. 2016. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 773–791.

[8] Ran Cheng, Tobias Rodemann, Michael Fischer, Markus Olhofer, and Yaochu Jin. 2017. Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. *IEEE Transactions on Emerging Topics in Computational Intelligence* 1, 2 (2017), 97–111.

[9] I. Das. 1999. On characterizing the knee of the Pareto curve based on normal-boundary intersection. *Structural Optimization* 18, 2 (1999), 107–115.

[10] I. Das and J. E. Dennis. 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657.

[11] K. Deb and S. Gupta. 2011. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering Optimization* 43, 11 (2011), 1175–1204.

[12] Peter J Fleming, Robin C Purshouse, and Robert J Lygoe. 2005. Many-objective optimization: An engineering design perspective. In *International conference on evolutionary multi-criterion optimization*. Springer, 14–32.

[13] David Hadka, Patrick M Reed, and Timothy W Simpson. 2012. Diagnostic assessment of the Borg MOEA for many-objective product family design problems. In *2012 IEEE Congress on Evolutionary Computation*. IEEE, 1–10.

[14] E. J. Hughes. 2007. Radar waveform optimisation as a many-objective application benchmark. In *Evolutionary Multi-Criterion Optimization (EMO) (Lecture Notes in Computer Science)*, Vol. 4403. Springer, 700–714.

[15] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. 2008. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 2419–2426.

[16] Antonio López Jaimes, Akira Oyama, and Kozo Fujii. 2013. Space trajectory design: Analysis of a real-world many-objective optimization problem. In *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2809–2816.

[17] Vineet Khare, Xin Yao, and Kalyanmoy Deb. 2003. Performance scaling of multi-objective evolutionary algorithms. In *EMO*, Vol. 2632. Springer, 376–390.

[18] Mario Köppen and Kaori Yoshida. 2007. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 727–741.

[19] Kaname Narukawa and Tobias Rodemann. 2012. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application. In *2012 Sixth International Conference on Genetic and Evolutionary Computing*. IEEE, 316–319.

[20] Yuka Ogino, Ryoya Iida, and Tobias Rodemann. 2017. Using desirability functions for many-objective optimization of a hybrid car controller. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 297–298.

[21] Robin C Purshouse and Peter J Fleming. 2003. Evolutionary many-objective optimisation: An exploratory analysis. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, Vol. 3. IEEE, 2066–2073.

[22] Tobias Rodemann. 2018. A many-objective configuration optimization for building energy management. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[23] Tobias Rodemann. 2019 (accepted). A Comparison of Different Many-Objective Optimization Algorithms for Energy System Optimization. In *Applications of Evolutionary Computation (LNCS 11454)*, P. Kaufmann and P.A. Castillo (Eds.). Springer, 1–16.

[24] Tobias Rodemann, Kaname Narukawa, Michael Fischer, and Mohammed Awada. 2015. Many-objective optimization of a hybrid car controller. In *European Conference on the Applications of Evolutionary Computation*. Springer, 593–603.

[25] Hiroyuki Sato, Hernán E Aguirre, and Kiyoshi Tanaka. 2007. Controlling dominance area of solutions and its impact on the performance of MOEAs. In *International conference on evolutionary multi-criterion optimization*. Springer, 5–20.

[26] Martin Schlueter, Chit Hong Yam, Takeshi Watanabe, and Akira Oyama. 2015. Many-objective optimization of interplanetary space mission trajectories. In *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 3256–3262.

[27] Hemant Kumar Singh, Kalyan Shankar Bhattacharjee, and Tapabrata Ray. 2018. Distance based subset selection for benchmarking in evolutionary multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation* (2018).

[28] Hemant Kumar Singh, Amitay Isaacs, and Tapabrata Ray. 2011. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation* 15, 4 (2011), 539–556.

[29] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. 2017. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation* 21, 3 (2017), 440–462.

[30] Guo Yu, Yaochu Jin, and Markus Olhofer. 2018. A Method for a Posteriori Identification of Knee Points Based on Solution Density. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[31] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[32] X. Zhang, Ye Tian, and Y. Jin. 2014. A knee point driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* PP, 99 (2014), 1–1.