2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Online Knowledge-based Evolutionary Multi-Objective Optimization

Bin Zhang, Kamran Shafi and Hussein Abbass

***Abstract*— Knowledge extraction from a multi-objective optimization process has important implications including a better understanding of the optimization process and the relationship between decision variables. The extant approaches, in this respect, rely on processing the post-optimization Pareto sets for automatic rule discovery using statistical or machine learning methods. However such approaches fall short of providing any information during the progress of the optimization process, which can be critical for decision analysis especially if the problem is dynamic. In this paper, we present a multi-objective optimization framework that uses a knowledge-based representation to search for patterns of Pareto optimal design variables instead of conventional point form solution search. The framework facilitates the online discovery of knowledge during the optimization process in the form of interpretable rules. The core contributing idea of our research is that we apply multi-objective evolutionary process on a population of bounding hypervolumes, or rules, instead of evolving individual point-based solutions. The framework is generic in a sense that any existing multi-objective optimization algorithm can be adapted to evaluate the rule quality based on the sampled solutions from the bounded space. An instantiation of the framework using hyperrectangular representation and non-dominated sorting based rule evaluation is presented in this paper. Experimental results on a specifically designed test function as well as some standard test functions are presented to demonstrate the working and convergence properties of our algorithm.**

I. Introduction

MANY real world problems are formulated as multi-objective optimization problems (MOP). Evolutionary Multi-objective Optimization Algorithms (MOEA) are arguably the most famous family of metaheuristcs for solving MOP, which aim at obtaining a set of diverse trade-off solutions with Pareto optimality [1]. However, presenting only a set of approximate, discrete and static optimal solutions is often not sufficient for decision makers in a multi-objective context. The decision makers are often trapped in a dilemma where no (or only a few) solutions are viable for implementation even though the so-called optimized result has been returned. In such situations, what interests the decision makers more is not only an optimized set of solutions but an understanding of the problem and an identification of patterns in the design space that lead to better solutions. This becomes even critical when there is a need to generalize optimization results for problems with similar design structures, or when the problem definitions change dynamically. Knowledge extraction or discovery from a multi-objective optimization process, thus, has important implications including better understanding of the optimization process as well as the relationship between decision variables.

Bin Zhang, Kamran Shafi and Hussein Abbass are with School of Engineering and Information Technology, University of New South Wales, Canberra, Australian Capital Territory, Australia. (email: Bin.Zhang@student.adfa.edu.au, {K.Shafi, H.Abbass}@adfa.edu.au).

One of the extant approaches for knowledge discovery in MOP rely on pre-optimization formulation and monotonicity analysis of the objective functions [2]. While these techniques can provide insights into the relationship between decision variables and optimal solutions, their use is limited due to the monotonic assumptions required for constraints and objective functions. On the other hand, recent research [7][4][5][6] has focused on devising techniques to automatically extract knowledge from post-optimization processing of the Pareto optimal solutions. The objective of these approaches is to determine important design rules by approximating the final Pareto set with respect to the decision variables using statistical and machine learning techniques.

There are two problems with the post-optimization approaches: First the knowledge discovery process does not start until the optimization process is completed and second no information is available about how the search has progressed during the optimization process. The first problem implicates higher computational time, a longer wait to obtain the extracted knowledge and hence waste of resources while the second problem relates to a poor understanding of the optimization process – a key motivation for carrying out the knowledge discovery process. The second problem is also restrictive for the dynamic optimization problems where the search or fitness landscapes may change over time and there is never a final Pareto set which might be post-processed for knowledge extraction and discovery. The parsing of Pareto set and application of regression techniques to approximate the Pareto front can also become a bottleneck for high-dimensional MOP as most simple regression techniques do not scale to high dimensional data spaces. Finally, the post-optimization extracted knowledge is useless for improving the optimization problem at hand.

In this paper, we present a multi-objective optimization framework that uses a knowledge-based representation to search Pareto optimal areas instead of individual solutions as is done traditionally in the MOEA research. To clarify, here we refer to *knowledge* as *patterns in the design space that lead to Pareto optimal solutions in the objective space*. In this context, the framework facilitates the online discovery of knowledge during the optimization process in the form of interpretable rules. The core contributing idea of our research is that we apply evolutionary process on a population of n-dimensional bounding hypervolumes (or rules), where n corresponds to the number of variables or dimensions in the design space. The rule population is then evolved towards Pareto optimal areas instead of evolving towards individual Pareto solutions. The fitness of rules is partly dependent upon the quality of the sampled solutions from the bounded volume of the rule according to a multi-objective criterion, such as non-dominated sorting, and partly on other characteristics, such as, the relative volume of the rules and proportion of good solutions. We refer to this framework as *Knowledge-Based Multi-objective Optimization Framework using Evolutionary Algorithms (KB-MOFEA)*.

KB-MOFEA is generic in a sense that any existing MOEA can be adapted to evaluate the rule quality in the objective space based on the sampled solutions from the bounded space represented by each rule. Moreover, the framework is representation independent in that different hypervolume representations can be used, including hyperrectangles which allow expressing rules in a simple if-then form. KB-MOFEA has a number of advantages including:

- It could ensure including all areas of the design space in the exploration process at initialization. This is especially useful for problems with large number of decision variables leading to sparse search spaces. Further, poor design areas are automatically excluded from the search process through evolution.
- It allows sampling all areas of the search space without being computationally too expensive. Once the optimal areas are discovered, they can be sampled infinitely to obtain a nice spread of solutions in the decision space.
- The multi-objective evolution of hypervolumes means that a set of rules are available at every step of the evolution capturing the current state of the optimization process. This is especially useful for the dynamic optimization problems where the decision makers might need a set of solutions and hence the knowledge about them at any point in time.
- The hypervolume representation, and in particular hyper-rectangular representation, provides a powerful and intuitive way of capturing knowledge, especially in high-dimensional search spaces.
- The rule-based representation of the optimal design space provides decision makers a greater flexibility in exercising their preferences.

In this paper, we present an instantiation of KB-MOFEA implemented with hyper-rectangular rule representation and non-dominated sorting based rule evaluation as used by NSGAII algorithm [8]. We term the resulting algorithm as *the Rule-Based NSGAII (RBNSGAII)*. To evaluate RBNSGAII, a simple bi-objective test function generator is developed with two decision variables. The function generator allows generating objective functions with multiple rectangular-shaped optimal areas in the design space. The experiments with a four-box test function validates the correct working of RBNSGAII as the algorithm nicely converge to the optimal areas with minimum number of rules. The experimental results with DTLZ functions also show that algorithm can successfully discover the optimal areas with minimum number of bounding rules.

It is important to note that our framework is different from the conventional evolutionary rule learning systems [10] that use MOEA to evolve classification rules [9][25][26][27]. The main purpose of the above class of systems is to learn rules for data classification during a training phase where the systems receive input data with class labels. The evolutionary algorithms are used to generate and refine these rules during training which are then used to classify future cases. There is no input data with correct labels in our context.

The rest of this paper is organized as following. The overall framework of knowledge-based multi-objective optimization is presented in Section III. The hyperrectangular rule-based implementation of the framework with non-dominated sorting based rule evaluation (RBNSGAII) is presented in Section IV. Section V reports on the evaluation of RBNSGAII and includes the details of the test functions, experimental setup and discussion of the results. The conclusions and future research directions are discussed in Section VI.

II. BACKGROUND

A. MOP and MOEA

A multi-objective optimization problem (MOP) typically contains a set of objective functions for minimization or maximization subject to a number of constraints. Formally an MOP can be defined as follows [11] (Minimization for instance):

$$\begin{array}{ll} \text{Minimize} & F(X) = (f_1(X), f_2(X), ..., f_m(X)) \\ \text{Subject to} & x \in \Omega \end{array} \qquad (1)$$

Here, solution X is a n-dimensional vector $X = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$. Ω is the feasible set of solutions restricted by equality or inequality constraints and variable bounds [12]. In real-world problems, the objectives $f_1, f_2, \cdots, f_m$ are not harmonious, but conflicting with each other, instead. Hence, there is no single solution that can optimize the set of objective functions simultaneously. Improvement of one objective will deteriorate at least one of the rests. The Pareto optimality is used to differentiate outstanding solutions from others [13]. We say solution X dominate Y if all $f_i(X) \leq f_i(Y)$ for $i = 1, 2, \cdots, m$ and at least one of them, say f_{i^*}, satisfies $f_{i^*}(X) < f_{i^*}(Y)$. A solution is Pareto optimal if it is not dominated by any other solution in the search space and the set of all these non-dominated solutions is referred to as the Pareto set (PS). While in objective space, the set of corresponding objective vectors $(F(X_1), F(X_2), \cdots)$ for a given Pareto set $(X_1, X_2, \cdots)$ is called the Pareto front (PF). For many MOPs, there are numerous or even infinite Pareto optimal solutions.

Obviously, the goal of multi-objective optimization is to identify the Pareto set and Pareto front for a given MOP. However, identifying the exact and entire Pareto set of a given MOP is often practically impossible due to the inherent non-linearity, complexity and comprehensibility of the problem and the enormous size of the search space. In this case, multi-objective evolutionary algorithms, especially the multi-objective genetic algorithms (MOGA), as a population-based, bio-inspired metaheuristic, emerged and gradually dominated the research in MOP as they are able to approximate the Pareto front (Pareto set) in a single run. Ever since the first MOGA, called VEGA [14], was proposed, many distinguished algorithms, such as NSGA-II [8], DMEA [15], MOEA/D [16], has been developed with remarkable performance improvements and applied to a huge variety of applications. For the sake of brevity, the detailed evolution process is not discussed here. Some dedicated reviews and surveys on MOPs and MOEAs are listed for reference purpose [11][12].

B. Knowledge discovery from MOPs

In the literature, there is already some research available regarding the knowledge extraction from multi-objective problems. The knowledge here is general, not constrained by the form of rules. The first source of knowledge is naturally and directly from the definition and formulation of specific multi-objective problem, such as monotonicity analysis [2]. It is a pre-optimization technique for investigation of important properties among decision variables and the optimal solutions when there are monotonic objective functions or constraint functions. Such methods [17][18] try to capture and utilize the mathematical characteristics of functions used to model the original problem to get some insights for the understanding of MOP. However, they are often restricted by strong conditions and not popularly used.

On the contrary, the mainstream of knowledge discovery from multi-objective optimization employs an extra analysis of the Pareto optimality after optimization. Deb et al. first created the concept of *innovization* (innovation through optimization) aimed to unveil innovative design principles by means of multiple conflicting objectives [19][20]. Then they proposed a post-optimization analysis framework for automated discovery of vital knowledge from Pareto optimal solutions [7] and successfully demonstrated the applicability of this framework with applications such as truss structure design [4][5]. They finally differentiate the knowledge to higher levels and lower levels [6]. In their methodology, the knowledge denotes hidden problem structure characteristics, such as the correlations between variables and objectives, sensitivity to variables or constraints and so on, and is formulated as polynomials. Then they exploited and optimized these polynomial relationships between variables and objective functions with another evolutionary process after optimization to original MOP under the belief that the optimal solutions satisfying strong relationships will cluster together and the dominating cluster reveals the design principle. However, polynomial relationships are not general patterns and the application seems to be restricted with one or two well designed examples.

Beside the mathematical form of knowledge, association rules are also targeted for extraction from the Pareto front by combining optimization and data mining techniques [21]. Not only does the form of knowledge vary, but also many techniques are involved for this topic. The rough set theory is employed for the knowledge discovery purposes in [22]. A self organizing map is used to visualize tradeoffs of Pareto solutions for data mining in order to find clusters with high correlations [3]. A new multi-objective genetic programming is also proposed for multi-objective design exploration [23].

All the approaches above tend to integrate data mining and machine learning techniques into the optimization process to obtain interesting knowledge. Learning abilities and evolutionary search both benefit each other. In this paper, we will adapt the evolutionary search for knowledge discovery with rule representation for the pervasive application of rules, which are considered as one of the highly usable and readable outputs of data mining.

III. KB-MOFEA

In this section, we present our proposed generic framework for knowledge-based multi-objective optimization – KB-MOFEA. An algorithmic description of KB-MOFEA is provided in Algorithm 1. The main components of the framework include: generation of an initial rule population using a preferred representation, e.g., hyper-rectangular representation, and according to a given initialization scheme such as purely random or grid-based initialization; sampling solutions from the bounded space of each rule in the population according to a sampling scheme, e.g., uniform random or based on a statistical measure; evaluation of sampled solution according to a given criterion, e.g., based on non-dominated sorting; evaluation of rules based on solution quality as well as other measures such as rule volume; generation of next population of rules using an evolutionary algorithm and rule adjustments such as merging of overlapping rules.

Algorithm 1 KB-MOFEA: Knowledge-Based Multi-objective Optimization Framework using Evolutionary Algorithms

```
Initialization: generate a population of rules covering the decision space according to a given representation and a given method;
while no stopping criterion is satisfied do
    Sampling: Select a given number of solutions from the space bounded by each rule in the population according to a given sampling scheme;
    Solution Evaluation: Evaluate all sampled solutions according to given criteria;
    Rule Evaluation: Evaluate all rules in the population as a function of sampled solutions quality and other rule characteristics;
    Rule Evolution: Apply an evolutionary algorithm to generate a new rule population;
    Re-evaluate parent population if needed;
    Prune rules if needed;
end while
return the final rule population bounding the Pareto optimal decision space;
```

IV. KB-NSGAII

KB-MOFEA can be instantiated with and utilize any point-based solution evaluation method as an underlying component, such as the non-dominated sorting operation in NSGA-II and the decomposition operation in MOEA/D, to name a few. This section presents a detailed implementation of RBNSGAII – an instantiation of KB-MOFEA using hyper-rectangular rule representation and NSGAII type rule evaluation. Algorithm 2 provides complete pseudocode of RBNSGAII. The following subsections further explain different operations in detail used in this implementation.

Algorithm 2 RBNSGAII

```
Initialize hyper-rectangular rule population P_1 of size N using a grid-based approach;
Sample S_R solutions using each rule in P_1 using Latin Hypercube method;
Rank all (N × S_R) sampled solutions using the non-dominated sorting;
Evaluate the quality of all the rules in P_1;
Initialize reference point ((ρ_ref, ν_ref)) using the best quality rule in P_1;
Compute rules fitness in P_1;
while gen ≤ Gen do
    Choose rules from P_1 for reproduction to form a parent population P_p using niched binary tournament selection;
    Perform real crossover and mutation on P_p to generate a new rule population P_2;
    Sample S_R solutions from each rule in P_2;
    Rank all (2 × N × S_R) solutions in population P_p and P_2 using the non-dominated sorting;
    Evaluate the quality of all the rules in P_p and P_2;
    Update the reference point;
    Calculate the fitness for rules in P_p and P_2;
    For each rule R_i in P_2, use it to replace the first rule R_j in P_p when R_j is in the same niche with R_i but with an inferior fitness;
    Shrink and resample solutions for each rule in P_p;
    P_1=P_p;
end while
```

A. Rule Representation and Initialization

The hyperrectangular rules use an interval-based representation. In specific, a rule consists of n intervals representing n decision variable using upper and lower bounds.

$$(x^{1,l}, x^{1,h}, x^{2,l}, x^{2,h}, \cdots, x^{n,l}, x^{n,h})$$

where $x^{i,l}$ and $x^{i,h}$ denote the lower and upper bounds of variable x^i, respectively.

The implementation of RBNSGAII in this paper uses a grid based initialization that allows covering the entire decision space systematically without leaving a gap or an overlap between initial rules. The choice of initialization might impact the algorithm performance to some degree e.g. in terms of convergence speed. However, this sensitivity analysis is not the subject of this paper and focus is on presenting the main idea.

B. Solution Sampling from Rules

While individual solutions can be sampled from the rules uniform randomly. This would require a large number of samples to get an accurate representation of the bounded area. A high number of samples however means high computational time in our framework. On the other hand fewer samples could lead to a biased sampling. So an effective sampling mechanism is important for the working of our algorithms. While sampling is a rich area of research, in this paper we chose to use Latin Hypercube sampling (LHS) [24] which allows stratified and steady sampling of the search space using a divide and sample approach. Future research will focus on studying the effect of different sampling mechanism, including adaptive sampling mechanisms, on the performance of the algorithms.

C. Rule Evaluation and Fitness

A rule is evaluated using two measures: the quality of solutions sampled from the evaluated rule compared to all sampled solutions and the volume or size of the rule. To measure solution quality, all solutions sampled from the rule population are ranked using non-dominated sorting procedure as used in NSGAII. Then the overall quality of sampled solutions ρ is measured as:

$$\rho = \sum_{e_{nd}} \alpha^{\frac{i}{S_R}} - \sum_{e_d} \beta \times (rank_j) \tag{2}$$

where e_{nd} and e_d refer to the number of non-dominated and dominated solutions, respectively, in the sampled solution pool with indices i and j iterating over each set. Non-dominated solutions are given $rank = 1$, whereas dominated solutions are ranked $rank = 2, 3, \cdots$. α and β are two arbitrary numbers that denote the magnitude with which a rule is rewarded or penalized for enclosing a non-dominated or a dominated solution within its bounds respectively. In the current implementation, α and β are both set to 10.

The second measure, size of the rule ν, is computed as the summation of interval ranges in each dimension.

$$\nu = \sum_{i=1}^{n} (x^{i,h} - x^{i,l}) \tag{3}$$

Instead of using a product function to compute the actual volume occupied by the rule, we use a summation function to reduce the impact of a few infinitely small interval lengths on the overall rule quality.

Finally the fitness of a rule is computed as:

$$fitness = e^{-\frac{(\rho-\rho_{ref})^2}{2}} e^{-\frac{(\nu-\nu_{ref})^2}{2}} \tag{4}$$

ρ_{ref} and ν_{ref} were mentioned in Algorithm 2 above. They are computed as a pair. ρ_{ref} corresponds to the best ρ value achieved by a rule during the evolution up to the current generation. ν_{ref} is computed as the max size of all rules with a ρ value equal to the current ρ_{ref}.

This fitness function serves two purposes, first it aims at preferring the rules which enclose more non-dominated solutions and second it encourages larger or more general rules over specific rules. Since we do not apriori know the exact size of the optimal design space, this fitness function uses a dynamic reference point in the hope of finding the optimal design space size during the search.

D. Rule Evolution

The next generation of rules involve choosing N parents and creating N children through crossover and mutation operators. To choose the parent population, P_p, a modified niched-binary tournament selection scheme with elitism is used. For every rule in the old population, P_1, another rule is chosen uniform randomly from the whole population. Then the two rules are checked if they belong to the same niche or not. Two rules belong to the same niche if

- none of them contain any non-dominated solutions;
- either of the two contain one or more sampled solutions which are non-dominated; or
- both contain one or more non-dominated solutions and they overlap;

If none of the above conditions are true, the rules are not considered to belong to the same niche. Next a rule with better fitness value is inserted in the parent population if the two rules belong to the same niche. Otherwise, the second randomly selected rule is discarded and the first rule is inserted into the parent population.

Once the parent population is created, children are created by selecting two rules from the parent population in sequence and crossing them over. Each child is then mutated with a given probability. Both crossover and mutation operators are adopted from [8]. A consistency check is performed to ensure that the bounds are within the limits and are logical. Once the new population, P_2, is created by creating N children, the fitness of rules in both populations are updated using procedure explained in the above section. Finally, the two populations P_p and P_2 are combined to generate the next generation population. The combination involves a similar procedure as the selection procedure explained above. Each rule in P_2 is compared with each rule in P_p and rules in P_p are replaced by rules in P_2 if they fall within the same niche as a rule in P_p and have better fitness.

E. Shrinking

A rule shrinking heuristic is adopted to improve the convergence speed of the algorithm. In shrinking the rules boundaries are adjusted to the minimum bounding hyperrectangles that cover their best ranked solutions globally (e.g., if a rule has five solutions with following global ranks: 2, 3, 3, 2, 4, then the shrinking operator will consider first and fourth solutions). The second best ranked solutions are considered if the rules only contain a single best ranked solution.

V. Experiments

This section covers the details of the test functions we used to evaluate the performance of our proposed algorithm RBNSGAII, parameter settings for the algorithm and the experimental results.

A. Test Problems

Two types of functions are used to evaluate the proposed RBNSGAII. The first of them, that we term as *HMOA* for *Hyperrectangular Multi-objective Optimal Area* test function, is specifically designed to evaluate the research question: *To what degree our proposed rule based algorithm can converge to the optimal area with a shape matching the rule representation?* In a sense, we used this test function to tune our algorithm. However, a formulation of these concepts is warranted and is left for the future work.

Formally, the objective of HMOA is to minimize

$$\begin{array}{l} f_1(X) = \sin(x_1 + \frac{\pi}{4}) + \sin y \\ f_2(X) = \cos(x_1 + \frac{\pi}{4}) + \sin y \end{array} \tag{5}$$

where y is given by:

$$\begin{cases} x_2 - n & n(2\pi+1) - \frac{\pi}{2} \le x_2 < n(2\pi+1) + \frac{3\pi}{2} \\ n2\pi + \frac{3\pi}{2} & n(2\pi+1) + \frac{3\pi}{2} \le x_2 < n(2\pi+1) + \frac{3\pi}{2} + 1 \end{cases} \tag{6}$$

when there exists $n \in \mathbb{Z}$.

In the implementation, we restrict the decision variables as $x_1 \in \left[-\frac{\pi}{4}, \frac{19\pi}{4}\right]$ and $x_2 \in [0, 5\pi]$. The interval of x_2 means only $n \in \{0, 1, 2\}$ are possible. Hence y equals

$$\begin{cases} x_2 & 0 \le x_2 < \frac{3\pi}{2} \\ \frac{3\pi}{2} & \frac{3\pi}{2} \le x_2 < \frac{3\pi}{2} + 1 \\ x_2 - 1 & \frac{3\pi}{2} + 1 \le x_2 < \frac{7\pi}{2} + 1 \\ \frac{7\pi}{2} & \frac{7\pi}{2} + 1 \le x_2 < \frac{7\pi}{2} + 2 \\ x_2 - 2 & \frac{7\pi}{2} + 2 \le x_2 < 5\pi \end{cases} \tag{7}$$

The optimal areas for the 2-dimensional HMOA are then identified as

$$\begin{array}{ll} x_1 \in \left[\frac{3\pi}{4}, \frac{5\pi}{4}\right] & x_2 \in \left[\frac{3\pi}{2}, \frac{3\pi}{2} + 1\right] \\ x_1 \in \left[\frac{3\pi}{4}, \frac{5\pi}{4}\right] & x_2 \in \left[\frac{7\pi}{2} + 1, \frac{7\pi}{2} + 2\right] \\ x_1 \in \left[\frac{11\pi}{4}, \frac{13\pi}{4}\right] & x_2 \in \left[\frac{3\pi}{2}, \frac{3\pi}{2} + 1\right] \\ x_1 \in \left[\frac{11\pi}{4}, \frac{13\pi}{4}\right] & x_2 \in \left[\frac{7\pi}{2} + 1, \frac{7\pi}{2} + 2\right] \end{array} \tag{8}$$

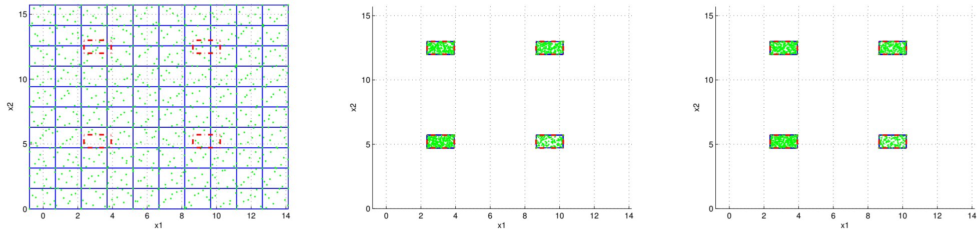
.

Next, we test our algorithms using the standard DTLZ1 functions.

$$\begin{array}{l} f_1(X) = \frac{1}{2} x_1 x_2 \dots x_{k-1}(1 + g(X)) \\ f_2(X) = \frac{1}{2} x_1 x_2 \dots (1 - x_{k-1})(1 + g(X)) \\ \dots \\ f_{k-1}(X) = \frac{1}{2} x_1 (1 - x_2)(1 + g(X)) \\ f_k(X) = \frac{1}{2}(1 - x_1)(1 + g(X)) \\ 0 \le x_i \le 1, i = 1, 2, \dots, n \end{array} \tag{9}$$

$g(X)$ is recommended as following:

$$\begin{array}{l} g(X) = 100 \times \\ \{(|X| - k + 1) + \\ \sum_{i=k}^{n} [(x_i - \frac{1}{2}) - \cos\left(20\pi\left(x_i - \frac{1}{2}\right)\right)]\} \end{array} \tag{10}$$



(a) Initialization (b) Gen=100 (c) Gen=2000

Fig. 1. Rule evolution in RBNSGAII over time when tested with HMOA function. The red dashed boxes show the optimal area of the design space. The rules are represented with solid line blue boxes and the green dots show the sampled solutions.

Theoretically, the set of Pareto solutions of DTLZ1 is $0 \leq x_i \leq 1, i = 1, 2, \ldots, k-1$ and $x_i = \frac{1}{2}, i = k, k+1, \ldots, n$ while the Pareto front is $\sum_1^k f_i = \frac{1}{2}$.

In specific, we used the 2D and 3D DTLZ1 functions listed below:

DTLZ1-2D:

$$\begin{aligned} f_1(X) &= \tfrac{1}{2}x_1(1+g(x_2)) \\ f_2(X) &= \tfrac{1}{2}(1-x_1)(1+g(x_2)) \end{aligned} \tag{11}$$

DTLZ1-3D:

$$\begin{aligned} f_1(X) &= \tfrac{1}{2}x_1x_2(1+g(x_3)) \\ f_2(X) &= \tfrac{1}{2}x_1(1-x_2)(1+g(x_3)) \\ f_3(X) &= \tfrac{1}{2}(1-x_1)(1+g(x_3)) \end{aligned} \tag{12}$$

The Pareto solutions of DTLZ1-2D are all located on the line segment $0 \leq x_1 \leq 1, x_2 = 0.5$ in the decision space and the Pareto front is $f_1 + f_2 = 0.5$. Accordingly, DTLZ1-3D has all the optimal solutions on a hyperplane $0 \leq x_1, x_2 \leq 1, x_3 = 0.5$ and the front is $f_1 + f_2 + f_3 = 0.5$.

B. Experimental Setup

Some of the common parameters for the experiments are listed in Table I.

TABLE I
COMMON PARAMETERS FOR RBNSGAII

Name	Value	Name	Value
Rule Pop Size	100	Sampling Size	10
Max Generation	2000(1000)	Reserved Digits	3
Crossover Rate	0.9	Mutation Rate	0.1

The evolution starts with 100 rules and grid-based initialization. In every evaluation, 10 solutions are sampled from every rule (i.e., a total of 1000 solutions for the whole population) using LHC sampling method explained above. Rules are real coded up to 3 digits after the decimal point.

C. Results

Figures 1 show the outcome of applying RBNSGAII to solve HMOA test function. First, Figure 1 shows the evolution of rules over time starting from the first to the last generation. Starting from a grid based initialization, one can note that the algorithm quickly converged to find the optimal design areas as early as generation 10. However, the optimal rules are found around generation 100. Although the algorithm converge quickly to the optimal area, there are still many overlapping rules in the population as demonstrated by the rule population listed in Table II. Since all these rules covers the optimal area they are all competitive and it would take a long time for evolution to remove such overlaps (see Table III). Obviously, the convergence to final optimal areas could be improved by introducing a rule pruning operators when population reaches a steady state as is shown by Figure 3. Figure 2 shows the comparison of true Pareto front for HMOA and solutions sampled from the rules of the final generation (gen=2000). As can be seen RBNSGAII is able to get a nice spread of the solutions owing to all 1000 solutions sampled from the optimal decision areas.

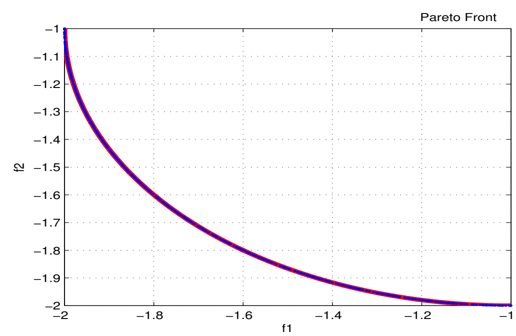


Fig. 2. The Pareto solutions obtained by RBNSGAII for HMOA function. The red solid line shows the true Pareto front and the blue dots shows the solutions sampled using the final generation rules.

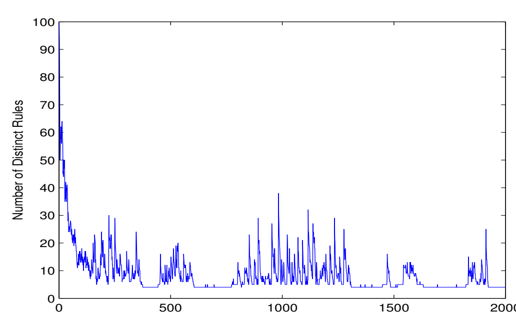


Fig. 3. The number of unique rules over time in the population for HMOA function.

TABLE II

DISTINCT RULES AT GENERATION 1000 FOR HMOA

$x^{1,l}$	$x^{1,h}$	$x^{2,l}$	$x^{2,h}$
2.357	3.927	4.713	5.712
8.64	10.21	4.713	5.712
8.64	10.21	11.996	12.995
2.357	3.926	4.713	5.712
2.357	3.926	11.996	12.995
2.357	3.886	4.713	5.712
2.357	3.877	4.713	5.701
2.357	3.85	4.713	5.597

TABLE III

DISTINCT RULES AT GENERATION 2000 FOR HMOA

$x^{1,l}$	$x^{1,h}$	$x^{2,l}$	$x^{2,h}$
2.357	3.926	4.713	5.712
8.64	10.21	4.713	5.712
8.64	10.21	11.996	12.995
2.357	3.926	11.996	12.995

Figures 4 and 5 repeat the same experiments as discussed above for HMOA test function for DTLZ1-2D and DTLZ1-3D respectively. For both functions the optimal design areas correspond to a single Pareto area. Subsequently, RBNSGAII converged much quickly than in the case of HMOA. Tables IV and Table V show the distinct rules in the final populations corresponding to the exact optimal areas in both cases.

TABLE IV

DISTINCT RULE AT GENERATION 1000 FOR DTLZ1-2D

$x^{1,l}$	$x^{1,h}$	$x^{2,l}$	$x^{2,h}$
0	1	0.5	0.5

TABLE V

DISTINCT RULE AT GENERATION 1000 FOR DTLZ1-3D

$x^{1,l}$	$x^{1,h}$	$x^{2,l}$	$x^{2,h}$	$x^{3,l}$	$x^{3,h}$
0	1	0	1	0.5	0.5

VI. CONCLUSION

This paper presents a novel knowledge-based multi-objective optimization framework which aims at searching for optimal areas of the design space instead of individual solutions using a rule-based representation. A population of rules, corresponding to the bounding areas in the design space, are evolved to search for the optimal areas. The rules are evaluated based on the quality of sampled solutions from their bounded area. The framework allows using any existing MOEA to be used for the quality of rule evaluation. This facilitates the online discovery of knowledge during the optimization process in an interpretable form.

An implementation of the framework using a hyperrectangular rule representation and NSGAII based rule evaluation is presented in this paper. The resulting algorithm is tested on a few standard as well as custom designed test functions.

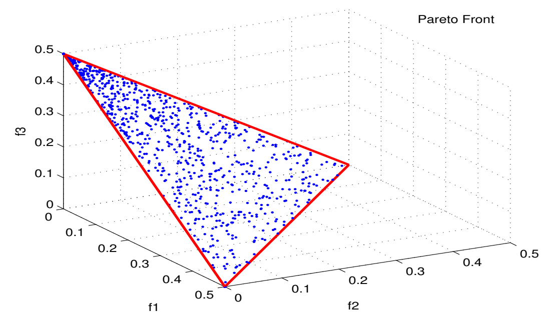


Fig. 6. The Pareto solutions obtained by RBNSGAII for DTLZ1-3D function. The red solid line shows the true Pareto front and the blue dots shows the solutions sampled using the final generation rules.

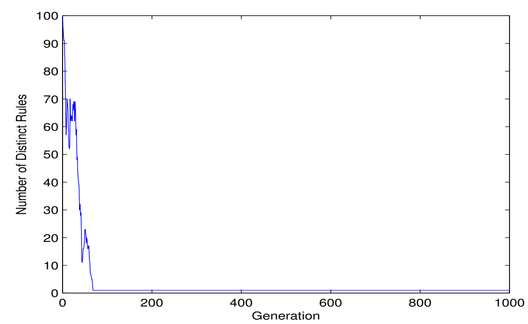


Fig. 7. The number of unique rules over time in the population for DTLZ1-3D function.

The experimental results show that the algorithm is able to successfully identify the optimal areas with minimum number of rules.

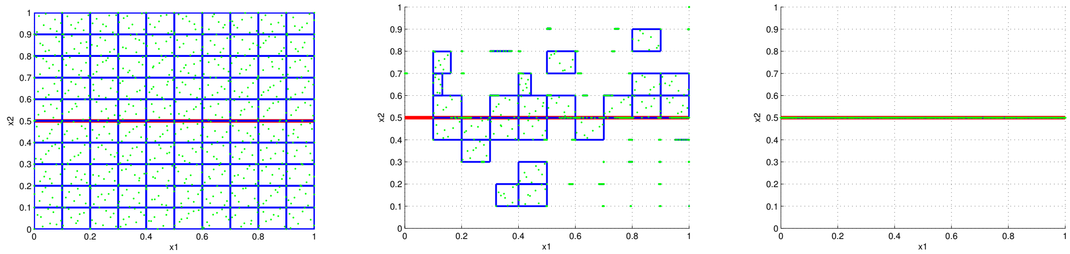
We believe that the knowledge-based MOEA framework presented in this paper has important implications for many domains including engineering design and decision support systems and has a broad scope for extensions. There are a number of directions stemming from this work for future research including extensions of the framework using a number of leading MOEA including SPEA and MOEA/D; extensions of the framework using a number of other rule representations including hyper-ellipsoids and fuzzy representations and thorough evaluation of the framework with complex optimization problems including those with non-linear, convex and dynamic objective functions under the above extension. The first two classes of extensions would establish the generality of the framework and the extensive testing mentioned above would be required to establish the effectiveness of the framework.

REFERENCES

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley, 2009.

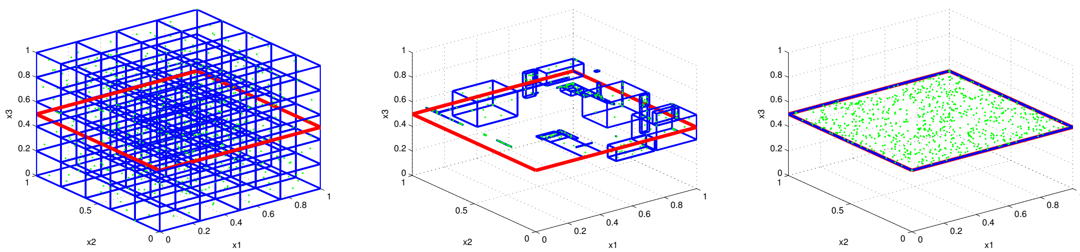
[2] P.Y. Papalambros and D.J. Wilde, *Principles of optimal design: modeling and computation*, Cambridge university press, 2000.

[3] S. Obayashi and D. Sasaki, "Visualization and data mining of Pareto solutions using self-organizing map," *Evolutionary multi-criterion optimization*, Springer, 2003.



(a) Initialization (b) Gen=5 (c) Gen=1000

Fig. 4. Rule evolution in RBNSGAII over time when tested with DTLZ1-2D function. The red dashed boxes show the optimal area of the design space. The rules are represented with solid line blue boxes and the green dots show the sampled solutions.



(a) Initialization (b) Gen=10 (c) Gen=1000

Fig. 5. Rule evolution in RBNSGAII over time when tested with DTLZ1-3D function. The red dashed boxes show the optimal area of the design space. The rules are represented with solid line blue boxes and the green dots show the sampled solutions.

[4] S. Bandaru and K. Deb, "Automated innovization for simultaneous discovery of multiple rules in bi-objective problems," *Evolutionary Multi-Criterion Optimization*, Springer, 2011.

[5] S. Bandaru and K. Deb, "Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique," *Engineering Optimization*, vol. 43, no. 9, pp. 911-941, 2011.

[6] S. Bandaru and K. Deb, "Higher and lower-level knowledge discovery from Pareto-optimal sets,". *Journal of Global Optimization*, pp. 1-18, 2013.

[7] S. Bandaru and K. Deb,"Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design," *2010 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2010.

[8] K. Deb, et al., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[9] S. Srinivasan and S. Ramakrishnan, "Evolutionary multi objective optimization for rule mining: a review," *Artificial Intelligence Review*, vol. 36, no. 3, pp. 205-248, 2011.

[10] A. Fernandez, et al., "Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 913-941, 2010.

[11] A. Zhou, et al., "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32-49, 2011.

[12] K. Deb, "Advances in evolutionary multi-objective optimization," *Search Based Software Engineering*, pp. 1-26, Springer, 2012.

[13] W. Stadler, "A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960," *Journal of Optimization Theory and Applications*, vol. 29, no. 1, pp. 1-52, 1979.

[14] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proceedings of the 1st international Conference on Genetic Algorithms*, L. Erlbaum Associates Inc., 1985.

[15] N. Long, L. T. Bui, and H. A. Abbass, "DMEA-II: the direction-based multi-objective evolutionary algorithm-II," *Soft Computing*, pp. 1-16, 2013.

[16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[17] B.C. Williams and J. Cagan, "Activity Analysis: Simplifying Optimal Design Problems Through Qualitative Partitioning," *Engineering Optimization*, vol. 27, no. 2, pp. 109-127, 1996.

[18] S. Sarkar, A. Dong, and J.S. Gero, "A learning and inference mechanism for design optimization problem (re)-formulation using singular value decomposition," *Proceedings of DETC*, 2008.

[19] K. Deb, "Unveiling innovative design principles by means of multiple conflicting objectives," *Engineering Optimization*, vol. 35, no. 5, pp. 445-470, 2003.

[20] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization," *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, ACM, 2006.

[21] K. Sugimura, et al., "Kriging-model-based multi-objective robust optimization and trade-off-rule mining using association rule with aspiration vector," *IEEE Congress on Evolutionary Computation CEC'09*, IEEE, 2009.

[22] K. Sugimura, S. Obayashi, and S. Jeong, "Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser," *ASME*, 2007.

[23] T. Tatsukawa, et al., "A New Multiobjective Genetic Programming for Extraction of Design Information from Non-dominated Solutions," *Evolutionary Multi-Criterion Optimization*, Springer, 2013.

[24] R. L. Iman. "Latin hypercube sampling," John Wiley & Sons, Ltd, 2008.

[25] J. Xiong, J. Liu, Y. Chen and H. A. Abbass, "A Knowledge-based Evolutionary Multi-objective Approach for Stochastic Extended Resource Investment Project Scheduling Problems," *IEEE Transactions on Evolutionary Computation*, Vol. PP, IEEE, 2013.

[26] K. Shafi, H. A. Abbass, and W. Zhu, "Real time signature extraction from a supervised classifier system," *IEEE Congress on Evolutionary Computation, 2007*, IEEE.

[27] K. Shafi, A. Bender and H. A. Abbass, "Multi objective learning classifier systems based hyperheuristics for modularised fleet mix problem," *Simulated Evolution and Learning*,pp. 381-390, Springer Berlin Heidelberg, 2012.