Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Tracking Variable Fitness Landscape in Dynamic Multi-objective Optimization using Adaptive Mutation and Crossover Operators

Ima O. Essiet[1], Yanxia Sun[1]

[1]Department of Electrical and Electronics Engineering Science, University of Johannesburg, South Africa

Corresponding author: Yanxia Sun (e-mail: ysun@uj.ac.za).

ABSTRACT Many real-world problems are modeled as multi-objective optimization problems whose optimal solutions change with time. These problems are commonly termed dynamic multi-objective optimization problems (DMOPs). One challenge associated with solving such problems is the fact that the Pareto front or Pareto set often changes too quickly. This means that the optimal solution set at period t may likely vary from that at *(t+1)*, and this makes the process of optimizing such problems computationally expensive to implement. This paper proposes the use of adaptive mutation and crossover operators for the non-dominated sorting genetic algorithm III (NSGA-III). The aim is to find solutions that can adapt to fitness changes in the objective function space over time. The proposed approach improves the capability of NSGA-III to solve multi-objective optimization problems with solutions that change quickly in both space and time. Results obtained show that this method of population reinitialization can effectively optimize selected benchmark dynamic problems. In addition, we test the capability of the proposed algorithm to select robust solutions over time. We recognize that DMOPs are characterized by rapidly changing optimal solutions. Therefore, we also test the ability of our proposed algorithm to handle these changes. This is achieved by evaluating its performance on selected robust optimization over time (ROOT) and robust Pareto-optimality over time (RPOOT) benchmark problems.

INDEX TERMS convergence, diversity, dynamic multi-objective optimization, inverted generational distance, reference points

I. INTRODUCTION

Many real-world problems are modeled by parameters which change over time [1-4]. Rather than model such problems as static optimization problems, they are best described using dynamic optimization problems. With respect to multiple objectives, dynamic problems are characterized by a moving, constantly changing Pareto front (PF) or Pareto set (PS). A dynamic multi-objective optimization problem (DMOP) is generally considered to be a changing sequence of multi-objective optimization problems [5]. Some real-world instances of DMOPs are solved in [6] and [7]. The fitness landscape of a DMOP is dynamic because of time-varying objective functions and/or constraints. In this paper, we consider a DMOP as consisting of multiple objective functions of the form:

$$min_x f(x,t) = \begin{bmatrix} f_1(x,t) \\ \cdot \\ \cdot \\ f_n(x,t) \end{bmatrix} \quad (1)$$

subject to:

$\mathbf{x} \in \beta(t)$

where $\beta(t)$ is the decision variable space. $\beta(t) \subset \Re^m$ where m is the number of feasible decision variables. The decision variables are constituted such that there are generally u inequality constraints and v equality constraints according to:

$$\beta(t) = x \in \mathbb{R}^m\colon a_u(x,t) \le 0 \; and \; b_v(x,t) = 0 \quad (2)$$
$$for \;\; u = 1..,k \; and \; v = 1..,l$$

With regard to the Pareto optimal front (POF), an objective function vector $\mathbf{c}_t$ dominates another vector $\mathbf{d}_t$ when $\mathbf{c}_t \le \mathbf{d}_t$. In other words:

$$\mathbf{c}_{j,t} \le \mathbf{d}_{j,t} \forall j = 1,2,......n \quad (3)$$

Therefore, all the elements in $\mathbf{c}_{j,t}$ constitute the non-dominated POF. The Pareto optimal solution (POS) set is obtained from the decision vector $\beta(t)$ in which there exists a subset of $\beta(t)$ called:

$$\mathbf{x}'(t) = \left[x_1'(t), x_2'(t).., x_f'(t)\right] \quad (4)$$

such that no other vector in $\beta(t)$ dominates $\mathbf{x}'(t)$.

One important point to note in the optimization of DMOPs is that a balance between convergence and diversity of the POS must be maintained despite the dynamic fitness landscape [8]. In other words, there must be a balance between the ability of a search algorithm to effectively explore the solution space without failing to select locally non-dominated solutions that may eventually constitute the POS or POF. One such approach is to effectively reinitialize the search population after a change has occurred within the solution space over time. From literature, one technique of ensuring that this happens is by adapting the search algorithm's mutation and crossover operators [5]. The crossover operator is used to search for feasible solutions within the solution space. Mutation operator alters the genes of the offspring. In other words, it improves fitness level of candidates within the solution space. Therefore, these two operators can significantly improve the quality of the PF in dynamic optimization problems if they are well tuned.

In this paper, we propose the use of adaptive crossover and mutation operators for the NSGA-III evolutionary algorithm in the presence of time-varying objective functions. We call this dynamic version of NSGA-III, dynamic NSGA-III (dyNSGA-III). The aim of this proposed approach is to address the problem of high-dimensionality, or changing dimensionality associated with DMOPs. The main contributions of this paper include:

- The proposal of a principal component analysis (PCA) strategy with *n*-mutation to improve selection pressure and diversity of original NSGA-III under high-dimensionality.
- Testing capability of proposed strategy to obtain robust solutions to problems with multiple time-varying PF and PS.

The results of the proposed approach would be compared to the performance of three other well-performing dynamic multi-objective evolutionary algorithms (DMOEAs) in terms of sensitivity to changing fitness landscape. We use the robust moving peaks benchmark (RMPB) test suite for single objective and robust Pareto-optimality over time (RPOOT) for up to three objectives. The rest of the paper is organized as follows: Section II discusses successfully implemented strategies for tackling DMOPs. Section III presents the proposed approach to effectively track the moving POF characterizing DMOPs. Section IV presents the test problems and performance parameters for estimating the ability of the proposed algorithm to retain robust solutions. Section V discusses results obtained and Section VI concludes the paper.

II. STRATEGIES FOR IMPROVING DYNAMIC MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Several researchers have proposed improvements to the capability of EAs to handle dynamic, multimodal problems. However, several challenges remain [9]. One of these challenges is difficulty in establishing the right balance between convergence and diversity. Convergence describes the ability of an MOEA to settle on the final non-dominated solution or set of solutions that most accurately solves the problem. Diversity is concerned with the spread of non-dominated solutions within the solution space. This problem is considered by many researchers to be the most significant for MOEAs. Another problem is as the number of objectives increases, the distance between suitable mates also increases. This means that when such candidates mate, their offspring will be far away from both mates. This means that more effort would have to be made to recombine candidates to ensure effective convergence and diversity of the non-dominated solution set. A third challenge is when the number of objectives exceeds two, it becomes challenging to estimate the performance of MOEAs due to high dimensionality of the solution space. Therefore, it becomes difficult to compare the performance of several MOEAs. Examples of most used performance metrics include inverted generational distance (IGD) and hypervolume (HV) metrics. Lastly, when the number of objectives describing a problem is large (typically beyond five), it becomes difficult to visualize the non-dominated solution space.

In recent years, several approaches have been proposed by various researchers to improve the optimization of DMOPs. In [10], a prediction strategy based on reference points was used to partition the population into subpopulations. With this strategy, a sequence of subpopulation centers in the previous environments was attached to one reference point and was used to predict the center of the new environment. The performance of the proposed algorithm was compared to population prediction strategy (PPS) and dynamic search strategy (DSS) for various dynamic environments. The proposed algorithm outperformed the other two algorithms for highly dynamic problems. In [11], a squirrel search algorithm was proposed to optimize DMOPs. This algorithm was based on decomposition with both evolutionary direction prediction and bidirectional memory populations. The proposed direction prediction strategy involved the use of modification vectors and judgment individuals. Therefore, there was no need for threshold settings. In terms of the bidirectional memory populations, two cases were considered: when the population evolved along the evolutionary direction of the previous environment, and when it evolved against the previous evolutionary direction. The proposed algorithm's performance was compared with

3 other well-performing dynamic optimization algorithms for 6 dynamic multi-objective optimization problems. The test parameters considered involved both severity and frequency of change within the solution space. The proposed algorithm outperformed the other three dynamic optimization algorithms.

In [12], a strategy was proposed to predict the dynamic location of the Pareto set by clustering the population into several representative groups. The number of clusters was adapted to the intensity of environmental change. The proposed method was compared with 5 other well-performing algorithms including multi-objective evolutionary algorithm based on dominance (MOEAD) and dynamic non-dominated sorting genetic algorithm (dNSGA-II). The algorithms were tested on 10 DMOPs; performance indices included spacing metric, dynamic hypervolume metric and mean inverted generational distance. The proposed algorithm outperformed the other algorithms, particularly problems involving rotating Pareto set with dynamic environments.

In [13], a novel raccoon family optimization (RFO) algorithm was proposed to solve the task of planning and scheduling of multiple projects. The objective of the mathematical model was to maximize total net profit of the multiple projects while considering early completion bonus, late completion penalty cost and resource cost. The performance of RFO algorithm was found to be superior to raccoon optimization algorithm (ROA), artificial bee colony (ABC) algorithm and genetic algorithm (GA) at 95% confidence interval. A Pareto-based guided artificial bee colony (PGABC) algorithm was proposed to handle the multi-objective optimization problem of lotsizing and mixed model scheduling for flexible production lines in [14]. Taguchi method was used to tune the parameters for PGABC and the algorithm was tested on 9 different problem instances based on demand and production system. Three conflicting objectives were considered: lotsizing and mixed model sequencing to minimize makespan, balancing workload among parallel production lines, and maximizing net profit in production lines. PGABC outperformed multi-objective artificial bee colony (MOABC), non-dominated sorting genetic algorithm III (NSGA-III) and improved strength Pareto evolutionary algorithm (SPEA2) for Pareto front (PF) improvement. Particularly, performance metrics were PF spread and proximity to true PF that optimized the problem. In [15], a hybrid spider monkey optimization (HSMO) algorithm was proposed to solve an integrated planning and scheduling problem for printed circuit board (PCB) assembly lines. The multi-objective problem involved line assignment to PCB models, component allocation to machines, and component placement sequencing to maximize net profit. The performance of HSMO was compared to ABC, GA, particle swarm optimization (PSO) and simulated annealing (SA) algorithms on a real-world problem adapted from a PCB manufacturing industry in China. It was reported that HSMO achieved near-optimal solutions compared to the other 4 algorithms.

Several other innovations with respect to solving dynamic optimization problems can also be found in [16-19]. In [16], a single randomly mutating time-variant archive was used to balance convergence and diversity of the dynamic Pareto front (PF). The Gee-Tan-Abbass (GTA) test suite was used to evaluate performance on problem dimensions of 500 and 1500, respectively. Local fitness approximation and prediction approach was used in [17] to locate optimal solutions to DMOPs. In this approach, the solutions that were sought were those that changed slowly over time. Adaptive gradient refinement based on a multi-layer co-evolutionary approach was proposed in [18]. This approach attempted to solve the problem of rapidly changing PF associated with finding suitable solutions to selected DMOPs. Maintaining both diversity and fitness of solutions in time-varying search environment was emphasized. In [19], a two-archive evolutionary algorithm was proposed to tackle the problem of changing shape of PF in the presence of changing number of objectives. One archive tackled convergence of the PF, while the other maintained its diversity.

In this paper, we propose the use of adaptive crossover and mutation operators to improve the capability of the NSGA-III algorithm to handle dynamic Pareto sets. We will test the ability of the proposed dyNSGA-III to maintain fit and robust solutions as the candidate landscape changes over time. The methodology for achieving this is discussed in the next section.

III. PROPOSED METHODOLOGY

The non-dominated sorting genetic algorithm III (NSGA-III) is capable of handling optimization problems with many objectives. It does this by using reference points that are positioned in the hyperplane representing the multi-objective optimization problem [20-21]. NSGA-III solves both constrained and unconstrained optimization problems. Regarding constrained problems, NSGA-III effectively handles three scenarios: when the Pareto front is optimal with an infeasible barrier, when the Pareto front is partly infeasible, and when it is wholly infeasible [21]. These capabilities determined our decision to improve the performance of NSGA-III to handle dynamic multi-objective optimization problems. Several techniques of mutation and crossover operator tuning have been reported [19]. We will focus on adaptive principal component analysis (PCA) mutation and *n*-point crossover, respectively. PCA is suitable for problems with high dimensionality because it selects features with highest covariance. This means that correlation among similar features in the problem variables is reduced [22]. The *n*-point crossover approach is selected to increase the randomness with which offspring for successive generations are generated [23]. This results in dimensionality reduction of the given problem, which improves visualization of the solution space. A real-coded genetic algorithm (GA) generally uses a population comprising N chromosomes, with each chromosome comprising M genes. The chromosome population is represented as:

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ . \\ . \\ c_N \end{bmatrix} \tag{5}$$

In general, each chromosome c_{ab} (where a and b denote chromosome position and generation respectively) is composed of genes g_{ab} such that $g_{ab} \in \lfloor g_{l,b}\ , g_{u,b} \rfloor$. $g_{l,b}$ and $g_{u,b}$ are the real-coded lower and upper bounds of the search space respectively. The PCA mutation approach involves the compression of high-dimensional input data into a lower dimensional space [19].

The input data set represented by Equation (5) can be viewed as a collection of N points within an m-dimensional Euclidean solution space equidistant from the mean value $E(\mathbf{c})$. The PCA mutation then looks for solutions in directions of maximum variance. The covariance matrix is obtained according to:

$$COV = E\{(\mathbf{c} - E(\mathbf{c}))(\mathbf{c} - E(\mathbf{c})^T)\} \tag{6}$$

Eigenvalues (γ_j) and eigenvectors (θ_j) are obtained according to:

$$COV\theta_j = \gamma_j \theta_j \quad \forall j = 1, \ldots m \tag{7}$$

The solution space is then re-centered using the PCA mutated eigenvectors to obtain the modified population as:

$$\mathbf{P} = (\mathbf{c} - E(\mathbf{c})) \cdot \boldsymbol{\theta} \tag{8}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_j]$ is the matrix of eigenvectors. The algorithm for the adaptive mutation of the solution space using PCA is shown in Algorithm 1.

The crossover operator enables offspring of the mutated chromosomes to find suitable solutions within the search space, varied according to the distance of potential solutions from the optimum solution(s). The n-point crossover (recombination) has been selected so that the number of genes for recombination can be varied according to the distance of potential solutions from the optimum solution(s). For n-point crossover, n genes are randomly selected from both parents. These are swapped between the parents and then they are recombined to produce the offspring. In this paper, we consider the single arithmetic recombination strategy [24]. In this method, the arithmetic average of both parent chromosomes is taken at a random point at which an allele is located. This is then used to create an offspring according to:

$$ch_{offspring} = x_1, \ldots, x_{l-1}, \eta \cdot y_l + (1-\eta) \cdot x_l, x_{l+1}, \ldots, x \tag{9}$$

where x_n and y_n are genes from parents, η is recombination operator.

The algorithm for describing the adaptive crossover operation is detailed in Algorithm 2.

The search parameters for the proposed dyNSGA-III algorithm are specified in Table 1. The test problems used to test the performance of the proposed algorithm are from the Gee-Tan-Abbass (GTA) benchmark test suite for dynamic multi-objective optimization [25]. Five additive and five multiplicative forms of the DMOPs were used to test the performance of dyNSGA-III. GTA1a – 6m are bi-objective DMOPs, while GTA9m – 12m are tri-objective. The main differences between the additive and multiplicative forms of the DMOPs are in terms of their lower bounds and modality [25].

Algorithm 1 Adaptive Mutation for dyNSGA-III

Start
Initialize pop. size (N_p), reference points (p_r), mutation probability (p_m)
While (stopping criterion*=false)
Adaptively vary p_m and measure Euclidean distance from $E(\boldsymbol{c})$
Output COV, θ_j, γ_j, $\mathbf{P}$
End

Algorithm 2 Adaptive Recombination for dyNSGA-III

Start
Initialize pop. size (N_p), reference points (p_r), crossover rate (c_r)
While (stopping criterion*=false)
Adaptively vary c_r and modify offspring chromosomes according to Equation (9)
Output best local $ch_{offspring}$
End

*For Algorithms 1 and 2, the stopping criterion is taken to be maximum Euclidean distance with respect to N_p and COV.

TABLE I PARAMETERS FOR DYNSGA-III ALGORITHM

Parameter	Setting
Severity of change (s_t)	5 (initial)
Frequency of change (f_t)	10 (initial)
Number of dimensions	50
Number of reference points (p_r)	21
Population size (N_p)	100
Mutation probability (p_m)	0.1
Crossover rate (c_r)	0.5 (adaptive)

There is a degradation of selection pressure of MOEAs as the number of objectives increases. dyNSGA-III remedies this problem by adopting the strategy proposed in [26]. Here, a less computationally expensive approach is used to generate reference points for the m-dimensional vector $\boldsymbol{\Gamma}_j$ using a random function. Also, a clustering operator is used to obtain a normalized PF based on the following:

$$d_o(x) = \left\|\mathbf{v}(\mathrm{x})^T \boldsymbol{\Gamma}_j\right\| / \left\|\boldsymbol{\Gamma}_j\right\| \tag{10}$$

$$d_p(x) = \left\|\mathbf{v}(\mathrm{x}) - d_o(x)\left(\frac{\boldsymbol{\Gamma}_j}{\|\boldsymbol{\Gamma}_j\|}\right)\right\| \tag{11}$$

where $\boldsymbol{\Gamma}_j = [\gamma_{j,1}, \gamma_{j,2}, \ldots \ldots, \gamma_{j,m}]$, $\mathbf{v}(\mathrm{x})$ is the normalized objective vector, $d_o(x)$ is distance between a line passing through the origin and a point p in the objective function space, $d_p(x)$ is perpendicular distance between $\mathbf{v}(\mathrm{x})$ and the line through the origin.

To illustrate the use of PCA to improve solution diversity of the PF, consider a hypothetical 2-objective hyperplane in

Fig. 1. The line through the origin with least varying solutions is eliminated while a line with greater variance with respect to potential solutions for the PF is selected.

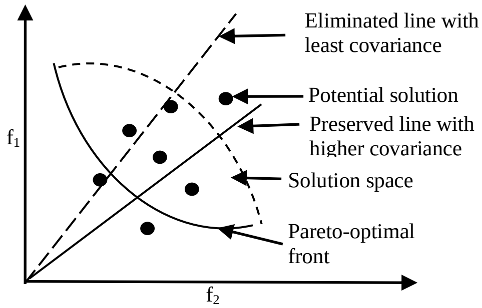

FIGURE 1. PCA selection of line with potential solutions.

This approach significantly improves selection pressure of the original NSGA-III by balancing convergence and diversity as the number of objectives increases. Therefore, PCA handles perturbation of the population and improves dyNSGA-III performance on RPOOT problems.

IV. PERFORMANCE WITH ROBUST OPTIMIZATION OVER TIME (ROOT) AND ROBUST PARETO-OPTIMALITY OVER TIME (RPOOT) BENCHMARK PROBLEMS

The idea of ROOT and RPOOT is because certain solutions in dynamic environments become sub-optimal as the search space changes over time. This means that a solution that optimizes problem parameters at a given feature evaluation instance may not do so in the next instance. This scenario is likely to be computationally expensive in real-life applications since the DMOEA parameters would need to be reconfigured to suit the new problem landscape. Therefore, it is important to test the ability of the DMOEA to obtain robust solutions which can optimally solve a DMOP over at least two feature instances of the DMOP [27]. In this section, we will examine the ability of the dyNSGA-III algorithm to obtain robust solutions which remain optimal over time. The performance of the proposed algorithm would be compared with the dNSGA-II, SGEA and dMOPSO DMOEAs.

Without loss of generality, we assume according to [27] that a solution or sequence of solutions is robust when such solutions can be used to solve a DMOP for at least 2 consecutive instances of the problem. Therefore, due to the characteristics of ROOT and RPOOT, there are specified benchmark problems which are used to test the robustness of the DMOEA. These are presented in the next subsection.

A. ENHANCING DYNSGA-III PERFORMANCE TO HANDLE ROOT AND RPOOT BENCHMARK PROBLEMS

With ROOT, solutions are selected in such a way that they remain effective even as environmental changes in the solution space take place over time. This is a more realistic approach to solving dynamic optimization problems as opposed to continually tracking a single global moving optimum. This therefore means that we must incorporate prediction into the solution-finding process based on the time-changing solution space. Regarding the benchmark problems being considered, we adopt the approach in [17]. In this approach, we use a radial basis function (RBF) model as an approximator, and an autoregression (AR) model as a predictor. We note that ROOT problems are to test the capability of dyNSGA-III to provide robust solutions for a single objective, while RPOOT problems test robustness for up to three objectives at a time. Details of the approach can be seen in [17].

B. ROOT BENCHMARK PROBLEMS

The problems used to test the robustness of the proposed DMOEA are based on [28]. The problems are collectively referred to as robust moving peaks benchmark (RMPB) and they are divided into two subsets. The first problem subset (RMPB-I) consists of six test problems which describe the maximization of the average fitness of solutions obtained by the DMOEA with respect to ROOT. The second subset (RMPB-II) consists of another six test problems which is used to determine the maximum time that a solution or solution sequence remains the optimal solution of the DMOP. According to [29], the average fitness and survival time are described according to Equations (12) and (13).

$$\rho_a(\mathbf{x},t) = \frac{1}{T}\sum_{j=0}^{T-1} f_{a,t+j}(\mathbf{x}) \tag{12}$$

$$\rho_s(\mathbf{x},t) = \begin{cases} 0 \quad if \;\; f_{a,t}(\mathbf{x}) < F \\ 1+\max\{\, p, \forall\; j \in (t, t+1, \ldots\ldots. t+p)\} \quad if \;\; f_{a,t}(\mathbf{x}) \geq F \end{cases} \tag{13}$$

where T is the DOP lifecycle, $f_{a,t}(\mathbf{x})$ is the average fitness of a solution describing the parameter vector $\mathbf{x}$, F is the solution fitness threshold, p is the time step increase over the interval $[0,T]$.

The baseline fitness function in RMPB-I for average fitness maximization is specified according to [28] as:

$$\rho_{f,t}(\mathbf{x}) = \frac{1}{D}\sum_{i=1}^{D} \max\{\, h_t^{ab} - w_t^{ab} * \left| x_b - c_t^{ab} \right| \} \tag{14}$$

Also, the baseline fitness function for RMPB-II for survival time maximization is according to Fu, [28] as:

$$\rho_{s,t}(\mathbf{x}) = \min_{b=1}^{n}\{\max_{a=1}^{m}\{h_t^{ab} - w_t^{ab} * \left| x_b - c_t^{ab} \right|\}\} \tag{15}$$

D is the dimension of the parameter vector $\mathbf{x}$, while h_t^{ab}, w_t^{ab} and c_t^{ab} represent the height, width and centre of peak function a for dimension b at time t. Varying the height, width and centre of a given peak function introduces dynamics into the solution space over time, which represent changes in the parameter space for many real-life problems. Regarding RMPB-I and –II, six different dynamics are considered which include: small step, large step, random, chaotic, recurrent, and recurrent with noise dynamics [28].

C. ROOT PERFORMANCE EVALUATION METRICS

We consider performance of individual solutions obtained by the DMOEAs being compared using the average error e_{av} and sensitivity to changing environment θ_s [27]:

$$e_{av} = \frac{1}{N_a} \sum_{b=p_0}^{p_0+N_a-1} \left|\rho_b - \rho_i\right| \quad (16)$$

where N_a is total number of problem instances for a given peak function a, p_0 is the initial problem instance, ρ_b is the fitness value of the global optimum of the problem instance b, ρ_i is the fitness value of solution S_i.

$$\theta_s = \sqrt{\frac{1}{N_a - 1} \sum_{b=p_0}^{p_o+N_a-1} \left(\left|\rho_b - \rho_i\right| - e_{av}\right)^2} \quad (17)$$

The fitness of a sequence of robust solutions is determined by considering the length of the sequence solutions (l_s) with respect to best sequence error ($e_{best,s}$), average sequence error ($e_{av,s}$), worst sequence error ($e_{worst,s}$) and sequence sensitivity (Θ_s). These parameters are defined according to [27]:

$$e_{best,s} = \min_{i=1}^{j} e_i \quad (18)$$

$$e_{av,s} = \frac{1}{j} \sum_{i=1}^{j} e_i \quad (19)$$

$$e_{worst,s} = \max_{i=1}^{j} e_i \quad (20)$$

$$\Theta_s = \sqrt{\frac{1}{j-1} \sum_{i=1}^{j} (e_i - e_{av,s})^2} \quad (21)$$

where e_i is average error of the solution sequence S_i.

In general, the performance measure of the DMOEA with respect to ROOT is estimated using the relation according to [28]:

$$Performance = \frac{1}{N} \sum_{i=1}^{N} F(x_i) \quad (22)$$

where $F(x_i)$ is the robustness of the solution sequence S_i. In terms of the robustness estimation, the T metric measures average fitness with respect to a time window, while the V metric measures survival time with respect to a fitness threshold. In this paper, T is set to 10, while V is set to 20.

D. PERFORMANCE OF DYNSGA-III ON RPOOT PROBLEMS

We further investigate the capability of dyNSGA-III to handle robust Pareto-optimality over time (RPOOT) for multiple objectives [30]. We consider test instances of dynamic multi-objective benchmark functions. We test based on 4 classic test instances (F1-F4) [31] and 3 instances for complicated Pareto front (F9-F11) [32]. According to [29], we use performance metrics of robust survival time (t_r), robust inverted generational distance (d_r) and robust spacing (s_r). Details of these metrics are as follows:

$$t_r = \frac{1}{T} \sum_{i=1}^{T} t_i \quad (23)$$

where T is the length of the survival time window for Pareto-optimal solutions and t_i is the survival time for the i-th robust Pareto-optimal solution. A larger t_r indicates that the solutions adapt better to changing environments over time.

$$d_r = \frac{1}{M} \sum_{i=1}^{M} max_{j=k_{i_1},\ldots\ldots,k_i+M_{k_i}} IGD(j) \quad (24)$$

where M is the total number of robust Pareto-optimal solutions, and $max_{j=k_{i_1},\ldots\ldots,k_i+M_{k_i}} IGD(j)$ is the inverted generational distance of the j-th robust solution. d_r measures the average inverted generational distance of robust Pareto-optimal solutions over their survival time.

$$s_r = \frac{1}{M} \sum_{i=1}^{M} \left(\frac{1}{|\Delta(i)|-1} \sum_{j=1}^{|\Delta(i)|} (d - d_j)^2\right)^{1/2} \quad (25)$$

where d_j is minimum Euclidean distance between fitness values of j-th robust solution in robust Pareto front ($\Delta(i)$) and true Pareto front, and $d = \frac{1}{|\Delta(i)|} \sum_{j=1}^{|\Delta(i)|} d_j$. Robust spacing measures average distribution of the Pareto-optimal solution set. For the simulation, population size was set to 100 for F1-F4 and 500 for F9-F11 [29]. Threshold was set to η=0.4 and time window was T=2. Simulations were run 20 times for dyNSGA-III, SGEA and dMOPSO.

V. DISCUSSION OF RESULTS

The performance of the proposed dyNSGA-III algorithm was compared with three other well-performing dynamic multi-objective evolutionary algorithms (DMOEAs), namely: dynamic non-dominated sorting genetic algorithm II (dNSGA-II), multi-objective particle swarm optimization based on decomposition (dMOPSO) and steady-state and generational evolutionary algorithm (SGEA). Performance metrics considered are the hypervolume (HV) and the mean inverted generational distance (MIGD) (Table II and III respectively). The results are the mean values with the standard deviation in parentheses. Results in boldface indicate the best-performing algorithm. The additive and multiplicative forms of the DMOPs from the GTA test suite were used to test the DMOEAs. The severity (s_t) and frequency (f_t) of the dynamic search space were tuned in such a way that we consider the behavior of the DMOEAs for cases where $s_t < f_t$, $s_t = f_t$, and $s_t > f_t$.

From the results obtained, it can be seen that dyNSGA-III outperformed the other three DMOEAs in terms of both HV and MIGD performance metrics. dyNSGA performed very well on the multimodal problems (GTA9m – GTA12m), which demonstrates its ability to adapt well to problems involving many optima. It scaled well against the decomposition-based PSO (dMOPSO), which has the ability to adapt to changing search spaces and shifting PFs. One challenge of dynamic multi-objective optimization is that it becomes increasingly difficult to locate optimum solutions when there are both multiple objectives, and

TABLE II MEAN AND STANDARD DEVIATION VALUES OF DMOEAS FOR HYPERVOLUME (HV) METRIC. STANDARD DEVIATION VALUES IN PARENTHESES

DMOP	(s_t , f_t)	dNSGA-II	SGEA	dMOPSO	dyNSGA-III
GTA 1a	(5,10)	3.211E-01(4.592E+00)	2.247E+02(3.512E-01)	3.109E-01(3.986E-01)	**2.492E-02(3.773E-01)**
	(10,10)	4.217E-01(3.882E-01)	4.474E-01(4.921E-01)	3.728E-02(2.159E-02)	**3.147E-02(1.935E-02)**
	(20,10	6.438E-01(3.991E-01)	3.483E-01(3.294E-01)	2.115E-01(3.582E-01)	**2.004E-01(3.129E-02)**
GTA 2a	(5,10)	**3.105E-01(3.893E-01)**	2.247E+01(2.301E-01)	3.109E-01(3.986E-01)	3.517E-01(2.885E-01)
	(10,10)	3.606E-02(3.305E-01)	3.116E-01(2.628E+00)	**2.111E-02(1.869E-02)**	3.531E-02(2.794E-02)
	(20,10)	4.149E-01(4.714E-01)	**2.117E-02(2.318E-01)**	3.746E-01(3.164E-01)	2.419E-01(5.653E-02)
GTA 3a	(5,10)	1.621E-01(2.992E-02)	**3.071E-02(2.028E-02)**	2.052E-01(2.027E-01)	1.449E-01(3.108E-01)
	(10,10)	1.942E+00(2.763E-01)	4.840E-01(3.721E+01)	3.502E-02(2.194E-01)	**2.803E-02(4.382E-02)**
	(20,10)	5.031E-01(2.594E-01)	3.684E-01(3.116E+02)	3.136E-01(2.334E-01)	**2.236E-02(3.459E-02)**
GTA 4a	(5,10)	**3.519E-02(2.117E-01)**	3.853E-02(4.063E-01)	3.819E-02(2.581E-01)	3.692E-02(2.093E-01)
	(10,10)	2.054E-01(2.229E-01)	3.832E-01(3.962E-01)	3.115E-02(2.128E-01)	**2.688E-02(2.629E-02)**
	(20,10)	**3.432E-02(3.112E-01)**	4.553E-01(1.113E-02)	3.258E-01(2.127E-02)	4.076E-02(2.075E-02)
GTA 5a	(5,10)	3.937E-02(1.418E-01)	3.456E-02(3.0962E-01)	3.295E-02(2.197E-01)	**2.117E-02(1.295E-02)**
	(10,10)	2.666E-01(2.438E-01)	2.063E-01(3.732E-01)	2.507E-02(3.198E-01)	**1.064E-02(2.952E-01)**
	(20,10)	3.638E+00(2.107E-01)	4.118E-01(2.062E-02)	3.110E-01(2.586E-02)	**3.329E-02(2.086E-02)**
GTA 6m	(5,10)	3.885E-02(3.486E-01)	**1.195E-02(1.096E-02)**	3.295E-02(2.197E-01)	2.195E-02(1.836E-02)
	(10,10)	3.386E-01(4.465E-01)	2.063E-01(3.732E-01)	2.406E-02(2.225E-01)	**2.583E-03(2.329E-02)**
	(20,10)	3.307E-01(3.065E-01)	2.527E-02(2.108E-02)	3.106E-02(2.075E-02)	**2.006E-03(1.663E-02)**
GTA 9m	(5,10)	2.954E-01(3.754E-01)	4.037E-02(4.748E-01)	**2.064E-02(2.062E-02)**	2.905E-02(3.033E-02)
	(10,10)	**2.053E-03(2.047E-02)**	3.337E-01(2.017E-01)	4.053E-02(3.116E-01)	2.557E-03(2.336E-02)
	(20,10)	**1.630E-02(2.058E-02)**	3.061E-02(3.505E-02)	3.721E-02(3.146E-02)	2.884E-02(2.924E-02)
GTA 10m	(5,10)	2.836E-02(3.892E-02)	3.965E-02(4.105E-01)	2.049E-02(3.058E-02)	**1.885E-03(2.227E-02)**
	(10,10)	3.116E-01(2.047E-02)	2.406E-01(2.437E-01)	4.158E-02(1.296E-01)	**2.067E-03(3.258E-02)**
	(20,10)	1.566E-01(2.731E-02)	3.113E-01(2.062E-01)	3.024E-02(3.188E-02)	**2.884E-02(2.006E-02)**
GTA 11m	(5,10)	2.401E-01(2.782E-01)	3.483E-02(4.105E-01)	3.270E-02(3.374E-02)	**2.045E-02(3.018E-02)**
	(10,10)	4.834E-01(5.492E-02)	2.337E-01(2.285E-01)	**3.062E-02(3.115E-02)**	2.067E-01(2.384E-01)
	(20,10)	3.820E-02(4.018E-02)	3.641E-01(3.292E-02)	3.505E-02(3.774E-02)	**2.023E-03(2.205E-02)**
GTA 12m	(5,10)	1.398E-01(1.273E-01)	4.144E-02(3.119E-02)	3.189E-02(3.288E-02)	**2.009E-03(2.116E-02)**
	(10,10)	1.873E-01(2.493E-02)	2.886E-02(1.732E-01)	2.587E-02(2.479E-02)	**1.993E-02(2.021E-02)**
	(20,10)	3.442E-02(3.984E-02)	3.593E-02(3.226E-02)	2.458E-02(2.841E-02)	**1.889E-02(1.993E-02)**

a time-changing Pareto front. The proposed approach adapts well to both multiple objectives and a dynamic Pareto front by adjusting both the fitness and orientation of the search candidates within the solution space. The problem of premature convergence and poor exploitation of the solution space is also addressed by satisfying the stopping criterion specified in Algorithms 1 and 2. We ensure that the best possible Euclidean distance is achieved by the algorithm with respect to the mean value $E(\boldsymbol{c})$.

Fig. 2 shows the convergence rate of each one of the four tested DMOEAs over 5,000 feature evaluations. From the Figure, we see that dyNSGA-III and dMOPSO are the best performers. We observe that dyNSGA-III converges at a slightly slower rate than dMOPSO over the feature evaluations. This slower rate of convergence is advantageous in highly multimodal problems where the exploitative capability of the search algorithm is tested.

Regarding the ability of dyNSGA-III to obtain robust solutions to dynamic optimization problems over time, we compare its performance with that of the other three specified DMOEAs for optimizing a single objective. We also evaluate performance for the RPOOT benchmark problems for multiple objectives. We compare the performance of dyNSGA-III with SGEA and dMOPSO since dNSGA-II is worst performer for robust single-objective optimization. Details of ROOT benchmark test problems used are specified in Section IV(B). Performance metrics used are specified in Section IV(C).

From the results obtained in Table IV and V, we see that robustness average fitness and survival time results obtained are mostly higher for dyNSGA-III compared to SGEA, dNSGA-II and dMOPSO. The ROOT problems used are denoted as $RMPB_{xy}$ (where x indicates the problem baseline, and y indicates the kind of dynamic that characterizes the problem). Results in boldface indicate the best-performing metric. We use feature evaluations as time steps to evaluate performance of the DMOEAs at various instances in the simulation. We observe that dyNSGA-III performs very well on RMPB-I problems with time window T set to 10 (Table V).

For the RMPB-II problems, we set the fitness threshold V to 20 to evaluate the survival time of the selected DMOEAs. From the results obtained, the dyNSGA-III algorithm outperformed the other three DMOEAs for most of the instances of the test problems. However, we further examined the susceptibility of the proposed algorithm to noise dynamics which is characterized by $RMPB_{16}$ and $RMPB_{26}$. We consider the error metrics as well as the solution sequence sensitivity for each of the DMOEAs for the two selected problems to further establish the robustness of dyNSGA-III. Results are shown in Table IV with best-performing results indicated in boldface. Therefore, we observe that dyNSGA-III performs better than dMOPSO for the multiplicative forms of the DMOPs (GTA9m – 12m). These problems have many local minima, which means that search agents can easily become trapped before reaching the global optimum.

Details of RPOOT benchmark problems and performance metrics are specified in Section IV(D). The RPOOT benchmark problems used to test the ability of dyNSGA-III to obtain robust solutions over changing multi-objective space include: F1-F4 which present complexities such as nonconvexity, disconnectedness and deceptiveness [31]. F9-F11 present additional complexities including changing Pareto set (PS) and Pareto front (PF) [32]. These complexities are created to test the ability of DMOEAs to adjust to time-varying characteristics describing a given MOP, while maintaining suitable solutions over time.

From the results obtained in Table VI, we observe that dyNSGA-III and SGEA are the best performers for RPOOT benchmark problems. However, dyNSGA-III could perform better for problems F9-F11 which involve complicated PF. The computational complexity of dyNSGA-III is $O(2N^2log^{M-2}N)$ where N is population size and M is dimensionality of objective function vectors. This computational complexity is comparable to that of SGEA, dMOPSO and original NSGA-III.

TABLE III MEAN AND STANDARD DEVIATION VALUES OF DMOEAS FOR MEAN INVERTED GENERATIONAL DISTANCE (MIGD) METRIC. STANDARD DEVIATION VALUES IN PARENTHESES

DMOP	(s_t, f_t)	dNSGA-II	SGEA	dMOPSO	dyNSGA-III
GTA 1a	(5,10)	3.115E-01(3.047E-01)	1.306E+00(2.875E-01)	2.317E-02(3.584E-02)	**2.419E-03(2.784E-02)**
	(10,10)	2.169E-01(3.114E-01)	2.224E-01(3.218E-01)	2.853E-01(1.482E-01)	**3.117E-02(1.582E-02)**
	(20,10	1.115E-01(1.032E-01)	2.592E-01(1.593E-01)	3.120E-02(2.834E-02)	**2.379E-02(2.183E-02)**
GTA 2a	(5,10)	1.650E-02(1.631E-02)	2.963E+01(3.014E-01)	1.937E-02(1.847E-02)	**1.729E-03(1.897E-03)**
	(10,10)	2.058E-02(1.994E-02)	2.756E-01(2.847E-01)	2.337E-02(2.260E-02)	**3.433E-02(3.280E-02)**
	(20,10)	2.215E-02(2.119E-02)	2.505E-01(2.476E-01)	2.227E-02(2.183E-02)	**2.774E-03(2.873E-03)**
GTA 3a	(5,10)	**2.047E-03(2.159E-03)**	4.008E-02(4.195E-02)	2.052E-01(2.027E-01)	2.216E-02(2.732E-02)
	(10,10)	3.197E-02(2.491E-02)	2.258E-01(1.067E-01)	2.562E-02(2.369E-02)	**4.769E-03(5.693E-03)**
	(20,10)	1.683E-01(1.493E-01)	2.680E-01(2.684E-01)	1.406E-01(1.329E-01)	**2.481E-02(2.280E-02)**
GTA 4a	(5,10)	**1.985E-03(1.026E-03)**	3.606E-02(3.592E-01)	2.538E-02(2.240E-02)	4.655E-01(1.024E-02)
	(10,10)	3.115E-01(3.218E-01)	3.631E-01(3.730E-01)	3.157E-02(2.189E-02)	**3.054E-02(2.047E-02)**
	(20,10)	**1.796E-02(1.360E-02)**	5.726E-01(1.113E-02)	3.258E-01(2.769E-02)	2.575E-02(2.216E-02)
GTA 5a	(5,10)	2.337E-02(1.649E-02)	2.479E-02(2.054E-02)	3.568E-03(2.117E-03)	**3.226E-03(3.189E-03)**
	(10,10)	2.398E-02(2.546E-02)	2.932E-01(2.821E-01)	2.717E-03(3.003E-03)	**1.558E-03(4.279E-03)**
	(20,10)	1.961E-01(2.107E-01)	1.034E-01(2.184E-02)	3.471E-02(2.627E-02)	**2.569E-02(1.058E-02)**
GTA 6m	(5,10)	2.961E-02(2.941E-02)	**2.019E-02(1.995E-02)**	2.007E-01(1.948E-01)	2.268E-02(3.445E-02)
	(10,10)	3.310E-01(3.217E-01)	2.215E-01(2.175E-01)	**2.418E-02(2.388E-02)**	1.237E-01(2.329E-02)
	(20,10)	3.684E-01(3.559E-01)	2.549E-02(2.446E-02)	2.116E-02(2.104E-02)	**1.389E-03(1.369E-03)**
GTA 9m	(5,10)	3.691E-01(3.501E-01)	3.511E-02(3.642E-01)	**2.582E-02(2.495E-02)**	3.117E-02(3.106E-02)
	(10,10)	**2.225E-03(2.179E-03)**	3.119E-01(3.217E-01)	2.871E-02(2.941E-02)	2.693E-03(2.731E-02)
	(20,10)	2.236E-02(2.185E-02)	**1.270E-02(1.106E-02)**	3.718E-02(3.888E-02)	2.694E-02(2.792E-02)
GTA 10m	(5,10)	3.815E-02(3.932E-02)	3.842E-02(3.726E-02)	2.711E-02(2.641E-02)	**1.896E-03(1.934E-03)**
	(10,10)	3.825E-01(2.782E-02)	2.258E-01(2.348E-01)	4.316E-02(4.239E-01)	**2.285E-03(2.118E-03)**
	(20,10)	3.556E-01(2.093E-02)	3.278E-01(2.994E-01)	3.217E-02(3.118E-02)	**2.998E-02(2.923E-02)**
GTA 11m	(5,10)	2.306E-01(2.415E-01)	3.821E-02(3.634E-02)	2.653E-02(2.579E-02)	**2.526E-02(1.285E-03)**
	(10,10)	4.395E-01(3.451E-02)	3.068E-01(3.033E-01)	**2.167E-02(1.647E-02)**	2.108E-01(1.121E-01)
	(20,10)	3.581E-02(3.661E-02)	3.427E-01(3.371E-01)	2.873E-02(2.779E-02)	**2.495E-03(2.384E-02)**
GTA 12m	(5,10)	1.470E-01(1.333E-01)	4.384E-02(3.469E-02)	3.007E-02(2.986E-02)	**1.772E-03(1.629E-02)**
	(10,10)	1.904E-01(2.512E-02)	2.694E-02(1.653E-01)	2.463E-02(2.589E-02)	**1.726E-02(1.882E-02)**
	(20,10)	3.378E-02(4.056E-02)	3.567E-02(3.492E-02)	2.304E-02(2.996E-02)	**1.593E-02(1.448E-02)**

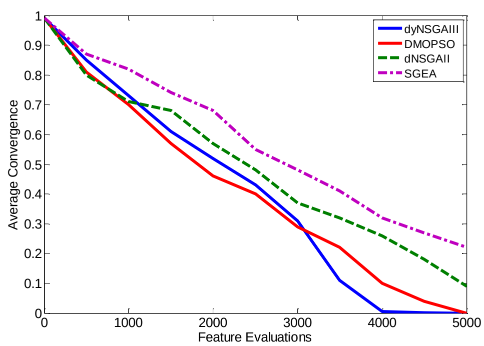

FIGURE 2. Average convergence of 4 tested DMOEAs.

VI. CONCLUSIONS

This paper has proposed the use of adaptive mutation and crossover operators to track the moving Pareto front associated with dynamic optimization problems. Many real-world problems are modeled as multi-objective optimization problems with time-dependent Pareto fronts. While several researches have been presented which suggest approaches to better track the moving Pareto fronts, the problem of suboptimal solutions due to premature convergence of the DMOEAs still persists.

Optimal tuning of the mutation and recombination operators of genetic algorithms is vital for improving their ability to select the best Pareto set to solve a DMOP. This is

TABLE IV PERFORMANCE COMPARISON OF DMOEAS FOR ERROR AND SENSITIVITY INDICES FOR NOISY INSTANCES OF RMPB-I AND -II

DMOEA	Performance Index	$RMPB_{16}$	$RMPB_{26}$
dNSGA-II	$e_{best,s}$	2.34E-02	1.67E-01
	$e_{av,s}$	1.88E-01	**2.07E-03**
	$e_{worst,s}$	1.04E-02	1.85E-02
	θ_s	0.234	0.217
SGEA	$e_{best,s}$	1.64E-03	1.95E-03
	$e_{av,s}$	1.64E-02	1.52E-02
	$e_{worst,s}$	2.84E-03	1.89E-02
	θ_s	0.195	0.219
dMOPSO	$e_{best,s}$	1.57E-03	2.73E-02
	$e_{av,s}$	2.79E-02	2.47E-03
	$e_{worst,s}$	1.95E-03	1.34E-03
	θ_s	0.177	0.165
dyNSGA-III	$e_{best,s}$	**3.32E-04**	**1.79E-03**
	$e_{av,s}$	**3.18E-03**	2.70E-03
	$e_{worst,s}$	**1.88E-03**	**4.85E-04**
	θ_s	**0.093**	**0.127**

because these operators determine the fitness of search candidates within the solution space. Poor mating and reproduction of parents will produce offspring that will not guarantee the integrity of successive generations of search candidates. This results in the inability of the DMOEA to keep track of the moving Pareto front, particularly when the DMOP is also multimodal.

In this paper, we propose the use of Euclidean distance measurement as a means of slowing down the movement of the Pareto front. We achieve this by specifying a covariance matrix with respect to the mean value within the search space. We then ensure that the distance between mean value and a potential candidate of the Pareto set is maximum, and this constitutes our stopping criterion. From the results obtained, we observe that our proposed dyNSGA-III algorithm scales very well compared to other well-performing DMOEAs. We observe that our proposed algorithm performs best for multimodal DMOPs which demonstrates its capability to adapt well to changing search environments. In terms of convergence rate, we see that the proposed algorithm does not converge too fast over the feature evaluations which means that it has good exploitative capability. To test the capability of the proposed dyNSGA-III algorithm to obtain robust solutions to dynamic optimization problems over time, we test its performance on the RMPB test suite. The reason for this is because many real-life problems are characterized by rapidly changing optima.

Overall, this research has highlighted prominent challenges that exist while attempting to solve DMOPs. These include:

i. Maintaining balance between convergence and diversity.
ii. Effectively tracking changing PF due to time-varying fitness landscape with respect to candidate solutions.
iii. Maintaining robust solutions despite a moving PF.

The algorithm proposed in this research has tackled the above problems using the following methodology:

i. Ensuring effective mutation and crossover (recombination) of population candidates using PCA mutation and *n*-point crossover.
ii. Using Euclidean distance measurement to slow down movement of the PF by implementing covariance matrix with respect to the mean value within the search space.
iii. Ensuring that performance of the proposed algorithm (compared to other well-performing DMOEAs) is not diminished by problems of high dimensionality and rapidly changing objectives and PF. We achieved this by using both HV and IGD performance metrics (Tables II and III).

Future research will consider improving the performance of the proposed dyNSGA-III algorithm on robust Pareto-optimal solutions over time (RPOOT) for multiple objectives with complicated PF. We will also consider the effect of adaptive reference point placement in the search space in addition to adaptively varying the crossover and mutation rates for the NSGA-III algorithm. This will help us to examine the possibility of further improving the capability of DMOEAs to handle highly dynamic, multimodal problems. In addition, we will consider problems with more than three objectives, which are representative of many real-world problem models. Regarding the practical adaptability of the proposed DMOEA, we will also test its performance on optimization of dynamic mathematical models representing real-life systems which are susceptible to frequent environmental changes over time. The efficacy of specific MOEAs such as artificial bee colony algorithm (ABC), Hybrid Spider Monkey algorithm (HSMA), and Raccoon family optimization algorithm (RFOA) has been highlighted in recent literature. In future, we will compare the performance of the improved dyNSGA-III algorithm with these MOEAs.

ACKNOWLEDGMENT

This research is supported partially by South African National Research Foundation Grants (No. 112108 and 112142), and South African National Research Foundation Incentive Grant (No. 95687), Eskom Tertiary Education Support Programme, Research grant from URC of University of Johannesburg.

REFERENCES

[1] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An adaptive approach for solving dynamic scheduling with time-varying number of tasks: Part II," in *Proc. IEEE CEC*, Jun. 2011, pp. 1711–1718.

[2] J. Ding, C. Yang, Q. Xiao, T. Chai, and Y. Jin, "Dynamic Evolutionary Multiobjective Optimization for Raw Ore Allocation in Mineral Processing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 1, pp. 36–48, Feb. 2019, doi: 10.1109/TETCI.2018.2812897.

TABLE V PERFORMANCE OF DMOEAS OVER FEATRURE EVALUATIONS FOR RMPB-I AND –II WITH TIME WINDOW T=10 AND FITNESS THRESHOLD V=20

Test Problem	Problem Instance	# of Feature Evaluations	dNSGA-II	SGEA	dMOPSO	dyNSGA-III
RMPB-I (T=10)	$RMPB_{11}$	1000	31.74	33.12	33.56	**34.14**
		3000	28.49	26.52	29.57	**29.91**
		5000	36.28	31.40	36.44	**37.71**
	$RMPB_{12}$	1000	41.43	42.17	**42.22**	41.87
		3000	40.65	39.51	41.93	**42.05**
		5000	38.72	34.82	39.41	**41.63**
	$RMPB_{13}$	1000	32.19	30.04	33.18	**34.16**
		3000	39.73	37.29	38.62	**40.04**
		5000	34.31	35.03	33.18	**35.94**
	$RMPB_{14}$	1000	29.46	31.94	31.82	**32.17**
		3000	30.41	28.84	31.83	**31.97**
		5000	31.68	32.74	34.73	**34.95**
	$RMPB_{15}$	1000	27.76	26.32	25.72	**29.21**
		3000	29.67	30.08	29.47	**31.94**
		5000	26.39	24.94	**27.72**	26.54
	$RMPB_{16}$	1000	39.37	38.28	40.59	**41.84**
		3000	41.84	40.65	40.44	**42.16**
		5000	39.96	41.48	42.72	**43.29**
RMPB-II (V=20)	$RMPB_{21}$	1000	1.05	0.95	1.26	**1.31**
		3000	1.15	1.32	1.54	**1.98**
		5000	0.68	0.79	**1.09**	0.93
	$RMPB_{22}$	1000	1.23	1.36	1.32	**1.93**
		3000	1.11	**1.79**	1.63	1.59
		5000	2.04	1.94	1.98	**2.15**
	$RMPB_{23}$	1000	**2.27**	1.91	1.88	1.96
		3000	1.38	1.69	1.85	**2.06**
		5000	1.50	0.94	1.16	**1.69**
	$RMPB_{24}$	1000	0.84	0.74	1.14	**1.21**
		3000	1.33	1.49	**1.52**	1.43
		5000	0.69	0.72	0.58	**1.04**
	$RMPB_{25}$	1000	1.24	0.93	1.16	**1.49**
		3000	2.02	1.95	2.18	**2.27**
		5000	1.47	**2.42**	1.94	2.21
	$RMPB_{26}$	1000	1.65	1.49	1.88	**1.96**
		3000	0.97	**1.04**	0.85	0.93
		5000	1.08	1.62	1.48	**1.71**

TABLE VI PERFORMANCE OF DMOEAS FOR SELECTED RPOOT BENCHMARK PROBLEMS

Test Instance	Performance Index	SGEA	dMOPSO	dyNSGA-III
F1	t_r	2.419±0.019	2.957±0.281	**3.377±0.045**
	d_r	2.281E-02(4.921E-02)	3.662E-02(2.512E-02)	**1.329E-02(0.568E-03)**
	s_r	**3.194E-03(2.116E-02)**	1.741E-01(2.118E-01)	2.184E-02(1.529E-02)
F2	t_r	2.247±0.092	2.386±0.129	**4.301±0.058**
	d_r	**1.218E-03(2.732E-02)**	1.993E-01(2.042E-02)	2.041E-02(2.715E-02)
	s_r	2.683E+01(1.937E+00)	2.127E-02(2.227E-02)	**2.285E-04(3.116E-03)**
F3	t_r	**4.052±0.011**	2.588±0.009	3.831±0.001
	d_r	4.840E-03(2.851E-02)	2.819E-02(1.952E-01)	**2.491E-04(1.936E-04)**
	s_r	1.994E+01(2.872E+02)	2.176E-03(2.621E-01)	**1.926E-03(2.916E-03)**
F4	t_r	2.772±0.042	2.891±0.000	**5.027±0.003**
	d_r	2.951E-01(2.117E+00)	2.394E-02(1.924E-01)	**2.955E-03(1.417E-02)**
	s_r	3.174E-02(1.024E-02)	2.931E-01(1.302E-01)	**3.492E-04(2.075E-03)**
F9	t_r	**4.113±0.000**	3.139±0.041	2.998±0.002
	d_r	2.951E-01(1.954E-01)	**2.117E-03(3.064E-03)**	2.185E-01(2.003E-01)
	s_r	3.174E-02(2.218E-02)	2.196E-03(2.519E-02)	**3.117E-04(1.942E-02)**
F10	t_r	2.128±0.008	**4.431±0.001**	3.917±0.000
	d_r	**1.034E-03(2.051E-02)**	3.716E-02(1.023E-02)	2.388E-02(2.116E-02)
	s_r	**2.318E-03(1.942E-02)**	3.037E-02(2.118E-03)	3.519E-02(1.282E-02)
F11	t_r	2.916±0.003	2.285±0.001	**4.882±0.021**
	d_r	3.115E-03(2.019E-02)	3.962E-02(2.931E-02)	**2.318E-04(2.168E-03)**
	s_r	2.031E-02(3.111E-02)	2.501E-02(3.296E-02)	**2.185E-03(2.924E-02)**

[3] K. Deb, U. V. Rao, and S. Karthik, "Dynamic multiobjective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. EMO*, 2007, vol. 4403, pp. 803–817.

[4] A. K. Hutzschenreuter, P. A. Bosman, and H. Poutr´e, "Evolutionary multi-objective optimization for dynamic hospital resource management," in *EMO*, 2009, vol. 5467, pp. 320–334.

[5] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multi-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, 2014.

[6] Y.-N. Guo, X. Zhang, D.-W. Gong, Z. Zhang, and J.-J. Yang, "Novel Interactive Preference-Based Multiobjective Evolutionary Optimization for Bolt Supporting Networks," *IEEE Trans. Evol. Computat.*, vol. 24, no. 4, pp. 750–764, Aug. 2020, doi: 10.1109/TEVC.2019.2951217.

[7] Y.-N. Guo, J. Cheng, S. Luo, D. Gong, and Y. Xue, "Robust Dynamic Multi-Objective Vehicle Routing Optimization Method," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1891–1903, Nov. 2018, doi: 10.1109/TCBB.2017.2685320.

[8] M. Helbig and A. P. Engelbrecht, "Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2014, pp. 3151–3159, doi: 10.1109/CEC.2014.6900303.

[9] J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht, "A scalability study of many-objective optimization algorithms," *IEEE Trans. Ev. Comp.*, vol. 22, no. 1, pp. 79–96, 2018.

[10] Y. Jin, C. Yang, J. Ding, and T. Chai, "Reference point-based prediction for evolutionary dynamic multi-objective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 3769–3776, doi: https://doi.org/10.1109/CEC.2016.7744267.

[11] Y. Wang, T. Du, T. Liu, and L. Zhang, "Dynamic multi-objective squirrel search algorithm based on decomposition with evolutionary direction prediction and bidirectional memory populations," *IEEE Access*, vol. 7, pp. 115997–13, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2932883.

[12] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multi-objective optimization problems," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3362–3374, 2019, doi: https://doi.org/10.1109/TCYB.2018.2842158.

[13] M. Rauf, Z. Guan, L. Yue, Z. Guo, J. Mumtaz, and S. Ullah, "Integrated Planning and Scheduling of Multiple Manufacturing Projects Under Resource Constraints Using Raccoon Family Optimization Algorithm," *IEEE Access*, vol. 8, pp. 151279–151295, 2020, doi: 10.1109/ACCESS.2020.2971650.

[14] L. Yue, Z. Guan, L. Zhang, S. Ullah, and Y. Cui, "Multi objective lotsizing and scheduling with material constraints in flexible parallel lines using a Pareto based guided artificial bee colony algorithm," *Computers & Industrial Engineering*, vol. 128, pp. 659–680, Feb. 2019, doi: 10.1016/j.cie.2018.12.065.

[15] J. Mumtaz, Z. Guan, L. Yue, Z. Wang, S. Ullah, and M. Rauf, "Multi-Level Planning and Scheduling for Parallel PCB Assembly Lines Using Hybrid Spider Monkey Optimization Approach," *IEEE Access*, vol. 7, pp. 18685–18700, 2019, doi: 10.1109/ACCESS.2019.2895954.

[16] I. Essiet, Y. Sun, and Z. Wang, "Novel Algorithm for optimizing the Pareto Set in Dynamic Problem Spaces," in *2018 Conference on Information Communications Technology and Society (ICTAS)*, Durban, South Africa, Mar. 2018, pp. 110–115.

[17] Y. Jin, K. Tang, X. Yu, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Comp.*, vol. 5, no. 1, pp. 3–18, Mar. 2013, doi: 10.1007/s12293-012-0090-2.

[18] Z. Yang, Y. Jin, and K. Hao, "A bio-inspired self-learning coevolutionary dynamic multi-objective optimization algorithm for internet of things services," *IEEE Trans. Ev. Comp.*, vol. 23, no. 4, pp. 675–688, 2019, doi: https://doi.org/10.1109/TEVC.2018.2880458.

[19] R. Chen, K. Li, and X. Yao, "Dynamic Multiobjectives Optimization With a Changing Number of Objectives," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, Feb. 2018, doi: 10.1109/TEVC.2017.2669638.

[20] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Trans. Ev. Comp.*, vol. 18, no. 4, pp. 577–601, 2014.

[21] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach," *IEEE Trans. Ev. Comp.*, vol. 18, no. 4, pp. 602–622, 2014.

[22] Y. Ait-Sahalia and D. Xiu, "Using principal component analysis to estimate a high dimensional factor model with high-frequency

data," *J. Econom.*, vol. 201, no. 2, pp. 384–399, 2017, doi: https://doi.org/10.1016/j.jeconom.2017.08.015.

[23] K. De Jong and W. Spears, "A formal analysis of the role of multipoint crossover in genetic algorithms," *Annals Math. Art. Int.*, vol. 5, no. 1, pp. 1 – 26, 1992.

[24] A. E. Eiben and J. E. Smith, "Representation, Mutation and Recombination," 2nd ed., vol. 2, Springer, 2015, pp. 49–78.

[25] S. B. Gee, K. C. Tan, and H. A. Abbass, "A Benchmark Test Suite for Dynamic Evolutionary Multiobjective Optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 461–472, Feb. 2017, doi: 10.1109/TCYB.2016.2519450.

[26] Y. Yuan, H. Xu, and B. Wang, "An improved NSGA-III procedure for evolutionary many-objective optimization," in *Proceedings of the 2014 conference on Genetic and evolutionary computation - GECCO '14*, Vancouver, BC, Canada, 2014, pp. 661–668, doi: 10.1145/2576768.2598342.

[27] X. Yu, Y. Jin, K. Tang, and X. Yao, "Robust optimization over time — A new perspective on dynamic optimization problems," in *IEEE Congress on Evolutionary Computation*, Jul. 2010, pp. 1–6, doi: 10.1109/CEC.2010.5586024.

[28] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Robust Optimization Over Time: Problem Difficulties and Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 731–745, Oct. 2015, doi: 10.1109/TEVC.2014.2377125.

[29] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Finding Robust Solutions to Dynamic Optimization Problems," in *Applications of Evolutionary Computation*, vol. 7835, A. I. Esparcia-Alcázar, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 616–625.

[30] Y. Guo, H. Yang, M. Chen, J. Cheng, and D. Gong, "Ensemble prediction-based dynamic robust multi-objective optimization methods," *Swarm and Evolutionary Computation*, vol. 48, pp. 156–171, Aug. 2019, doi: 10.1016/j.swevo.2019.03.015.

[31] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, Oct. 2004, doi: 10.1109/TEVC.2004.831456.

[32] S. Biswas, S. Das, P. N. Suganthan, and C. A. C. Coello, "Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2014, pp. 3192–3199, doi: 10.1109/CEC.2014.6900487.