DPb-MOPSO: A Novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization Algorithm

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

03-12-2020 / 12-01-2022

CITATION

Aboud, Ahlem; Rokbani, Nizar; Fdhila, Raja; Qahtani, Abdulrahman M.; Almutiry, Omar; dhahri, habib; et al. (2020): DPb-MOPSO: A Novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization Algorithm. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.13325354.v2>

DOI

[10.36227/techrxiv.13325354.v2](https://doi.org/10.36227/techrxiv.13325354.v2)

DPb-MOPSO: A Novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization Algorithm

Ahlem Aboud ^{*,1,2,✉}, Nizar Rokbani ^{2,3}, Raja Fdhila ², Abdulrahman M. Qahtani ⁴, Omar Almutiry ⁵, Habib Dhahri ⁵, Amir Hussain ⁶, and Adel M. Alimi ^{2,7}

Abstract—Particle swarm optimization system based on the distributed architecture over multiple sub-swarms has shown its efficiency for static optimization and has not been studied to solve dynamic multi-objective problems (DMOPs). When solving DMOP, tracking the best solutions over time and ensuring good exploitation and exploration are the main challenging tasks. This study proposes a novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) algorithm including two parallel optimization levels. At the first level, all solutions are managed in a single search space. When a dynamic change is successfully detected in the objective values, the Pareto ranking operator is used to enable a multiple sub-swarm' subdivisions and processing which drives the second level of enhanced exploitation. A dynamic handling strategy based on random detectors is used to track the changes of the objective function due to time-varying parameters. A response strategy consisting in re-evaluate all unimproved solutions and replacing them with newly generated ones is also implemented. Inverted generational distance, mean inverted generational distance, and hypervolume difference metrics are used to assess the DPb-MOPSO performances. All quantitative results are analyzed using Friedman's analysis of variance while the Lyapunov theorem is used for stability analysis. Compared with several multi-objective evolutionary algorithms, the DPb-MOPSO is robust in solving 21 complex problems over a range of changes in both the Pareto optimal set and Pareto optimal front. For 13 UDF and ZJZ functions, DPb-MOPSO can solve 8/13 and 7/13 on IGD and HVD with moderate changes. For the 8 FDA and dMOP benchmarks, DPb-MOPSO was able to resolve 4/8 with severe change on MIGD, and 5/8 for moderate and slight changes. However, for the 3 kind of environmental changes, DPb-MOPSO assumes 4/8 of the solving function on IGD and HVD metrics.

Index Terms—Dynamic Multi-Objective Optimization, Dynamic Particle Swarm Optimization, Dynamic Problems, Detectors, Dynamic Response, Friedman Analysis of Variance, Lyapunov Theorem.

¹ University of Sousse, ISITCom, 4011, Sousse, Tunisia.

² REsearch Groups in Intelligent Machines (REGIM Lab), University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia.

³ High Institute of Applied Science and technology of Sousse, University of Sousse, Tunisia.

⁴ Department of Computer Science, College of Computers and Information Technology, Taif University, P.O.Box. 11099, Taif 21944, Saudi Arabia.

⁵ College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia.

⁶ Edinburgh Napier University, School of Computing, Edinburgh EH10 5DT, Scotland, U.K.

⁷ Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa.

✉ e-mail : ahlem.aboud@regim.usf.tn

* Corresponding author.

I. INTRODUCTION

DYNAMIC Multi-Objective Optimization (DMOO) approaches aims to solve Dynamic Multi-Objective Problems (DMOPs), which may have two or three conflicting functions characterized by time-varying objectives, parameters, and/or constraints. Farina *et al.* [1] categorized DMOPs into four types. For type I, the Pareto Optimal Set (POS) changes, and the Pareto Optimal Front (POF) remains unchanged, while for type II both POS and POF change over time. However, the POF changes, and the POS remains constant for DMOPs in type III. Finally, for type IV, and despite the environmental changes both POS and POF keep unchanged.

Multi-Objective Evolutionary Algorithms (MOEAs) and bio-inspired intelligent techniques like the Genetic Algorithm (GA) [2], and the Particle Swarm Optimization (PSO) method [3]–[5] has being investigated to manage the evolutionary stagnation issues for static single/multi-objective optimization. Also, a set of MOEAs have being established as suitable for dynamic multi-objective optimization [6]–[8], the proposals were able to find the best compromise solutions set on POS and POF, without managing environmental changes.

Different Dynamic MOEAs (DMOEAs) have been developed to overcome the loss of convergence and diversity issue when solving time-varying problems. Some mechanisms are considered for diversity detection by monitoring a few random individuals to re-evaluate the objectives or constraints violations values [10]. The change detection is usually done frequently by monitoring the evolution of the fitness function using one fixed solution [11], the best obtained position [12], or the average of all evaluated solutions [13]. The convergence detection methods using one or small set of detectors have the limit to wrongly detect the converged/ or the non-converged individuals. Also, the efficiency of the detection mechanisms decreases when the size of the affected environmental change decrease, and while the needs of an important number of detectors.

Furthermore, change detection includes additional techniques to react effectively to the change. The reinitialization of certain percentage of population, and the hypermutation are explicitly used to ensure good convergence and diversity in the dynamic search space [14], [15]. Nonetheless, genetic and population-based strategies on distributed methods are considered for static optimization [16] and have not yet been processed for DMOPs. Besides, many of the transfer learning-based methods proposed for DMOPs are time-consuming, and increases the diminishing of solutions diversity [17].

In general, the sequential search procedure proposed in the standard MOPSO approach [18] have followed a centralized architecture, where the particles are iteratively updated according to a random global best position (leader) and their historical personal best position. As consequence, if the personal best and the global best are confused or close to each other's, the MOPSO algorithm may stagnates in local optima and loses diversity over time [19]. Furthermore, the use of multi-objective particle swarm optimization (MOPSO) over a distributed architecture, consisting of multiple sub-swarms [20], has not yet been investigated to solve DMOPs. Also, it

should be noted, that the standard PSO method [3] is not very efficient for solving DMOPs with time-varying POS and POF [9] because of its limited capacities in addressing the dual issue of the outdated memory and the loss of diversity over time. The main difference of the new proposed distributed MOPSO toward MOPSO is that it includes a search space control allowing to manage sub-swarms in a specific different potential solutions space.

So, the main advantages of using a distributed MOPSO architecture for solving dynamic multi-objective problems, DMOPS, are as follows:

- Distributed MOPSO allows higher diversity during the search process.
- Distributed sub-swarms conduct independent search with separate pools of best solutions.
- The proposal includes a dynamic handling strategy which allows to manage the evolution of the search toward optimality.
- The dynamic switch mechanism allows to recover from any engagement within a local optimum.

This research is interested to propose a Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO). The system adopts a distributed architecture with two parallel optimization levels to handle DMOPs with various types of changes on POS and POF. For the upper level (L_1), the swarm acts in a single search space, pushing all particles to the best-compromise solutions. Nevertheless, if a dynamic change is successfully detected by monitoring the objective values at the first level. The second level processing provides exploitation enhancement over a dynamic multiple sub-swarm's subdivision based on the Pareto ranking operator. DPb-MOPSO proposes a dynamic handling strategy to detect and effectively respond to changes. For the detection strategy, 10% of randomly selected solutions $POS(t)$ have been used as detectors and compared with the previous solutions set $POS(t-1)$. The response strategy consists in re-evaluating all solutions with negative improvement and replaced them with new random solutions. Solutions with positive progress are transferred to build a new population in order to speed up the optimization process after changes. Non-dominated solutions are then re-evaluated in the archive to avoid outdated memory issue.

The state of the art involves comparable algorithms to the novel DPb-MOPSO system based essentially on three metrics [15]: Inverted Generational Distance (IGD), Mean Inverted Generational Distance (MIGD), and Hypervolume Difference (HVD), which are chosen referred to the compared state-of-the-art methods in both papers [17], and [21]. Five DMOEAs, are compared with the DPb-MOPSO system including; Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [22], dynamic Competitive-cooperative Evolutionary Algorithm (dCOEA) [23], Population Prediction Strategy (PPS) [24], Dynamic Non-dominated Sorting Genetic Algorithm II (DNSGA-II) [10] and Steady-State Generational Evolutionary Algorithm (SGEA) [21]. Also, four PSO-based methods are considered namely: Dual multi-objective particle swarm optimization (dMOPSO) [25], distributed Multi-

Objective Particles Swarms Optimization, based on dynamic subdivision of the population using Pareto fronts (pbMOPSO) [16] and Multi-Objective Particles Swarms Optimization (MOPSO) [18], and Dynamic-MOPSO [19].

The transfer learning-based methods aims to modify the main system of PPS [24], and MOEA/D-KF [26] with the baseline MOEA/D algorithm [22]. A set of TL methods are compared in [17], and denoted by; MMTL-MOEA/D, RI-MOEA/D, PPS-MOEA/D, KF-MOEA/D, SVR-MOEA/D [27] and Tr-MOEA/D [28] approaches are compared with DPb-MOPSO system.

The remaining of this paper is organized as follows. Section II introduces the overview of dynamic multi-objective optimization methods. Section III details the proposed DPb-MOPSO algorithm. Section IV introduces the preliminary of the experimental study. The analysis and discussion of the results are shown in Section V. The Lyapunov theorem used for stability analysis is presented in Section VI. Finally, Section VII summarizes the paper and suggest some prospects for future works.

II. DYNAMIC MULTI-OBJECTIVE OPTIMIZATION METHODS

Dynamic multi-objective optimization problem f_t is a MOP with m time-varying objective function, variables, and/or constraints. An algorithm G is needed to solve DMOPs in a given time period $[t_{begin}, t_{end}]$. The mathematical definition is expressed in Equation 1.

$$\begin{aligned} \text{Minimize } F(x, t) &= (f_1(x, t), f_2(x, t), \dots, f_m(x, t)) & (1) \\ \text{Subject to: } g_i(x, t) &\leq 0 \text{ and } h_j(x, t) = 0 \forall i, j = 1, \dots, n \\ x &\in [x_{min}, x_{max}], t \in [t_{begin}, t_{end}] \end{aligned}$$

Several studies were interested in dynamic multi-objective optimization at three research levels, and therefore include the development of benchmarks, algorithms and quality indicators. The most known DMOPs are; three dMOP problems such in [23], five continuous FDA functions such in [1] with dynamic shift on POF and POS, five ZJZ (F5 to F10) [24] with non-linear correlation between decision variables, nine Unconstrained Dynamic Functions (UDFs) [29] and a set of real-world problems [30].

Five categories of DMOEAs have been also proposed for DMOP and classified in Table I. The first category is the standard MOEAs designed for static multi-objective optimization and considered for DMOP. Among them the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [31], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [2], MOEA/D [22], and dCOEA [23].

Furthermore, a set of Transfer Learning-based (TL) methods were proposed to construct a predictive model that can learn from previous experience to predict future changes. Generally, TL methods have been proposed based on the baseline MOEA/D system [22]. When the environment changes, the modified MOEA/D algorithm with random initialization (RI-MOEA/D) [17] generates 10% of the new population. The Tr-

MOEA/D algorithm [28] integrates transfer learning and population-based evolutionary algorithms (EAs) to solve the DMOPs. Tr-MOEA/D algorithm determines the best potential space to select the optimal values of numerous hyper-parameters after each change. Memory-driven Manifold Transfer Learning algorithm (MMTL-MOEA/D) [17] aims to predict the initial population based on the elite of the previous solutions. The Kalman Filter (KF) prediction model based on MOEA/D (KF-MOEA/D) [26] uses a priori estimation to predict the posterior experience. An alternative Support Vector Regression MOEA/D (SVR-MOEA/D) [27] has also been proposed to predict the new population over time. The PPS system [24] is an autoregressive model aims to predict the center point and the manifold based on previous experience. However, a new variant is denoted by PPS-MOEA/D [17], where the original algorithm is modified using system using MOEA/D system [22].

Moreover, different diversity-based approaches maintain population diversity through a re-initialization or a hyper-mutation behavior after each transition or throughout the execution time. DnSGA-II-A and DnSGA-II-B [10] selects a random solution, and assess its re-evaluation to monitor any environmental change. The Dynamic Constrained NSGA-II (DC-NSGA-II) [32] proposed an adaptive penalty function to deal with time-varying constraints. In order to maintain diversity over time, the Dynamic-MOPSO system [19] controls the evolution of the objective function and reinitializes non suitable solutions.

A memory-based method is proposed in dynamic Competitive-cooperative co-evolutionary Algorithm (dCOEA) [23]. In [28] an Adaptive Dynamic NSGA-II (A-Dy-NSGA-II) used an additional memory mechanism to implicitly or explicitly store outdated individuals for potential use. Steady-State Generational Evolutionary Algorithm (SGEA) [21] aims to reuse outdated solutions with good distribution to relocate solutions closer to the new POF.

Different parallel methods with multiple sub-populations are used for DMOPs such in [8] and [33]. However, a set of performance metrics [15] have been considered for accuracy, diversity, and robustness measurements. Ultimately, convergence and diversity are measured using IGD, MIGD, and HVD quality indicators. This paper presents a dynamic distributed PSO-based approach, and tends to overcome the limits of loss of convergence and diversity issues when solving DMOPs. Despite all the additional mechanisms to detect and effectively react to changes, many DMOEAs have been failed to assume good convergence and diversity in the dynamic search space. More details about the proposed DPb-MOPSO are presented in section III.

TABLE I. DYNAMIC MULTI-OBJECTIVE OPTIMIZATION METHODS REVIEW

	Existing Approaches for DMOPs	Tested Benchmarks
Transfer Learning-based Methods	MMTL-MOEA/D, Jiang <i>et al.</i> , (2020) [17]	FDA1-5 [1], dMOP1-3 [23]
	RI-MOEA/D, Jiang <i>et al.</i> , (2020) [17]	
	PPS-MOEA/D, Jiang <i>et al.</i> , (2020) [17]	
	KF-MOEA/D, Jiang <i>et al.</i> , (2020) [17]	
	SVR-MOEA/D, Cao <i>et al.</i> , (2020) [27]	FDA1-5 [1], dMOP1-3 [23] and the DF test suite [34]
Diversity-based Approaches	Tr-MOEA/D, Jiang <i>et al.</i> , (2018) [28]	dMOP1-3 [23], FDA1-5, DIMP2, DMOP2, HE2,7,9 [1]
	Dynamic-MOPSO, Aboud <i>et al.</i> , (2017) [19]	FDA1 [1] and DIMP2, dMOP3 [23]
	DC-NSGA-II, Azzouz <i>et al.</i> , (2015) [35]	DCTPs test problems [35]
Memory-based Approaches	DNSGA-II-A and DNSGA-II-B, Deb <i>et al.</i> , (2007) [10]	Modified FDA2 and hydro-thermal scheduling problem[10]
	A-Dy-NSGA-II, Azzouz <i>et al.</i> , (2017) [36]	FDA1, FDA2 [1], DMZDT test functions and WYL
	SGEA, Jiang <i>et al.</i> , (2016) [21]	5 FDA [1], 6 dMOP [23], 6 ZJZ [24] and 7 UDF [29].
Prediction-based Approaches	dCOEA, Gohet <i>et al.</i> , (2009) [23]	FDA1 [1] and 3 dMOP [23]
	MOEA/D-KF, Muruganantham <i>et al.</i> , (2016) [26]	FDA1-5 [1], dMOP1-2 [23] and F5-10 [24]
Parallel Approaches	PPS, Zhou <i>et al.</i> , (2014) [24]	FDA1, FDA4 [1] and dMOP1, dMOP2 [23], F5-F8 [24]
	DMOEA, Liu <i>et al.</i> , (2007) [8]	FDA1, modified FDA2, FDA3, FDA4 and FDA5 [1]
Static MOEAs adapted for DMOPs	PSFGA, Camara <i>et al.</i> , (2007) [33]	FDA1 and FDA2 [1]
	SPEA2, Kim <i>et al.</i> , (2004) [31]	DSW1, DSW2, DTF, FDA DMOPs [1]
	MSOPS, Hughes (2003) [37]	
	NSGA-II, Deb <i>et al.</i> , (2002) [2]	

III. THE DYNAMIC PARETO BI-LEVEL MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

The general flowchart of the proposed Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) approach is shown in Figure 1. DPb-MOPSO has two optimization levels, denoted the upper level (L_1) in Algorithm 2 and the lower level (L_2) in Algorithm 3. The two levels have included a dynamism handling strategy presented in subsection C to detect and effectively respond to changes.

A. DPb-MOPSO: Upper Level (L_1)

The upper level (L_1) starts with a random initialization of the position of N particles p_i in swarm S (see Algorithm 1). Each particle is a candidate solution x_i^* in n dimensional search space. All non-dominated solutions in L_1 are stored in the leader's archive A . One global leader $gbest$ is selected from the archive (A) based on the epsilon dominance comparator [38] between two random particles.

A non-dominated solution S_1^* is ϵ -dominate a second non-dominated solution S_2^* for some $\epsilon > 0$ and denoted by $S_1^* \succ_{\epsilon} S_2^*$, if and only if for all f_i objectives $\forall, i = 1, \dots, m$, $F(S_1^*) = \{f_1, \dots, f_m\}$, and $F(S_2^*) = \{f_1, \dots, f_m\}$ such that $(1 + \epsilon) \cdot f_i(S_1^*) \geq f_i(S_2^*)$. While, the $pbest$ solution for each particle has presented as the historical personal best solution.

Algorithm 1: Swarm Initialization

Input: P (swarm of N particles), N (size of the population)

Output: Set of non-dominated solutions in the leader's archive A

1. Initialize $t=0$;
 2. for $i = 1$ to N do
 3. randomly initialize the vector of position X_i and zero velocity V_i ;
 4. Evaluate fitness function for the particle p_i ;
 5. Initialize $pbest = p_i$;
 6. end for
 7. Store the non-dominated solutions in the leader's archive A
-

The update of position (X) and velocity (V) at discrete time t

is done based on the best self-experience ($pbest$) as a local component and the global experience of all neighbors as a global component ($gbest$) using Equations 2 and 3.

$$X_{t+1} = X_t + V_{t+1} \quad (2)$$

$$V_{t+1} = wV_t + c_1r_1(pbest(t) - X_t) + c_2r_2(gbest(t) - X_t) \quad (3)$$

where; w is the inertia weight, c_1 and c_2 are the acceleration coefficients. r_1 and r_2 are two random parameters designed to affect the cognitive and social components of each particle.

The detailed optimization process of the first level L_1 is given in Figure 2. If the fitness function has changed, the lower level (L_2) is started in a parallel execution with L_1 (see Algorithm 3). The fitness function of each particle is a DMOP with a cyclic time-varying parameter T presented in Equation 4, and changed at each periodic frequency τ_t which is fixed to 5, 10 and 20 indicating severe, moderate and slight environmental changes. After some periodic time (in this case is equal to the frequency τ_t), the dynamic change is checked by comparing the current population $POS(t)$ with the previous one $POS(t-1)$, more details are presented in the subsection C.

$$T = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \quad (4)$$

where; n_t , and τ are the severity of the change and the iteration counter respectively, $\forall t \in T_{max}$.

In each optimization step, all non-dominated solutions x_i^* in L_1 are stored in the leader's archive A , presenting the best approximated solutions to the true POE. Furthermore, the leaders archive A is a shared component between two levels, and used to select the best global leader ($gbest$).

B. DPb-MOPSO: Lower Level (L_2)

If the fitness function has changed during the optimization process of L_1 (see Algorithm 2), the lower level (L_2) starts with

parallel execution to the first level L_1 . The second level has a distributed architecture based on the parallel execution of multiple sub-swarms constructed based on the population P' extracted from L_1 . Then, the Pareto ranking operator is applied to subdivide P' into K sub-swarms of different sizes, where K sub-swarms are a set of fronts denoted by $F_0, \dots, F_{K-1}$.

Algorithm 2: DPb-MOPSO upper level (L_1)

Input: P (swarm of N particles), N (size of the population), T_{max} (max-iterations)

Output: Set of non-dominated solutions in the leader's archive A

1. Swarm initialization (see Algorithm 1)
2. while $t \leq T_{max}$ do
3. if the fitness function has changed then
4. start the level L_2 in the Algorithm 3;
5. end if
6. for $i = 1$ to N do
7. Select the leader g_{best} using the epsilon dominance comparator among two random particles in the archive A ;
8. Evaluate the fitness function F of each p_i ;
9. Current population $\leftarrow$ record the previous solutions set $POS(t-1)$;
10. Determine p_{best} and g_{best} ;
11. Update the positions X_i and the velocity V_i using equations 2 and 3;
12. // Apply the dynamism handling strategy.
13. for each interval of time = τ_t do
14. //Select detectors.
15. Current population $\leftarrow$ 10% of $POS(t)$ as random detectors;
16. Previous population $\leftarrow$ 10% of their projection in previous $POS(t-1)$;
17. for $i = 0$ to $|POS(t)|$ do
18. //Change detection strategy.
19. Compare fitness function of $POS(t)$ vs. $POS(t-1)$ using strong epsilon dominance operator;
20. if $f_i(t) > f_i(t-1)$ then
21. Negative change $\leftarrow$ true;
22. Select $POS(t)$ with negative changes;
23. Select $POS(t)$ with positive changes;
24. //Response strategy.
25. Re-initialize all solutions with negative changes in $POS(t)$;
26. Construct population $POS(t+1)$ using positive $POS(t)$;
27. Update A ;
28. end if
29. end for;
30. end for
31. Update the leader's archive A ;
32. $t \leftarrow t+1$;
33. end while;
34. Return the best non-dominated solutions from the archive A ;

Therefore, parameters configuration of each sub-swarm is a primordial step to ensure equality of size between all sub-swarms. The parameters setting of each sub-swarm (size, and the additional injected solutions) is done using a set of additional individuals which are randomly selected from the solution set $POS(t)$ performed in L_1 and injected into other fronts. Moreover, all K sub-swarms are optimized in parallel until the maximum number of iterations (T_{max}), performing the same process in Figure 2. All non-dominated solutions of L_2 from K sub-swarms are stored in the archive A . At the end of

the two optimization levels, the best non-dominated solutions in archive A are generated as the output of the proposal.

Algorithm 3: DPb-MOPSO lower level (L_2)

Require: P' (population of L_1), n (size of the front), T_{max} (maximum iterations), k (number of ranks);

Output: A (leader's archive);

1. Select the intermediate population P' of L_1 ;
 2. Apply the pareto ranking operator to subdivide P' into a set of fronts ranked from 0 to $K-1$ to construct multiple sub-swarms F_K ;
 3. Adjust the size of each sub-swarm F_K equal to n ;
 4. while $t \leq T_{max}$ do
 5. for $K = 0$ to $K-1$ do
 6. Parallel execution of DPb-MOPSO process in Figure 2.
 7. end for
 8. end while
 9. Store the non-dominated solutions on A ;
 10. Return the best non-dominated solutions in the archive A ;
-

C. DPb-MOPSO: Dynamic Changes Handling Strategy

Dynamic changes detection aims essentially to help a better search process and to prospect any interesting solutions. The two steps of dynamism handling of the proposed algorithm are detailed in lines 12 to 25 in the pseudo-code of Algorithm 2, and detailed as follows:

✓ Step 1: Environment detection strategy

The environmental changes are checked periodically at each frequency τ_t equal to 5, 10, and 20 showing severe, moderate, and slight changes respectively. All historical values of the previous population $POS(t-1)$ and the current $POS(t)$ are recorded at each iteration. By selecting 10% of the previous $POS(t-1)$ as random detectors, the comparison process starts after evaluation the fitness function F to obtain the new optimized population. Then, select 10% of the current mapping solutions in $POS(t)$ to control their evolution over time. The strong dominance operator of the Epsilon-Dominance method is used to compare the time-varying behavior of each particle between t and $t-1$. For all M objectives, if X_t is strictly better than X_{t-1} , ($X_t > X_{t-1}$), then the solution X_t strongly dominates the solution X_{t-1} .

✓ Step 2: Response change strategy

If the change is successfully identified, all solutions $POS(t)$ with negative changes are reinitialized. In order to assume that only solutions with positive effects have good diversity and convergence, all detectors with positive changes $POS(t)$ are transferred to build a new population at $t+1$. In addition, reassess the leader's profile (A) to update all non-dominant solutions and overcome the deterioration of research capabilities.

Fig. 1. The Distributed Flowchart of the Proposed Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) Algorithm.

Fig. 2. The Detailed Steps of DPb-MOPSO in the Upper and Lower Levels.

IV. EXPERIMENTAL STUDY: PRELIMINARIES

This section presents the empirical study referred to the contributions [17] and [21]. The proposed DPb-MOPSO approach is compared to six transfer learning-based methods [17], five DMOEAs [21], and four MOPSO-based approaches [19], [16], [18], [25]. All algorithms are executed during 30 independent runs. Each run is executed during the maximum number of iterations $T_{max} = 3 \times n_t \times \tau_t + 50$, where; τ_t is the frequency of change fixed to 5, 10 and 20 respectively, and n_t is the severity of the change fixed to 10. The Java implementation of the proposed method is done using the jMETAL framework [39] on a personal computer with 8 Go of Ram, 1 To, and i7 intel processor. Experimental settings are summarized in Table II.

TABLE II. PARAMETERS SETTINGS OF MOEAs.

MOPSO, dMOPSO, pbMOPSO, Dynamic-MOPSO and DPb-MOPSO		
Perturbation index= 1.0	Mutation probability= 1.0/number of variables	r1 and r2= rand (0.0, 1.0) c1 and c2= rand (1.5, 2.0) w= rand (0.1, 0.5)
dCOEA, PPS, MOEAD, SGEA, DNSGA-II, TL-methods		
Crossover probability=1.0	Mutation distribution= 20	Crossover distribution= 20
Common parameters for all algorithms		
Swarm size=100	Archive size=100	Max-iterations= $3 \times n_t \times \tau_t + 50$

A. Dynamic Multi Objective Problems

This study concerns 21 benchmarks which are made of: five FDA [1], three dMOP [23], six F(ZJZ) [24], and seven UDF

functions [29]. Table III lists all DMOPs parameter settings and the dynamic nature of POS and POF.

B. Quality Indicators:

IGD, MIGD and HVD are the three quality indicators that have been considered for convergence and diversity measurement, when the smallest value is the best. All metrics are chosen referred to the compared state-of-the-art methods. Jiang *et al.* [21] has considered IGD and HVD quality indicators to compare four DMOEAs (DNSGA-II, dCOEA, PPS, MOEA/D) with their proposed SGEA system. However, six transfer-learning methods [17] have been compared using MIGD indicator. The mathematical definition of all metrics is detailed as follows:

- IGD is calculated using Equation 5 to measure the minimum Euclidean distance d between i th points in the generate POF and the true POF^* .

$$IGD(POF^*, POF) = \frac{\sum_{i \in POF^*} d(i, POF)}{|POF^*|} \quad (5)$$

- MIGD is the average of the IGD values computed using Equation 6.

$$MIGD(POF_i^*, POF_i) = \frac{1}{T} \sum_{i \in T} IGD(POF_i^*, POF_i) \quad (6)$$

- The HVD indicator is designed to measure the difference hypervolume between the obtained POF^* to the true POF^* and presented in Equation 7.

$$HVD = HV(POF_i^*) - HV(POF_i) \quad (7)$$

TABLE III Characteristics and Parameter Settings of DMOPs

DMOPs		Nb.Obj.	Nb.Var.	Type	POS Change Nature	POF Change Nature	
						Shift	Curvature
FDA problems, Farina <i>et al.</i> [1]	FDA1	2	20	I	sinusoidal change + vertical shift	no change	no change + convex
	FDA2	2	15	II		vertical + density of solutions changes	cyclic change: convex to concave
	FDA3	2	30	II		vertical + spread of solutions change	cyclic change: convex
	FDA4	3	12	I		no change + spread of solutions change	no change + nonconvex
	FDA5	3	12	II		radius variations + dynamic spread	change + nonconvex
dMOP problems, Goh <i>et al.</i> [23]	dMOP1	2	10	III	no change	vertical	change: convex to concave
	dMOP2	2	10	II	sinusoidal change + vertical shift	vertical	change: convex to concave
	dMOP3	2	10	I	sinusoidal change + vertical shift	no change	no change + convex
ZJZ problems, Zhou <i>et al.</i> [24]	F5	2	20	II	trigonometric + vertical shift	vertical	change: convex to concave
	F6	2	20	II			
	F7	2	20	II			
	F8	3	20	II			
	F9	2	20	II			
	F10	2	20	II			
UDF problems, Biswas <i>et al.</i> [29]	UDF1	2	10	I	trigonometric+ vertical shift	diagonal	no change + linear + continuous
	UDF2	2	10	I	polynomial+ vertical shift	vertical	no change + linear + continuous
	UDF3	2	10	III	trigonometric + no variation	diagonal	linear + discontinuous
	UDF4	2	10	II	trigonometric + horizontal shift	angular shift	convex to concave + continuous
	UDF5	2	10	II	polynomial + vertical shift	angular shift	change: convex to concave + continuous
	UDF6	2	10	III	trigonometric + no variation	diagonal + angular	linear + discontinuous
	UDF7	3	10	III	trigonometric + no variation	shifting of the center	radius concave + 3D

V. EXPERIMENTAL STUDY: RESULTS AND STATISTICAL ANALYSIS

This section aims to analyse the quantitative results of IGD, MIGD and HVD indicators. The comparative analysis based on mean and standard deviation are reported in the supplemental material (see Tables I-V). Important values are highlighted in bold and gray color. The inferential statistical concept of Friedman's two-way Analysis of Variance (ANOVA) method is used to determine the significance level between all tested MOEAs.

A. Results Analysis through Mean and Standard Deviation

Different configurations were considered on DMOEAs to demonstrate the impact of changing frequency when solving DMOPs. These configurations are implemented for severe, moderate, and slight changes based on the constant value of the severity of change n_t and the variation of the frequency τ_t .

1) Results on UDF and F(ZJZ) Problems

Experimental results in Table I, II (see supplemental material) have shown that DPb-MOPSO is better than DNSGA-II, dCOEA, PPS, MOEA/D, SGEA, MOPSO, dMOPSO, pbMOPSO methods for solving the majority of UDF and F(ZJZ) functions using both IGD and HVD metrics. The 13 functions are classified as follows: type I (UDF1, UDF2), type II (F5-F10, UDF4, UDF5), and type III (UDF3, UDF6, UDF7). In order to show statistically reliable conclusion, we use boxplots of the one-way ANOVA test in Figure 3. We can conclude that, compared with Dynamic-MOPSO, DPb-MOPSO is a competitive system with the best convergence on IGD and the best distribution on HVD. In details, the best result of DPb-MOPSO on IGD are 8/13 (UDF1, UDF2, UDF4, UDF5, UDF6, F5, F6, F8) and 7/13 (UDF1, UDF2, UDF4, UDF5, UDF6, F6, F8) are on HVD, while Dynamic-MOPSO is 3/13 (F7, F9, F10) on IGD and 5/13 (F5, F7, F9, F10, UDF7) on HVD. When solving UDF3 on HVD and UDF7 on IGD, there is a slight difference between DPb-MOPSO and pbMOPSO. Compared with the dMOPSO system, DPb-MOPSO only has poor convergence for the UDF3 function on the IGD measurement.

2) Results on FDA and dMOP Problems

In the case of 8 FDA and dMOP functions, all quantitative results of MIGD, IGD and HVD indicators are reported in Tables II, III, IV, V respectively (see supplemental material) with severe, moderate and slight changes. In order to facilitate the analysis of the results, the boxplots of the one-way ANOVA test are considered in Figures 4, 5, and 6. In this part of the study, 8 FDA and dMOP benchmarks were tested. Most FDA and dMOP functions belong to type I (FDA1, FDA4, dMOP3) or types II (FDA2, FDA3, FDA5, dMOP2), and only dMOP1 belongs to type III. The one-way ANOVA results in Figure 4 determine the importance of DPb-MOPSO compared with MMTL-MOEAD, KF-MOEAD, PPS-MOEAD, SVR-MOEAD, Tr-MOEAD, RI-MOEAD, MOPSO, dMOPSO, pbMOPSO and Dynamic-MOPSO to solve 8 DMOPs through MIGD metric. Figures 5 and 6 have shown the boxplots of the one-way ANOVA test of the IGD and the HVD quality indicators. Compared with DNSGA-II, dCOEA, PPS, MOEA/D, SGEA, MOPSO, dMOPSO, and pbMOPSO, the DPb-MOPSO method is the best to solve 4/8 (FDA2, FDA4, FDA5, dMOP1), 5/8 (FDA2, FDA3, FDA4, FDA5, dMOP1), and 5/8 (FDA1, FDA3, FDA4, FDA5, dMOP1) respectively in the MIGD indicator of severe, moderate and slight environmental changes. MOPSO, Dynamic-MOPSO and MMLT-MOEAD only have good results

when solving 2/8 (FDA1, dMOP2), 1/8 (FDA3) and 1/8 (dMOP3) respectively on the MIGD with severe environmental changes. However, MOPSO, dMOPSO and MMLT-MOEAD only have the best mean values for FDA1, dMOP2 and dMOP3 on the MIGD with moderate changes. For slightly change, Dynamic-MOPSO, dMOPSO and MMTL-MOEAD are the best choices for FDA2, dMOP2 and dMOP3 respectively. Using IGD metric with severe changes, DPb-MOPSO aims to solve 4/8 (FDA2, FDA3, FDA4, dMOP1), MOPSO has 1/8 (FDA1), pbMOPSO has 2/8 (FDA5, dMOP2) and dCOEA has 1/8 (dMOP3). For moderate changes, we can conclude that 4/8 (FDA3, FDA4, FDA5, dMOP3) functions are resolved using DPb-MOPSO, 1/8 (FDA1) uses MOPSO, 1/8 (FDA2) uses DNSGA-II, 1/8 (dMOP2) uses pbMOPSO and 1/8 (dMOP3) uses dCOEA. For slight changes, we can find 4/8 (FDA3, FDA4, FDA5, dMOP3) benchmarks are resolved using DPb-MOPSO, 2/8 (FDA1, FDA2) use MOPSO, 1/8 (dMOP2) use pbMOPSO and 1/8 (dMOP3) use dCOEA. Compared with MOPSO which aims to solve 1/8 (FDA1), SGEA 1/8 (FDA2), pbMOPSO 1/8 (dMOP2) and dCOEA 1/8 (dMOP3), the DPb-MOPSO system can solve 4/8 (FDA3, FDA4, FDA5, dMOP1) functions for the 3 kinds of environmental changes using HVD.

B. Discussion through Friedman two-way ANOVA Test

The use of the mean and standard deviation values of the quality indicators are insufficient to evaluate the performance of swarm intelligence algorithms and highlight their significance level. For this reason, Friedman's two-way ANOVA ranking test and its Post-hoc procedure [40] are used to analyze the results of different environmental changes on IGD, MIGD and HVD. Friedman's two-way ANOVA test is a non-parametric alternative to repeated-measures ANOVA. The test does not assume a normal distribution to compare multiple treatments and determine the level of significance between algorithms. First, define the null hypothesis $H_0: \mu_1 = \mu_i$ (there is no difference between the means), and the alternative hypothesis $H_1: \mu_1 \neq \mu_i$ (there is a difference). If the P-value $\leq \alpha$ indicates the statistical significance level, then reject the null hypothesis, otherwise we accept H_1 , where Alpha (α) is equal to 0.05. Friedman's two-way ANOVA ranking is calculated by assigning a ranking value (τ_i) to each algorithm i while; 1 is the best result to k , and k is the worst result. For a tie case, we assign the average of the rankings. The decision rule is defined as the critical value (CV) determined according to the chi-Square (χ^2) table¹. The CV value is fixed according to the degree of freedom (df) equal to $k - 1$, where k is the number of comparison algorithms and $\alpha = 0.05$. If the calculated χ^2 is greater than CV, we will reject the null hypothesis. In addition, if the calculated P-value is less than 0.05, we assume that there is a significant difference between the test algorithms.

1) Friedman rankings test

According to Tables IV and V, we can conclude that DPb-MOPSO is a winner using IGD for solving FDA, dMOP functions for severe and slight environmental changes. However, when using the IGD, HVD, and MIGD quality indicators, it ranks higher in addressing the moderate environmental changes of 21 DMOPs (FDA, dMOP, UDF, and ZJZ (F)). Whereas, assuming that the overall statistical difference between all tested algorithms is significant, all calculated χ^2 are greater than CV. At the 0.05 level of significance, all means are not equal. Almost, most P-values less than 0.05 represent the assumption of rejecting the null

¹ <http://uregina.ca/~gingrich/appchi.pdf>.

hypothesis and assuming that some means are not exactly equal and there are significant differences. Furthermore, the null hypothesis that there is no difference between means is retained for slight changes using the HVD indicator with a P-value=0.207, which is greater than 0.05. Friedman's ranking test detected significant difference between multiple comparisons. However, it is difficult to determine which group pairs are significantly different. Therefore, Friedman $1 \times N$ ANOVA multiple comparisons were considered for the results discussion, aiming to compare DPb-MOPSO as a control method with other MOEAs.

TABLE IV. Friedman's ANOVA Mean Ranks on IGD and HVD for FDA, dMOP, F (ZJZ) and UDF.

Mean Ranks over IGD and HVD						
Dynamic changes	Severe ($\tau_1=5, n_1=10$)		Moderate ($\tau_1=10, n_1=10$)		Slight ($\tau_1=20, n_1=10$)	
DMOPs	FDA, dMOP		FDA, dMOP, F(ZJZ) UDF		FDA, dMOP	
QI	IGD	HVD	IGD	HVD	IGD	HVD
DNSGA-II	8.94	7.25	7.29	6.40	8.25	6.00
dCOEA	6.44	5.38	7.98	6.52	7.00	6.50
PPS	8.00	7.25	7.19	6.12	8.00	6.00
MOEA/D	8.38	7.75	7.74	6.79	7.38	6.88
SGEA	6.13	4.25	6.10	4.40	5.13	4.25
MOPSO	3.63	5.94	5.40	7.29	3.19	6.06
dMOPSO	5.50	6.38	6.31	7.40	5.31	6.81
pbMOPSO	2.63	4.00	3.21	4.45	5.00	5.00
Dynamic-MOPSO	3.25	3.63	1.95	2.83	3.25	4.25
DPb-MOPSO	2.13	3.19	1.83	2.79	2.50	3.25
Chi-Square χ^2	48.77	21.69	113.70	63.19	34.42	12.11
P-value ($\alpha = 0.05$)	1.9E-7	0.01	2.6E-2	3.3E-1	8.0E-5	0.207
Statistically Significant	Yes	Yes	Yes	Yes	Yes	No
Critical value	16.92					
df	9					

TABLE V. Friedman's ANOVA Mean Ranks on MIGD for FDA, dMOP.

Mean Ranks over MIGD			
DMOPs	FDA, dMOP		
Dynamic changes	Severe ($\tau_1=5, n_1=10$)	Moderate ($\tau_1=10, n_1=10$)	Slight ($\tau_1=20, n_1=10$)
MMTL-MOEA/D	5.63	5.50	5.50
KF-MOEA/D	9.13	8.63	7.88
PPS-MOEA/D	8.00	7.25	8.13
SVR-MOEA/D	8.50	8.00	8.13
Tr-MOEA/D	9.00	8.88	8.88
RI-MOEA/D	8.88	9.13	8.38
MOPSO	3.19	4.13	4.19
dMOPSO	4.56	6.13	5.06
pbMOPSO	3.00	3.50	4.38
Dynamic-MOPSO	3.75	2.88	3.00
DPb-MOPSO	2.38	2.00	2.50
Chi-Square χ^2	54.29	48.21	39.85
P-value ($\alpha = 0.05$)	4.29E-8	5.69E-7	1.80E-5
Statistically Significant	Yes	Yes	Yes
Critical value	18.31		
df	10		

2) Friedman $1 \times N$ ANOVA Test

The Friedman Post-hoc procedure was performed to compare group means, and DPb-MOPSO was the control method. The Friedman Post-hoc procedure process results in a P-value (PV) at the 0.05 significance level that determines the degree of rejection of the null hypothesis. In Table VI, we first consider the PV over MIGD metric and compare DPb-MOPSO with six methods based on transfer learning and four methods based on MOPSO. Note that the sign (+) means that DPb-MOPSO is the best algorithm, and ($\cong$) the same significant level is determined. Compared with KF-MOEA/D, Tr-MOEA/D, RI-MOEA/D, SVR-MOEA/D, PPS-MOEA/D and MMTL-MOEA/D, the importance of the DPb-MOPSO system can be observed with P-values of all environment changes are all less than 0.05. However, it has the same level of significance compared with

MOPSO, pbMOPSO and Dynamic-MOPSO. In addition, we only retained the null hypothesis of dMOPSO system on moderate changes, and compared with DPb-MOPSO, it has the same significance for severe and slight changes and MMTL-MOEA/D with slight changes. The PV values over IGD presents the significant difference of DPb-MOPSO compared to DNSGA-II, MOEA/D, PPS, and dCOEA for all the types of changes. In addition, we can infer that there is no difference in p-value greater than 0.05, for severe and slight changes between DPb-MOPSO compared with dMOPSO, MOPSO, pbMOPSO and Dynamic-MOPSO. Furthermore, the computed P-values on HVD metric are greater than 0.05, we retain the null hypothesis of no difference between mean values for the nine compared methods with slight changes. However, the DPb-MOPSO system performs six DMOEAs (DNSGA-II, MOEA/D, PPS, dCOEA, dMOPSO, MOPSO) in solving DMOPs with moderate changes except for the SGEA, pbMOPSO and Dynamic-MOPSO approaches have the same importance as the proposal.

TABLE VI. P-values obtained by the post hoc method over the results of Friedman procedure with $\alpha = 0.05$ over MIGD, IGD and HVD.

QI	DPb-MOPSO vs.	(τ_i, n_i)		
		(5, 10)	(10, 10)	(20, 10)
MIGD	KF-MOEA/D	$\leq 0.001 +$	$\leq 0.001 +$	$0.001 +$
	Tr-MOEA/D	$\leq 0.001 +$	$\leq 0.001 +$	$\leq 0.001 +$
	RI-MOEA/D	$\leq 0.001 +$	$\leq 0.001 +$	$\leq 0.001 +$
	SVR-MOEA/D	$\leq 0.001 +$	$\leq 0.001 +$	$0.001 +$
	PPS-MOEA/D	$0.001 +$	$0.002 +$	$0.001 +$
	MMTL-MOEA/D	$0.050 +$	$0.035 +$	$0.070 \cong$
	dMOPSO	$0.187 \cong$	$0.013 +$	$0.122 \cong$
	MOPSO	$0.624 \cong$	$0.200 \cong$	$0.309 \cong$
	pbMOPSO	$0.706 \cong$	$0.366 \cong$	$0.258 \cong$
	Dynamic-MOPSO	$0.407 \cong$	$0.598 \cong$	$0.763 \cong$
IGD	DNSGA-II	$\leq 0.001 +$	$\leq 0.001 +$	$\leq 0.001 +$
	MOEA/D	$\leq 0.001 +$	$\leq 0.001 +$	$0.001 +$
	PPS	$\leq 0.001 +$	$\leq 0.001 +$	$\leq 0.001 +$
	dCOEA	$0.004 +$	$\leq 0.001 +$	$0.003 +$
	SGEA	$0.008 +$	$\leq 0.001 +$	$0.083 \cong$
	dMOPSO	$0.026 +$	$\leq 0.001 +$	$0.063 \cong$
	MOPSO	$0.322 \cong$	$\leq 0.001 +$	$0.650 \cong$
	pbMOPSO	$0.741 \cong$	$0.139 \cong$	$0.099 \cong$
HVD	Dynamic-MOPSO	$0.457 \cong$	$0.899 \cong$	$0.620 \cong$
	DNSGA-II	$0.007 +$	$\leq 0.001 +$	The Friedman test retained the null hypothesis of no difference between mean values with P-value=0.207 > 0.05
	MOEA/D	$0.003 +$	$\leq 0.001 +$	
	PPS	$0.007 +$	$\leq 0.001 +$	
	dCOEA	$0.148 \cong$	$\leq 0.001 +$	
	SGEA	$0.483 \cong$	$0.083 \cong$	
	dMOPSO	$0.035 +$	$\leq 0.001 +$	
	MOPSO	$0.069 \cong$	$\leq 0.001 +$	
	pbMOPSO	$0.591 \cong$	$0.074 \cong$	
	Dynamic-MOPSO	$0.773 \cong$	$0.959 \cong$	

+: DPb-MOPSO is the best algorithm, $\cong$: DPb-MOPSO has the same significant level compared to other MOEAs (significance level is 0.05).

In conclusion, and in regards to the state-of-the-art methods, the proposed DPb-MOPSO has shown its importance compared with other MOPSO-based methods. In addition, compared with the new DPb-MOPSO, the MOPSO-based system is the most competitive. But they failed to resolve multiple DMOPs, thus demonstrating the utility and importance of using dynamic processing strategies for two-stage optimization to detect and effectively respond to changes. However, the DPb-MOPSO algorithm also failed in some test problems with the characteristics of time-varying spread, dynamic density of the solution, shifting of the center point and a discontinuous POF. the DPb-MOPSO may requires an additional mechanism to detect the dynamic density of the POS and effectively respond to it and assume good distribution using the HVD measurement.

One-way ANOVA Results in a Boxplot of 10 MOEAs on IGD for UDF, ZJZ(F) with Moderate Environmental Changes.

(a)

One-way ANOVA Results in a Boxplot of 10 MOEAs on HVD for UDF, ZJZ(F) with Moderate Environmental Changes.

(b)

Fig. 3. ANOVA Boxplot on (a) IGD and (b) HVD with Moderate Environmental Changes for UDF and F(ZJZ) functions.

One-way ANOVA Results in a Boxplot of 10 MOEAs on IGD for FDA, dMOP with Severe Environmental Changes.

(a)

One-way ANOVA Results in a Boxplot of 10 MOEAs on IGD for FDA, dMOP with Moderate Environmental Changes.

(b)

One-way ANOVA Results in a Boxplot of 10 MOEAs on IGD for FDA, dMOP with Slight Environmental Changes.

(c)

Fig. 5. ANOVA Multiple Comparison on IGD with (a) Severe, (b) Moderate, and (c) Slight Environmental Changes for FDA and dMOP functions

One-way ANOVA Results in a Boxplot of 11 MOEAs on MIGD for FDA, dMOP with Severe Environmental Changes.

(a)

One-way ANOVA Results in a Boxplot of 11 MOEAs on MIGD for FDA, dMOP with Moderate Environmental Changes.

(b)

One-way ANOVA Results in a Boxplot of 11 MOEAs on MIGD for FDA, dMOP with Slight Environmental Changes.

(c)

Fig. 4. ANOVA Multiple Comparison on MIGD with (a) Severe, (b) Moderate, and (c) Slight Environmental Changes for FDA and dMOP functions.

One-way ANOVA Results in a Boxplot of 10 MOEAs on HVD for FDA, dMOP with Severe Environmental Changes.

(a)

One-way ANOVA Results in a Boxplot of 10 MOEAs on HVD for FDA, dMOP with Moderate Environmental Changes.

(b)

One-way ANOVA Results in a Boxplot of 10 MOEAs on HVD for FDA, dMOP with Slight Environmental Changes.

(c)

Fig. 6. ANOVA Multiple Comparison on HVD with (a) Severe, (b) Moderate, and (c) Slight Environmental Changes for FDA and dMOP functions.

VI. STABILITY ANALYSIS USING LYAPUNOV THEOREM

The stability analysis is an important aspect of analyzing the robustness of dynamic systems. The Lyapunov theorem proposed by Alexandr Mikhailovich Lyapunov [41] is a well-known stability analysis mechanism. The theorem aims to measure the growth of the initial values over time, and the small differences from one instance to another is called the Lyapunov Exponent (LE). In addition, LE can grow to a very large differences to indicate the speed at which two initially close dynamics diverge or converge. A positive value of LE indicates divergence, while a negative value indicates convergence in phase space.

In [42], the difference between two points t_i and t_j is denoted by $t_i, t_{i+1}, \dots, t_{i+n}$ and $t_j, t_{j+1}, \dots, t_{j+n}$. The distance $d(k)$ between the two sequences after k steps is calculated using the following formula: $d(k) = |t_{i+k} - t_{j+k}|$ to measure the convergence and divergence using the Lyapunov Exponent. If the system is chaotic, $d(k)$ will rise exponentially with k . Therefore, we can compute and plot $\ln d(k)$ vs k to estimate the Lyapunov Exponent (LE). When the value of LE is very small and close over time, this state indicates that the system is stable and reaches the same solution over time.

In this study, we considered a set of IGD values of 30 independent runs of the DPb-MOPSO algorithm, where n_t and τ_t are equal to 10. The independent IGD values are calculated using the smallest Euclidean distance d_i between i th points in the generated POF and the true POF^* over time t . The stability analysis through Lyapunov Exponent is computed using $\ln(IGD())$. Figure 7 shows the spectrum of the Lyapunov Exponents of F, UDF, FDA, and dMOP functions.

Fig. 7. Spectrum Lyapunov Exponent of DPb-MOPSO for F(ZJZ), UDF, FDA and dMOP Problems.

Moreover, we can conclude that the Lyapunov exponents for all tested problems are under zero with negative values

assuming that the system converges over time. The spectrum of the Lyapunov exponent indicates the stability of DPb-MOPSO with negative values of LE. All location points determine the fast convergence of DPb-MOPSO to all tested problems.

VII. CONCLUSION

This study presents a novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) algorithm for solving different types of DMOPs (I, II and III). The proposed DPb-MOPSO has a distributed architecture with two optimization levels, including dynamic handling strategy that effectively detects and responds to changes. In contrast to several MOEAs, Friedman's two-way ANOVA test is used to evaluate the proposed DPb-MOPSO. Based on statistical analysis (at 0.05 significance level), DPb-MOPSO proved to be a more robust algorithm because it can ensure a good trade-off between convergence and diversity in a time-dependent environment. Further, the spectrum of Lyapunov exponents has shown to be an efficient tool for analyzing periodic motions and stability analysis in dynamical systems. The superiority of the proposed method is observed when handling DMOPs with two or three objectives including different types of changes in POS and POF. In conclusion, this study proves the novelty and robustness of DPb-MOPSO in solving dynamic multi-objective problems. However, these characteristics have yet to be explored for problems with more than three objectives. For future work, we will further enhance the proposed approach for DMOPs with dynamic spread and time-varying density of the solutions as well as for pertinent feature selection in real-world problem.

ACKNOWLEDGMENT

We deeply acknowledge Taif University for Supporting this study through Taif University Researchers Supporting Project number (TURSP-2020/327), Taif University, Taif, Saudi Arabia. The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

REFERENCES

- [1] 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.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Nov. 1995, vol. 4, pp. 1942–1948 vol.4, doi: 10.1109/ICNN.1995.488968.
- [4] A. Song, W. N. Chen, T. Gu, H. Yuan, S. Kwong, and J. Zhang, "Distributed Virtual Network Embedding System with Historical Archives and Set-Based Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 927–942, Feb. 2021, doi: 10.1109/TSMC.2018.2884523.
- [5] Y. Wang and L. Li, "Heterogeneous redundancy allocation for series-parallel multi-state systems using hybrid particle swarm optimization and local search," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 2, pp. 464–474, Mar. 2012, doi: 10.1109/TSMCA.2011.2159585.
- [6] H. Chen, M. Li, and X. Chen, "Using diversity as an additional-objective in dynamic multi-objective optimization algorithms," *2nd International Symposium on Electronic Commerce and Security, ISECS 2009*, vol. 1, pp. 484–487, 2009, doi: 10.1109/ISECS.2009.42.

- [7] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Applied Soft Computing*, vol. 8, no. 2, pp. 959–971, Mar. 2008, doi: 10.1016/J.ASOC.2007.07.005.
- [8] C. A. Liu and Y. Wang, "Dynamic multi-objective optimization evolutionary algorithm," *Proceedings - Third International Conference on Natural Computation, ICNC 2007*, vol. 4, pp. 456–459, 2007, doi: 10.1109/ICNC.2007.340.
- [9] A. Aboud, R. Fdhila, and A. M. Alimi, "Dynamic Multi Objective Particle Swarm Optimization Based on a New Environment Change Detection Strategy," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nov. 2017, vol. 10637 LNCS, pp. 258–268. doi: 10.1007/978-3-319-70093-9_27.
- [10] K. Deb, U. B. Rao N., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4403 LNCS, pp. 803–817. doi: 10.1007/978-3-540-70928-2_60.
- [11] A. Carlisle and G. Dozier, "Adapting Particle Swarm Optimization to Dynamic Environments," *undefined*, 2001.
- [12] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: Detection and response to dynamic systems," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, pp. 1666–1670, 2002, doi: 10.1109/CEC.2002.1004492.
- [13] B. Nasiri and M. R. Meybodi, "History-driven firefly algorithm for optimisation in dynamic and uncertain environments," *International Journal of Bio-Inspired Computation*, vol. 8, no. 5, pp. 326–339, 2016, doi: 10.1504/IJBIC.2016.079575.
- [14] A. Aboud, R. Fdhila, and A. M. Alimi, "MOPSO for dynamic feature selection problem based big data fusion," in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, Feb. 2017, pp. 3918–3923. doi: 10.1109/SMC.2016.7844846.
- [15] R. Azzouz, S. Bechikh, and L. ben Said, "Dynamic Multi-objective Optimization Using Evolutionary Algorithms: A Survey," *Adaptation, Learning, and Optimization*, vol. 20, pp. 31–70, 2017, doi: 10.1007/978-3-319-42978-6_2.
- [16] R. Fdhila, C. Walha, T. M. Hamdani, and A. M. Alimi, "Hierarchical design for distributed MOPSO using sub-swarms based on a population Pareto fronts analysis for the grasp planning problem," *13th International Conference on Hybrid Intelligent Systems, HIS 2013*, pp. 203–208, Oct. 2014, doi: 10.1109/HIS.2013.6920483.
- [17] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, "A Fast Dynamic Evolutionary Multiobjective Algorithm via Manifold Transfer Learning," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3417–3428, May 2020, doi: 10.1109/TCYB.2020.2989465.
- [18] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, pp. 1051–1056, 2002, doi: 10.1109/CEC.2002.1004388.
- [19] A. Aboud, R. Fdhila, and A. M. Alimi, "Dynamic Multi Objective Particle Swarm Optimization Based on a New Environment Change Detection Strategy," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nov. 2017, vol. 10637 LNCS, pp. 258–268. doi: 10.1007/978-3-319-70093-9_27.
- [20] R. Fdhila, C. Walha, T. M. Hamdani, and A. M. Alimi, "Hierarchical design for distributed MOPSO using sub-swarms based on a population Pareto fronts analysis for the grasp planning problem," *13th International Conference on Hybrid Intelligent Systems, HIS 2013*, pp. 203–208, Oct. 2014, doi: 10.1109/HIS.2013.6920483.
- [21] S. Jiang and S. Yang, "A Steady-State and Generational Evolutionary Algorithm for Dynamic Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, Feb. 2017, doi: 10.1109/TEVC.2016.2574621.
- [22] 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, Dec. 2007, doi: 10.1109/TEVC.2007.892759.
- [23] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009, doi: 10.1109/TEVC.2008.920671.
- [24] A. Zhou, Y. Jin, and Q. Zhang, "A Population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, Jan. 2014, doi: 10.1109/TCYB.2013.2245892.
- [25] K. B. Lee and J. H. Kim, "DMOPSO: Dual multi-objective particle swarm optimization," *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 3096–3102, Sep. 2014, doi: 10.1109/CEC.2014.6900464.
- [26] A. Muruganatham, K. C. Tan, and P. Vadakkepat, "Evolutionary Dynamic Multiobjective Optimization Via Kalman Filter Prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016, doi: 10.1109/TCYB.2015.2490738.
- [27] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary Dynamic Multiobjective Optimization Assisted by a Support Vector Regression Predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 305–319, Apr. 2020, doi: 10.1109/TEVC.2019.2925722.
- [28] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer Learning-Based Dynamic Multiobjective Optimization Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, Aug. 2018, doi: 10.1109/TEVC.2017.2771451.
- [29] 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 *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, Sep. 2014, pp. 3192–3199. doi: 10.1109/CEC.2014.6900487.
- [30] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 193–208, 2014, doi: 10.1109/TEVC.2013.2248159.
- [31] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3242, pp. 742–751, 2004, doi: 10.1007/978-3-540-30217-9_75.
- [32] S. Seifollahi-Aghmiuni and O. Bozorg Haddad, "Multi Objective Optimization with a New Evolutionary Algorithm," *Water Resources Management 2018 32:12*, vol. 32, no. 12, pp. 4013–4030, Jul. 2018, doi: 10.1007/S11269-018-2034-1.
- [33] M. Cámara, J. Ortega, and F. J. Toro, "Parallel processing for multi-objective optimization in dynamic environments," *Proceedings - 21st International Parallel and Distributed Processing Symposium, IPDPS 2007; Abstracts and CD-ROM*, 2007, doi: 10.1109/IPDPS.2007.370433.
- [34] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, "Benchmark Problems for CEC2018 Competition on Dynamic Multiobjective Optimisation," 2018.
- [35] R. Azzouz, S. Bechikh, and L. ben Said, "Multi-objective optimization with dynamic constraints and objectives: New challenges for evolutionary algorithms," *GECCO 2015 - Proceedings of the 2015 Genetic and Evolutionary Computation Conference*, pp. 615–622, Jul. 2015, doi: 10.1145/2739480.2754708.
- [36] R. Azzouz, S. Bechikh, and L. ben Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Computing 2015 21:4*, vol. 21, no. 4, pp. 885–906, Aug. 2015, doi: 10.1007/S00500-015-1820-4.
- [37] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007, pp. 3944–3951. doi: 10.1109/CEC.2007.4424985.
- [38] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, Sep. 2002, doi: 10.1162/106365602760234108.
- [39] J. J. Durillo and A. J. Nebro, "JMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, Oct. 2011, doi: 10.1016/j.advengsoft.2011.05.014.
- [40] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/J.SWEVO.2011.02.002.
- [41] A. M. LYAPUNOV, "The general problem of the stability of motion," <http://dx.doi.org/10.1080/00207179208934253>, vol. 55, no. 3, pp. 531–534, 2007, doi: 10.1080/00207179208934253.
- [42] B. B. and D. C., "Stochastic neural network model for spontaneous bursting in hippocampal slices," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 66, no. 5 Pt 1, p. 18, Nov. 2002, doi: 10.1103/PHYSREVE.66.051908.

Ahlem Aboud she is currently a PhD student in computer science in ISITCom, University of Sousse and with the REsearch Group on Intelligent Machines (REGIM-lab) at ENIS Sfax, University of Sfax, Tunisia. She received his engineering degree in 2015 from the ISIMS Sfax, Tunisia. She is currently IEEE student member since October 2015. His research interests include, dynamic multi-objective optimization problem, evolutionary computation, collective intelligence methods, and Machine Learning Algorithms.

Nizar Rokbani (IEEE SM), He is an Assit Prof of Industrial Computing with Institute of Applied Science and technology of Sousse, Tunisia where he is a lecturer since 2014. He graduated in Electrical Engineering from the National Engineering School of Tunis, ENIT in 1995. He obtained a Master degree in industrial computing in 2003 from the National Engineering School of Sfax, ENIS, and a PHD in Electrical Engineering in 2013. He was an automation process engineer until 1998, he was then a technologic trainer at the institute of technological studies of Gabes, occupied a post of training adviser engineer at the national agency of vocational training from 2003 to 2006. His research interests include applications of intelligent techniques such as Swarm intelligence, computational intelligence, fuzzy logic, evolutionary algorithms to robotic systems and industrial processes. Dr. Nizar Rokbani is an IEEE member & volunteer, He participated in funding several society chapters, including RAS and SMC 2009, and OES 2016, he is also the founding counselor of ISSAT Sousse Sb 2015, and Polytech Sousse Sb, 2016. He is The IEEE RAS Tunisia Chair 2015-20120, the chapter was awarded: R8 Outstanding Society Chapter 2016, Best RAS Society Chapter award for 2017, and Outstanding Society chapter award of Tunisia Section for 2017. In 2019, He received the the Meritorious Achievement Award in Pre-University Education of the IEEE Educational Activities Board (EAB) for his effort in promoting STEM & Robotics education in tunisia primary schools.

Raja Fdhila she holds an engineering degree in computer science and a Master's degree in automation and signal processing from the National School of Engineers of Tunis in 2005 and 2007 respectively. She obtained her doctorate in computer systems engineering from ENIS Sfax in 2017. She is member of the REGIM-lab at ENIS Sfax. She works mainly in the field of artificial intelligence, and is particularly interested in multi-objective optimization, collective intelligence, multi-agent systems. She has been an IEEE member since 2008.

Dr Abdulrahman Qahtani is currently Assistant professor in computer science department at Taif University. Dr. Qahtani received his PhD from Southampton University in 2015. He has extensive experience in software engineering and development process in a distributed domain. His research focused on customization process across organizational boundaries. Recently, he applied machine learning algorithms on software engineering data to predict time and cost estimation for multi-clients projects.

Omar Almutiry received the PhD degree from the University of Southampton, UK, in 2017. Since 2018, he has been with the Applied Computer Science College, King Saud University, Saudi Arabia, where he is currently working as an Assistant Professor with the Department of Applied Computer Sciences. He is also assistant to the General Director of the Almuzahmiyah branch for academic affairs and development. His research areas include data science, health informatics, deep learning applications in healthcare and medical fields.

Habib Dhahri was born in Sidi Bouzid, Tunisia in 1975. He graduated in Computer Science in 2001, obtained his PhD in Computer Science in 2013 from the National Engineering School of Sfax (ENIS). He is now an assistant professor in Computer Science at the King Saud University. His area of interest includes computational intelligence, soft computing techniques and Machine Learning for Healthcare Data. He has authored and co-authored more than 30 publications in journals and conferences. He also serves as a reviewer for several international scientific journals.

Amir Hussain he received his B.Eng (highest 1st Class Honours with distinction) and Ph.D degrees, from the University of Strathclyde, Glasgow, U.K., in 1992 and 1997, respectively. Following postdoctoral and academic positions at the Universities of West of Scotland (1996-98), Dundee (1998-2000) and Stirling (2000-18) respectively, he joined Edinburgh Napier University (in Scotland, UK) in 2018 as Professor and founding Head of the Data Science and Cyber Analytics (DSCA) Research Group (managing 20 academics and research staff). As part of the latter, he is also founding Head of the Cognitive Big Data and Cybersecurity (CogBiD) Research Lab

Adel M. Alimi (S'91–M'96–SM'00) was born in Sfax, Tunisia, in 1966. He received the Degree in electrical engineering in 1990, the Ph.D. degree in electrical and computer engineering from the Polytechnic University of Montreal, Montr' eal, QC, ' Canada, in 1995, and the HDR degree in electrical and computer engineering from the National Engineering School of Sfax, Sfax, in 2000. He is currently a Professor of electrical and computer engineering with the University of Sfax, Sfax. His current research interests include applications of intelligent methods (neural networks, fuzzy logic, and evolutionary algorithms) to pattern recognition, robotic systems, vision systems, industrial processes, intelligent pattern recognition, learning, analysis, and intelligent control of large-scale complex systems. Prof. Alimi is a member of IAPR, INNS, and PRS. He is a member and an Associate Editor of the Editorial Board of many international scientific journals, including Pattern Recognition Letters, Neurocomputing, Neural Processing Letters, International Journal of Image and Graphics, Neural Computing and Applications, International Journal of Robotics and Automation, and International Journal of Systems Science. He was a Guest Editor of several special issues of international journals, including Fuzzy Sets and Systems, Soft Computing, the Journal of Decision Systems, Integrated Computer-Aided Engineering, and Systems Analysis Modeling and Simulation.