Multi-objective Optimization with Dynamic Constraints and Objectives: New Challenges for Evolutionary Algorithms

Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said
SOIE Laboratory, University of Tunis
Bardo, Tunis, Tunisia
azouzradhia@gmail.com, slim.bechikh@gmail.com, lamjed.bensaid@isg.rnu.tn

ABSTRACT

Dynamic Multi-objective Optimization (DMO) is a challenging research topic since the objective functions, constraints, and problem parameters may change over time. Several evolutionary algorithms have been proposed to deal with DMO problems. Nevertheless, they were restricted to unconstrained or domain constrained problems. In this work, we focus on the dynamicty of problem constraints along with time-varying objective functions. As this is a very recent research area, we have observed a lack of benchmarks that simultaneously take into account these characteristics. To fill this gap, we propose a set of test problems that extend a suite of static constrained multi-objective problems. Moreover, we propose a new version of the Dynamic Non dominated Sorting Genetic Algorithm II to deal with dynamic constraints by replacing the used constraint-handling mechanism by a more elaborated and self-adaptive penalty function. Empirical results show that our proposal is able to: (1) handle dynamic environments and track the changing Pareto front and (2) handle infeasible solutions in an effective and efficient manner which allows avoiding premature convergence. Moreover, the statistical analysis of the obtained results emphasize the advantages of our proposal over the original algorithm on both aspects of convergence and diversity on most test problems.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence – Problem Solving, Control Method, and Search.

General Terms

Algorithms, Design.

Keywords

Constraint handling, Multi-objective optimization, Dynamic optimization, benchmarks, self-adaptive penalty function.

GECCO '15, July 11 - 15, 2015, Madrid, Spain
 ISBN 978-1-4503-3472-3/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754708

1. INTRODUCTION

DMO Problems (DMOPs) involve the simultaneous optimization of several competing objectives subject to a number of given constraints where the objective functions, constraints and/or dimensions of the objective space and the design space may change over time. Many real-world engineering problems make part of this category of problems such as dynamic multi-objective scheduling and transportation problems, robot navigation, traffic signal control, program selection planning management and so forth. When dealing with such problems, the main difficulty lies on the fact that the optimal Pareto Front (PF) and the optimal Pareto Set (PS) may move over time. Moreover, as the constraints are also time-changing, the feasibility area would not remain unvariant which makes the task of optimization more difficult. Having such complex dynamic behaviors, solving this kind of problems needs to deal with mainly these issues: (1) evolving a near-optimal and diverse PF, (2) preserving the required diversity to ensure a high level of flexibility and (3) driving infeasible individuals toward feasible ones.

Applying Evolutionary Algorithms (EAs) to solve dynamic optimization problems has obtained great attention among many researchers. To the best of our knowledge, the earliest application of EAs to dynamic environments dates back to 1966 [15]. However, it was not until the late 1980s that the subject becomes a research topic. EAs were first applied to dynamic single-objective optimization problems. Several approaches have been proposed such as memory-based approaches [23], multipopulation approaches [4], predictive approaches [19], etc. When dealing with DMOPs, the task of optimization is much more difficult. In contrast to the single-objective case, there are few works dealing with DMOPs which include change prediction approaches [1, 8], memory-based approaches [3], and parallel approaches [17].

In real world, we often encounter problems that not only involve the optimization of several conflicting objectives simultaneously but also have a set of constraint conditions that must be satisfied. Several constraint handling techniques have been developed to be incorporated into EAs. Most of them are restricted to the single objective case [29]. Adapting constraint handling techniques to the multi-objective case is not yet highly explored. Nevertheless, some works have been established such as penalty functions-based techniques [27, 16], techniques based on maintaining the feasibility of solutions [14, 25], techniques based on some special ranking rules [11, 26], etc.

Despite the growing interest given to the use of EAs to solve dynamic optimization problems, most of the research was focused on the unconstrained or domain constrained problems (which are both regarded in this paper as unconstrained problems). Applying EAs to solve constrained DMOPs is not yet highly explored although this kind of problems is of significant importance in practice. Many real-world problems are constrained DMOPs such as optimal control problems, portfolio investment, chemical engineering design like the dynamic hydro-thermal power scheduling problem, dynamic scheduling and transportation problems such as the dynamic multi-objective vehicle routing problems and so forth. In fact, when dealing with such problems, the main difficulties consist on the need to not only efficiently handle the constraints but also rapidly and continually track the changing PF and drive infeasible solutions to feasible ones whenever the constraints change.

Motivated by the lack of attention on constrained DMOPs, we propose in this paper a set of dynamic constrained benchmark functions where the PF, the PS and the constraints are simultaneously time-dependent. The proposed test functions extend the static test suite developed in [12]. Moreover, we propose a new version of a basic dynamic constrained EA, which is DNSGA-II [13], where we replace the constraint dominance principle used to handle constraints by an adaptive penalty function-based technique proposed in [27]. In fact, this constraint dominance principle prefers feasible solutions over infeasible ones which may lead to the loss of some useful information about the infeasible region. This may cause a lack of diversity, mainly in dynamic environments, which leads inevitably to a premature convergence. Moreover, the adaptive penalty function adopted in this work may be useful for DMOPs since it adaptively regulates the penalty term relatively to the proportion of feasible solutions in current population. Thereby, we suppose that no further explicit actions would be needed whenever a change affects the feasible area. The remainder of this paper is organized as follows. Section 2 presents the literature review on this research area. Section 3 presents the proposed dynamic constrained test functions. Section 4 describes the proposed algorithm and details its building-blocks. Section 5 describes the experimental study. Finally, section 6 concludes the paper and gives some avenues for future research.

2. LITERATURE REVIEW

2.1 EAs for dynamic multi-objective optimization problems

When dealing with DMOPs, the multi-objective EA should be capable of attaining a fast convergence which also implies a rapid loss of diversity during the optimization process. Deb et al. [13] extended NSGA-II to handle DMOPs by introducing diversity at each change detection. One of the merits of this work is that it can also solve constrained DMOPs. When the behavior of the dynamic problem follows a certain trend, a prediction model may be used to exploit past information and anticipate the location of the new optimal solutions. Several works have been established using this approach such as the Feed-forward Prediction Strategy (FPS) [8], the Population Prediction Strategy (PPS) proposed recently by Zhou et al. [1], etc. Since the prediction may not be always successful, there is a need to combine such approach with a maintaining diversity mechanism. Moreover, this approach is suitable to only dynamic environments presenting a behavior that follows a certain trend. Several Memory-based approaches have been proposed in the literature. In [3], authors have presented a co-evolutionary multi-objective algorithm based on competitive and cooperative mechanisms to solve DMOPs using an emergent problem decomposition. More recently, Wang and Li [28] proposed new dynamic multi-objective test problems and a new Multi-Strategy ensemble MOEA (MS-MOEA) where the convergence speed is accelerated using a new offspring generation mechanism based on adaptive genetic and differential operators. The main drawback of memory-based approaches is that memory is very dependent on diversity and should, thus, be used in combination with diversity-preserving techniques. Another kind of approaches which is parallel approaches has also been used to solve DMOPs. Cámara et al. have proposed [17] a procedure for the adaptation of the Parallel Single Front Genetic Algorithm (PSFGA) to dynamic environments where the sequential algorithm is decomposed into several tasks that are run on different data distributed between several processors. The main drawback of this work is its sensitivity to the data decomposition. In [24] authors developed a Dynamic Orthogonal MOEA called DOMOEA, which presents a generalization of the Orthogonal MOEA (i.e., OMOEA-II) to dynamic environments. One of the disadvantages of this approach is that the statistical method used has been proven to be optimal for only additive and quadratic models. More recently, Deb [10] presented two different approaches that are usually used when resolving dynamic single-objective as well as multi-objective optimization problems. Moreover, an automated decision-making approach based on the utility function concept has been proposed. The majority of these works are based on some techniques that either detect or predict changes. Nevertheless, sometimes it is difficult or even impossible to detect these changes. Azzouz et al. have proposed the Multiple Reference Point-based Multi-Objective EA (MRP-MOEA) [20] that deals with dynamic problems with undetectable changes.

2.2 EAs for static constrained multi-objective optimization problems

Over the last decade, several constraint handling techniques have been developed to be incorporated into EAs. Most of them are restricted to the single objective case. Michalewicz and Schoenauer [29], roughly classified these techniques into four categories: (1) techniques based on the use of penalty functions, (2) techniques based on the preservation of the feasibility of solutions, (3) techniques based on the separation of objectives and constraints and (4) hybrid methods. As noted in [7], adapting constraint handling techniques to the multi-objective case is not yet largely explored. Some works have been established. We will briefly survey in the following the most interesting ones. In fact, the simplest and the most commonly used method is the use of penalty functions. The main idea behind it is to add a constraint violation measure to the objective function as a penalty term. If the penalty coefficients change with the generation number, the penalty function is said to be a dynamic (adaptive) penalty function. In 2009, Woldesenbet et al. [27] introduced a constraint handling technique based on adaptive penalty functions and distance measures by extending the corresponding version for the single objective constrained optimization. More recently in 2014, Jiao [16] proposed a modified objective function method with feasible-guiding strategy on the basis of NSGA-II. In fact, the objective function values of an individual are modified using its constraint violation values relatively to a feasibility ratio which is fed back from current population. Moreover, a feasible-guiding strategy is adopted to make use of preserved infeasible individuals and to evolve them towards feasible directions. A second approach consists on preserving the feasibility of solutions either by generating feasible individuals, or removing infeasible region from the search space, or repairing infeasible individuals and recovering them to feasible ones. Harada et al. [14] proposed a few efficient constraint-handling guidelines and designed a pareto descent repair operator that searches for feasible solutions out of infeasible individuals in the constraint function space. More recently in 2009, an infeasibility driven EA was proposed [25] to handle constraints by maintaining a small percentage of infeasible solutions close to the constraint boundaries. To make a separation between objectives and constraints, Fonseca and Fleming [5] suggested to treat constraints as high-priority objectives. They proposed to handle constraints by assigning high priority to constraints and low priority to objective functions, which allows searching the feasible solutions before searching the optimal solutions. One other constraint-handling technique consists on using some special ranking schemes or comparison rules to rank individuals. Deb et al. [11] proposed a constrained multiobjective EA based on the concept of constrained domination, which is also known as superiority of the feasible solution. Feasible solutions constrained dominates all infeasible solutions. When two feasible individuals are compared, the Pareto dominance is used while the level of constraints violation is used when comparing two infeasible individuals. In [26] authors proposed using three different non-dominated rankings of the population. The first ranking is obtained using the objective function values, the second is performed using the different constraints and the third is based on the combination of all objective functions and constraints. Then the algorithm will perform according to the predefined rules. Suganthan et al. [2] proposed a set of constraint-handling techniques to solve constrained problems. In this method, each constraint-handling technique has its own population.

2.3 EAs for dynamic constrained multi-objective optimization problems

Although DMO and constrained multi-objective optimization have received a focus separately, very few works have been done in solving constrained DMOPs. These works have been established mainly using artificial immune systems [30] and particle swarm optimization [9]. However, to the best of our knowledge, when using EAs, very few studies are available. One basic dynamic EA is proposed by Deb [13], which uses the constraint dominance principle used in NSGA-II to deal with constraints. However, since this principle prefers feasible solutions over infeasible ones, it often results in a premature convergence due to the loss of diversity over time.

3. DYNAMIC CONSTRAINED TEST FUNCTIONS

Deb et al. [12] proposed a set of static constrained multi-objective optimization problems (MOPs), denoted as CTP test problems, that challenge algorithms with different difficulties. We propose to modify this test suite into dynamic constrained MOPs that we denote as Dynamic CTP (i.e., DCTP), as follows:

DCTP1.

$$
\begin{aligned}
&Min(f_1(x,\,t),\,f_2(x,\,t)) \qquad (1)\\
&f_1(x,\,t) = x_1\\
&f_2(x,\,t) = t * g(x,\,t) * exp(-f_1(x,\,t)/g(x,\,t))\\
&s.t.\ g(x,\,t) = 1 + \sum_{i=2}^{D}(x_i - sin(0.01\pi t))\\
&c_1(x) = (f_2(x,\,t)/t) - (0.858 * exp(-0.541 * f_1(x,\,t))\\
&c_2(x) = (f_2(x,\,t)/t) - (0.728 * exp(-0.295 * f_1(x,\,t))\\
&x_i \in [0,\,1], \quad i = 1,\,2,\,...,\,D
\end{aligned}
$$

DCTP2 to DCTP8.

$$
\begin{aligned}
&Min(f_1(x,\,t),\,f_2(x,\,t)) \qquad (2)\\
&f_1(x,\,t) = x_1\\
&f_2(x,\,t) = t * g(x,\,t) * (1 - \sqrt{\frac{f_1(x,\,t)}{g(x,\,t)}})\\
&s.t.,\ g(x,\,t) = 1 + \sum_{i=2}^{D}(x_i - sin(0.01\pi t))\\
&c_j(x) = ((f_2(x,\,t)/t - e_j) * cos(\theta_j))\\
&-(f_1(x,\,t) * sin(\theta_j)) \geq\\
&a_j * |sin[b_j * \pi * (((f_2(x,\,t)/t - e_j) * sin(\theta_j))\\
&+f_1(x,\,t) * cos(\theta_j))^{c_j}]|^{d_j}, \quad j = 1,\,2,...,\,C\\
&x_i \in [0,\,1], \quad i = 1,\,2,\,...,\,D
\end{aligned}
$$

where D, C and t denote respectively the number of decision variables, the number of constraints and the time step. The settings of the different parameters is presented in table 1. These proposed benchmark functions present different challenges to the optimization algorithm since the PF, the PS and also the constraints change simultaneously over time. These characteritics make the task of optimization much more difficult than dynamic unconstrained problems. In addition, as presented in [12], these test functions present two kinds of tunable difficulties in a multi-objective optimization EA: (1) difficulty in the vicinity of the optimal PF where constraints do not make a major portion of the search space infeasible except near the optimal PF (the case of DCTP1 to DCTP5), and (2) difficulty in the entire search space where constraints produce different disconnected regions of feasible objective space (the case of DCTP6 to DCTP8). It is worth noting that the source code, the optimal PFs and the optimal PSs of these problems are shared via https://sites.google.com/site/slimbechikh/sourcecodes.

4. DYNAMIC PENALTY FUNCTION-BASED NSGA-II

4.1 Basic idea

Since dynamic constrained MOPs can be seen as static constrained MOPs for a short period after a change occurs, we can borrow ideas from the static constrained MOPs

	D	C	a	b	c	d	e	θ
DCTP1	30	-	-	-	-	-	-	-
DCTP2	30	1	0.2	10	1	6	1	$-0.2 * \pi$
DCTP3	30	1	0.1	10	1	0.5	1	$-0.2 * \pi$
DCTP4	30	1	0.75	10	1	0.5	1	$-0.2 * \pi$
DCTP5	30	1	0.1	10	2	0.5	1	$-0.2 * \pi$
DCTP6	30	1	40	0.5	1	2	-2	$0.1 * \pi$
DCTP7	30	1	40	5	1	6	0	$-0.05 * \pi$
DCTP8	30	2	a_1	b_1	c_1	d_1	e_1	θ_1
			40	0.5	1	2	-2	$0.1 * \pi$
			a_2	b_2	c_2	d_2	e_2	θ_2
			40	2	1	6	0	$-0.05 * \pi$

Table 1: Parameter settings

to deal with constraints in DMOPs. Thus, we propose to take advantge of a constraint handling technique that was proposed to deal with constraints in a static environment and that has presented promising results and to adopt it to dynamic problems. The basic idea behind our proposal is to integrate into an existing and very well known dynamic multi-objective EA a constraint-handling technique presenting an ability to not only handle constraints but also to preserve and drive prominent infeasible individuals towards feasible ones. By doing so, we suppose that the constraint-handling technique should be able to find feasible individuals and to mantain some infeasible solutions allowing to avoid premature convergence; while the dynamic EA would be able to ensure the diversity in the population and to track changing PFs. To this purpose, we choose to combine the basic dynamic EA DNSGA-II with the adaptive penalty function proposed in [27]. Moreover, we replace the constraint-dominance principle used in DNSGA-II by the standard Pareto dominance. The resulting algorithm is called Dynamic Constrained NSGA-II and is denoted as DC-NSGA-II. In the following, we give a brief description of DNSGA-II and the adaptive penalty function. Then, we describe the proposed algorithm DC-NSGA-II.

4.2 Dynamic NSGA-II: DNSGA-II

In [13], Deb et al. extended the NSGA-II procedure in order to handle DMOPs and track changing optimal PF. Firstly, they introduce a test to identify whether there is a change in the problem through randomly picking a few solutions from the parent population and re-evaluating their objective functions values and their constraints. Then, if a change is detected in the problem, the parent population is re-evaluated before merging it with child population. Moreover, diversity is introduced in a first version (DNSGA-II-A) through the replacement of $\zeta\%$ of the new population with new randomly created solutions. In the second version (DNSGA-II-B), diversity is ensured by replacing $\zeta\%$ of the new population with mutated solutions. One of the merits of these procedures is that they can also solve constrained DMOPs using the constraint-dominance principle. However, the main drawback of this principle is that it may lose some potential information of the infeasible region which may lead to a premature convergence. This is why in this work we propose to replace this constraint-handling technique by the use of an adaptive penalty function.

4.3 Constraint handling mechanism: An adaptive penalty function

In [27], a constraint handling technique for multiobjective EAs is proposed based on an adaptive penalty function and a distance measure. In fact, Constrained MOPs are converted into MOPs by using a modified objective functions that include distance measure and adaptive penalty. Both of them are constituted by combining adaptively the normalized constraint violations and the normalized objective functions relatively to a parameter r_f, which is decided by the proportion of feasible solutions in current population. Then, the modified objective functions are used in the non-dominance sorting to facilitate the search of optimal solutions not only in the feasible space but also in the infeasible regions since this penalty function can make some infeasible solutions with good objective functions values and low constraint violation values be selected. The modified objective value $F_i(x)$ of the i^{th} objective has two components: distance measure $d_i(x)$ and adaptive penalty $p_i(x)$. The two components are calculated as follows:

$$F_i(x) = d_i(x) + p_i(x) \tag{3}$$

$$d_i(x) = \begin{cases} v(x), & if\ r_f = 0 \\ \sqrt{\tilde{f}_i(x)^2 + v(x)^2}, & otherwise \end{cases} \tag{4}$$

$$p_i(x) = (1 - r_f)X_i(x) + r_f Y_i(x) \tag{5}$$

$$X_i(x) = \begin{cases} 0, & if\ r_f = 0 \\ v(x), & otherwise \end{cases} \tag{6}$$

$$Y_i(x) = \begin{cases} 0, & if\ x\ is\ a\ feasible\ solution \\ \tilde{f}_i(x), & if\ x\ is\ an\ infeasible\ solution \end{cases} \tag{7}$$

where $\tilde{f}_i(x)$, $v(x)$ and r_f represent respectively the i^{th} normalized objective function value, the sum of the normalized constraints violation values and the feasibility ratio that calculates the ratio of the feasible individuals in current population. One of the merits of this adaptive penalty is its flexibility ensuring the efficiency of the use of infeasible individuals. As presented in [27] and with reference to equations 3, 4, 5, 6 and 7, we can state the following:

- If there are no feasible individuals in the population, the objective values will be totally disregarded allowing to find feasible individuals before looking for optimal solutions.

- If there are no infeasible individuals in the population, individuals will be compared based only on their objective functions values.

- If there are feasible individuals in the population, then individuals with both low objective functions values and low constraint violation values will be preferred

- If two individuals have close distance values, then the penalty term determines the dominant one. If r_f is small, the inividual closer to the feasible space will be dominant. If r_f is high, the individual with smaller objective functions values will be dominant. Otherwise, both individuals will be nondominant solutions.

4.4 Dynamic Constrained NSGA-II: DC-NSGA-II

The penalty function adopted in this work is self adaptive and it permits to guide the search either towards the feasibile area or the optimal region relatively to the current population. Such feature is highly recommended in dynamic environments. Thus, by incorporating it into a dynamic EA, we suppose that a diversity introduction mechanism would be sufficient to handle constraints changes affecting the feasible area and no further explicit actions would be needed. Thereby, we propose to integrate the adaptive penalty function into DNSGA-II, in its two versions (i.e., DNSGA-II-A and DNSGA-II-B). Moreover, we use the modified objective functions values in the nondominance ranking to replace the constraint dominance principle. This combination will permit to handle both changes of objective functions and constraints. Also, we propose to modify the diversity introduction mechanism. In fact, since accepting infeasible solutions may slow down convergence, we suggest to add a feasibility condition before incorporating any random or mutated solution into the population. This may prevents infeasible solutions to be integrated into the population. Thus, we propose to iterate over the procedure of generating a new solution randomly or by mutation until this solution is feasible or a fixed maximum number of iterations is reached. This will permit to control the new added solutions which should guide the rest of the population towards the new feasible regions. The flowchart of the proposed algorithm DC-NSGA-II is presented in figure 1. We denote by DC-NSGA-II-A and DC-NSGA-II-B respectively DC-NSGA-II where the diversity is introduced through the replacement of some individuals with randomly generated solutions and DC-NSGA-II where the diversity is introduced through the replacement of some individuals with mutated solutions.

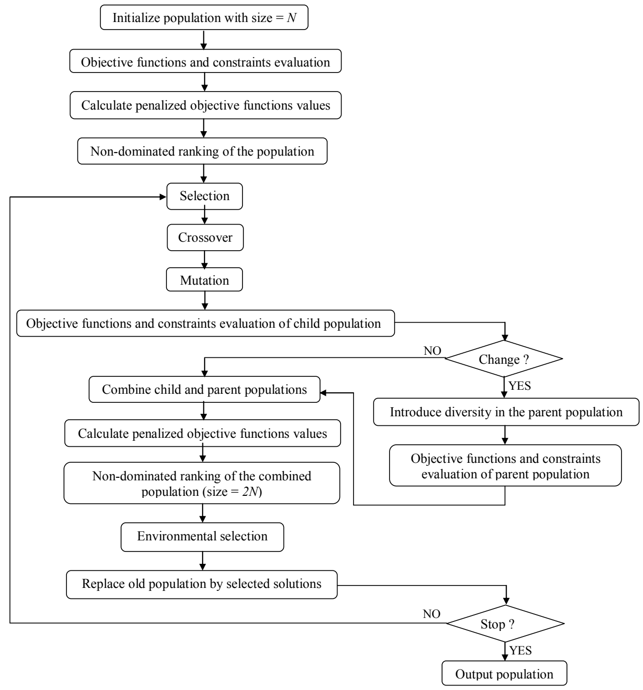

Figure 1: The flowchart of DC-NSGA-II.

5. EXPERIMENTAL STUDY

5.1 Performance metrics

5.1.1 Convergence indicators

We have opted to the use of the Inverted Generational Distance metric (IGD) [18] and the the HyperVolume (HV) Ratio metric ($HVRatio$) [6]. The IGD measures the gap between the optimal PF and the optimized one. The $HVRatio$ is the ratio between the HV of the obtained PF and the HV of the optimal one. The HV of a set A with respect to a RP ref denoted by $HV(A, ref)$ is the hyperarea of the set $R(A, ref)$. $HV(A, ref)$ measures how much of the objective space is dominated by A [6].

5.1.2 Diversity indicator

As a diversity indicator, we have used the maximum spread metric for dynamic environments (MS') introduced in [3].

In this work, for all performance indicators we use the mean indicator over time. Let θ be an indicator, the mean θ, denoted by $\overline{\theta}$, is calculated as follows

$$\overline{\theta} = \frac{1}{nbChanges} \sum_{i=1}^{nbChanges} \theta_i \tag{8}$$

where $nbChanges$ is the number of occured changes and θ_i is the value of θ computed before the occurence of the $(i+1)^{th}$ change.

5.2 Comparative Results

In this section, simulation results are presented using our proposed constrained DMOPs to assess the performance of the algorithms facing different difficulties. Moreover, we confronted our proposed algorithms, DC-NSGA-II-A and DC-NSGA-II-B to DNSGA-II-A and DNSGA-II-B respectively. For all the algorithms, we set the population size to 100 and all other parameters are set as in the original algorithm NSGA-II. For DC-NSGA-II versions, we set the maximum number of iterations before accepting an infeasible solution to be incorporated into the population at the step of diversity introduction to 5. The experiments are performed at a frequency of 10000 evaluations for each time step and with four test environments (t = 1, 2, 3, 4). For each experiment, thirty one independent simulation runs with randomly generated initial populations are conducted. For each test problem and each pair of algorithms, we perform a (two-sided) Wilcoxon test to decide whether or not the difference between the indicated median values (for each performance indicator) for the two algorithms is statistically significant on the considered problem instance. The results of the significance tests for the pairwise comparisons at level $\alpha = 0.05$ are presented in the form of (-: no significance) and (+: significance) respective appearence order of the algorithms.

Table 2 illustrates the obtained statistical results for the $\overline{IGD}$, $\overline{HVRatio}$ and $\overline{MS'}$ metrics. The PFs found by different algorithms after a single run for each test problem are drawn in figure 2. Relying upon table 2 and figure 2, we can infer the following conclusions. When observing the results of DCTP1, we can state that the optimal solutions found by DC-NSGA-II-A and DC-NSGA-II-B are widely spread and have respectively better convergence to the true PF than

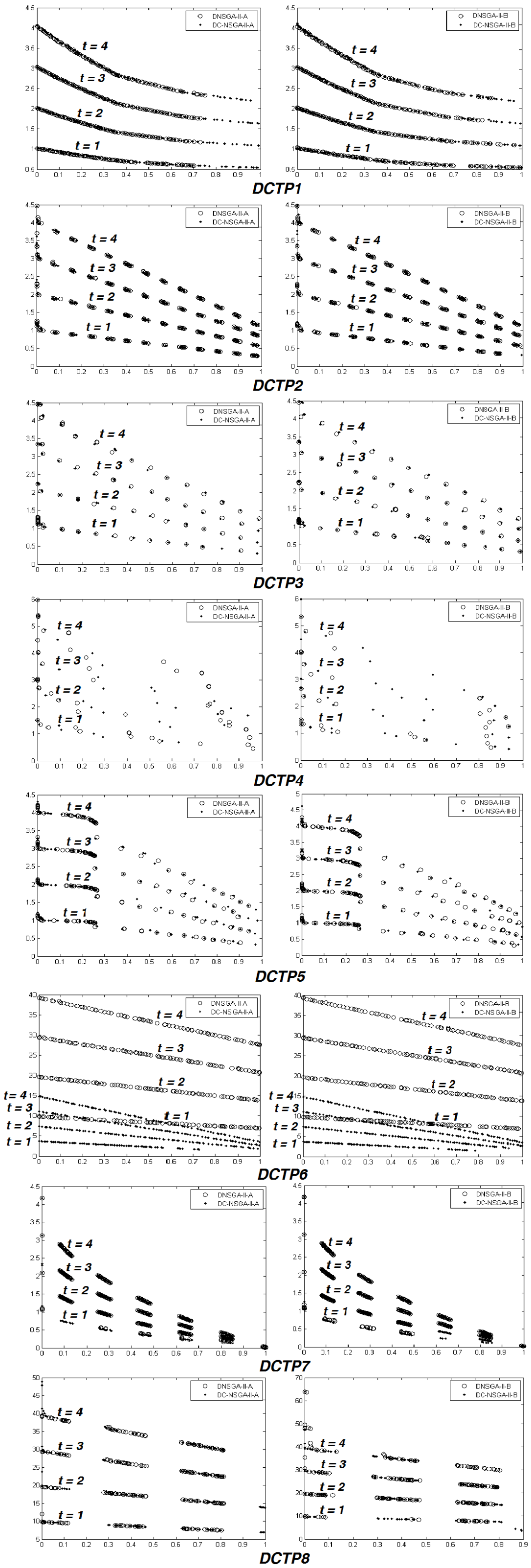

Figure 2: The obtained PFs using DNSGA-II-A, DC-NSGA-II-A (right), DNSGA-II-B and DC-NSGA-II-B (left).

DNSGA-II-A and DNSGA-II-B. The latter ones can only find a small part of the global PF and their obtained optimal solutions are very crowded. This indicates that they were not able to ensure a satisfactory diversity of the population. DC-NSGA-II-A has demonstrated a slightly better performance than DC-NSGA-II-B since random solutions ensure a higher level of diversity and do not give any attention to past optimal solutions. For DCTP2, since the constraint boundary is periodic, this test problem has a number of disconnected Pareto optimal regions at each time step. The experimental results demonstrate that using the penalty-based constrained handling method to keep infeasible solutions gives better results than only preferring the feasible ones. Moreover, the "*filtering*" mechanism used when applying the diversity introduction step promotes the performance of the proposed algorithms. DCPT3 is more difficult than DCTP2 since its disconnected feasible regions for each time step are constituted of only one Pareto optimal solution. With reference to the $\overline{IGD}$ metric, we may observe that our proposals have clearly better convergence than DNSGA-II versions to the global PF. The PF of DCTP4 is similar to that of DCPT3, but it is more complex because its Pareto optimal solutions are far from the constraint region. This is why we may observe the deterioration of the performance of the different algorithms. Nevertheless, the overall quality of the non-dominated solutions found by our proposals still is globally better than the ones obtained by any of the other algorithms. Concerning DCTP5, this test problem has a non-uniform PF which is constituted by a disjoint curve and some discrete points. The optimal solutions found by DC-NSGA-II-A are more approximate than all other algorithms with a satisfactory diversity of population. Moreover, as presented in figure 2, a discrete point near $f_1 = 0$ is still missed for the DNSGA-II versions. Although DCTP6 has a continuous optimal PF, its main difficulty is that its feasible objective space is presented in banded distribution. Thus, it is easy to be trapped in local optimal solutions. Figure 2 and table 2 clearly show that our proposed algorithms can converge to the true PF, while DNSGA-II versions are trapped in local PFs. This points out that their diversity of population should be improved further. When observing the obtained results on DCTP7 which has a disconnected optimal PF for each time step, we can see that the proposed algorithms get comparative performances, while DNSGA-II versions left a part of the disconnected optimal PFs. Finally, for DCTP8, where the feasible objective spaces are distributed in blocks, it was difficult to the different algorithms to avoid being trapped into local optimal PFs. This is because a very diversified population is needed to escape local optimal solutions which may not be the case when we deal with simultaneously time-dependent objectives and constraints. More advanced mechanisms should be used to solve such difficulty.

In summary, for the above test problems, our proposed algorithms behave globally better than the DNSGA-II versions with a statistical significance at level $\alpha = 0.05$. our proposals can not only track the time-varying PFs, but also find effectively the approximate Pareto optimal solutions with a satisfactory distribution. This is mainly due to the use of the adaptive penalty-based constraint handling technique which overcomes the disadvantage of infeasible individuals that always have worse rankings than feasible ones. Moreover, as the constraints change over time, feasible so-

	DC-NSGA-II-A			DC-NSGA-II-B			DNSGA-II-A			DNSGA-II-B		
	$\overline{IGD}$	$\overline{HVRatio}$	$\overline{MS'}$	$\overline{IGD}$	$\overline{HVRatio}$	$\overline{MS'}$	$\overline{IGD}$	$\overline{HVRatio}$	$\overline{MS'}$	$\overline{IGD}$	$\overline{HVRatio}$	$\overline{MS'}$
Md_1	**1,08E-02**	**9,86E+01**	**9,94E-01**	1,10E-02	9,85E+01	9,94E-01	2,13E-02	9,83E+01	9,33E-01	1,65E-02	9,84E+01	9,50E-01
Sd_1	**4,22E-04**	**1,79E-01**	**4,78E-03**	4,37E-04	1,35E-01	7,29E-03	1,38E-02	6,91E-01	4,39E-02	1,06E-02	4,72E-01	3,00E-02
Sg_1	**(+++)**	**(- - +)**	**(+++)**	(+++)	(- - -)	(+++)	(++ -)	(++ -)	(++ -)	(++ -)	(++ -)	(++ -)
Md_2	**5,00E-03**	**9,93E+01**	**9,99E-01**	5,10E-03	9,92E+01	9,99E-01	6,40E-03	9,89E+01	9,81E-01	6,20E-03	9,93E+01	9,99E-01
Sd_2	**2,19E-04**	**1,65E-01**	**3,45E-04**	5,45E-04	1,45E-01	8,37E-04	7,77E-03	4,46E-01	4,11E-02	3,62E-03	2,76E-01	2,31E-02
Sg_2	(- ++)	(- ++)	(- ++)	(- ++)	(- ++)	(- ++)	(+++)	(++ -)	(++ -)	(+++)	(++ -)	(++ -)
Md_3	**3,16E-02**	**9,57E+01**	9,84E-01	3,24E-02	9,56E+01	**9,88E-01**	5,19E-02	9,54E+01	9,32E-01	5,51E-02	9,55E+01	9,17E-01
Sd_3	**9,01E-03**	**2,98E-01**	1,73E-02	1,02E-02	3,46E-01	**2,62E-02**	4,47E-02	1,14E+00	5,74E-02	4,98E-02	1,89E+00	8,74E-02
Sg_3	**(- ++)**	**(- + -)**	(- ++)	(- ++)	(- - -)	**(- ++)**	(+++)	(+ - -)	(++ -)	(+++)	(- - -)	(++ -)
Md_4	1,73E-01	7,36E+01	7,91E-01	**1,58E-01**	**7,37E+01**	**8,44E-01**	3,01E-01	6,61E+01	6,43E-01	5,01E-01	6,05E+01	5,61E-01
Sd_4	1,48E-02	1,21E+00	4,79E-02	**2,83E-02**	**1,78E+00**	**5,86E-02**	1,71E-01	6,04E+00	1,13E-01	2,11E-01	8,42E+00	1,36E-01
Sg_4	(+++)	(- ++)	(- ++)	**(+++)**	**(- ++)**	**(- ++)**	(+++)	(++ -)	(++ -)	(+++)	(++ -)	(++ -)
Md_5	**8,90E-03**	**9,66E+01**	9,40E-01	9,30E-03	9,64E+01	**9,44E-01**	9,40E-03	9,62E+01	9,37E-01	9,10E-03	9,62E+01	9,18E-01
Sd_5	**5,89E-04**	**3,52E-01**	9,62E-03	8,45E-04	3,13E-01	**1,63E-02**	1,90E-03	2,15E+001	1,14E-01	2,23E-03	1,57E+00	7,03E-02
Sg_5	**(+++)**	(- ++)	(- ++)	(+ - -)	(- - -)	**(- - +)**	(+ - +)	(- - -)	(+ - -)	(+ - +)	(- - -)	(++ -)
Md_6	1,04E-01	9,50E+01	8,48E-01	**9,84E-02**	**9,53E+01**	**8,51E-01**	1,16E+01	0,00E+00	1,14E+00	1,16E+01	0,00E+00	1,14E+00
Sd_6	4,64E-02	1,87E+00	6,48E-02	**2,09E-01**	**8,77E+00**	**1,25E-01**	4,66E+00	3,88E+01	1,18E-01	3,44E-02	0,00E+00	5,03E-03
Sg_6	(- ++)	(- ++)	(+++)	**(- ++)**	**(- ++)**	**(+++)**	(++ -)	(++ -)	(++ -)	(++ -)	(++ -)	(++ -)
Md_7	**1,46E-02**	**9,85E+01**	**9,72E-01**	2,29E-02	9,84E+01	9,43E-01	2,68E-02	9,79E+01	9,13E-01	2,66E-02	9,78E+01	9,12E-01
Sd_7	**9,27E-03**	**5,85E-01**	**3,38E-02**	8,74E-03	7,49E-01	2,56E-02	2,45E-02	1,27E+004	4,67E-02	3,09E-02	1,58E+00	6,00E-02
Sg_7	**(+++)**	**(- ++)**	**(+++)**	(+++)	(- - -)	(+++)	(++ -)	(+ - -)	(++ -)	(++ -)	(+ - -)	(++ -)
Md_8	**1,19E+01**	1,94E+01	**2,73E-01**	1,23E+01	**1,95E+01**	2,71E-01	1,28E+01	1,87E+01	2,48E-01	1,23E+01	1,93E+01	2,59E-01
Sd_8	**1,50E+00**	6,53E+00	**1,01E-01**	3,71E+00	**1,87E+01**	1,81E-01	1,63E+00	9,97E+00	1,28E-01	3,08E+00	3,25E+00	4,68E-02
Sg_8	**(- - -)**	(- ++)	**(- ++)**	(- - -)	**(- - -)**	(- ++)	(- - -)	(- - -)	(++ -)	(- - -)	(- - -)	(++ -)

Table 2: Comparisons of statistical results of the non-dominated solutions found by the algorithms. Md_i, Sd_i, and Sg_i denote respectively the median value, the standard deviation value and the statistical significance for the $DCTP_i$ problem for $i = 1, 2, ..., 8$.

lutions at time t may not always remain feasible at time $t+1$ which makes discovering new feasible regions a difficult task. The use of a feasibility condition before incorporating any random or mutated solution guides the population towards the new feasible regions whenever a change takes place. When confronting our proposed versions, we can state that DC-NSGA-II-A has demonstrated a better performance than DC-NSGA-II-B on most test problems. This may be explained by the higher level of diversity ensured by the use of random solutions. This may help the algorithm to discover different search directions and to avoid giving any attention to past optimal solutions.

6. CONCLUSION

Motivated by the lack of attention on constrained DMOPs, we have proposed in this paper a set of test problems that extend a suite of static constrained MOPs where the PF, the PS and the constraints are simultaneously time-dependent. Moreover, we have proposed a new version of DNSGA-II to deal with dynamic constraints. We have replaced the constraint dominance principle by a more elaborated and self-adaptive penalty function. This allows searching for Pareto optimal solutions in both feasible and infeasible spaces. Besides, we have suggested to add a feasibility condition before incorporating any random or mutated solution into the population which may help the algorithm to find the prominent directions of the new feasible regions according to changing environments. The experimental results show that our algorithm is able to handle dynamic environments and to track changing PFs with time-varying constraints. Moreover, the obtained results have demonstrated the advantages of our proposals over the original DNSGA-II versions on both aspects of convergence and diversity on most test problems. This work may be extended in a number of ways. Firstly, as constraints changes can affect the feasibility area, some feasible solutions may become infeasible. Thus, elaborating a feasible guiding strategy that is able to guide these solutions towards the new feasible regions would be an interesting approach. Secondly, in addition to the challenging features characterizing constrained DMOPs, these problems may also suffer from multimodality [22]. More effective mechanisms may be proposed in this case to avoid being trapped into local optimal solutions. Moreover, other future directions for this work are: (1) integrating decision maker's preference information in dynamic optimization [21], and (2) making our algorithm into use to solve real world problems.

7. REFERENCES

[1] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Transactions on Cybernetics 44 (1) (2014) 40–53.

[2] B. Y. Qu, P. N. Suganthan, Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, Engineering Optimization 43 (4) (2011) 403–416.

[3] C. K. Goh, K. C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization, IEEE Transcations on Evolutionary computation 13 (1) (2009) 103–127.

[4] C. Li, S. Yang, A general framework of multipopulation methods with clustering in undetectable dynamic environments, IEEE Transaction on Evolutionary Computation 16 (4) (2012) 556–577.

[5] C. M. Fonseca, P. J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms: Parti: a unified formulation, IEEE Transactions on Systems and Cybernetics-Part A: Systems and Humans 1 (28) (1998) 26 – 37.

[6] D. A. van Veldhuizen, Multi-objective evolutionary algorithms: classification, analyses, and new innovations, Ph.D. thesis, Graduate School of engineering Air University (1999).

[7] E. Mezura-Montes, C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, Swarm and Evolutionary Computation 1 (4) (2011) 173 – 194.

[8] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach, in: Genetic and Evolutionary Computation Conference, 2006, pp. 1201–1208.

[9] J. Wei, L. Jia, A novel particle swarm optimization algorithm with local search for dynamic constrained multi-objective optimization problems, in: IEEE Congress on Evolutionary Computation (CEC), 2013, pp. 2436–2443.

[10] K. Deb, Single and multi-objective dynamic optimization: two tales from an evolutionary perspective, Tech. Rep. 2011004, Kanpur Genetic Algorithms Laboratory (2011).

[11] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[12] K. Deb, A. Pratap, T. Meyarivan, Constrained test problems for multi-objective evolutionary optimization, in: Evolutionary Multi-Criterion Optimization, Vol. 1993, 2001, pp. 284–298.

[13] K. Deb, U. Rao, S. Karthik, Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling, in: 4th International Conference on Evolutionary Multi-criterion Optimization, Vol. 4403, 2007, pp. 803–817.

[14] K. Harada, J. Sakuma, I. Ono, S. Kobayashi, Constraint-handling method for multi-objective function optimization: Pareto descent repair operator, in: 4th International Conference on Evolutionary Multi-criterion Optimization, 2007, pp. 156–170.

[15] L. J. Fogel, A. J. Owens, M. J. Walsh, Artificial Intelligence Through Simulated Evolution, 1966.

[16] L. Jiao, J. Luo, R. Shang, F. Liu, A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems, Applied Soft Computing 14 (2014) 363–380.

[17] M. Camara, J. Ortega, F. de Toro, Parallel processing for multi-objective optimization in dynamic environments, in: IEEE International Parallel and Distributed Processing Symposium, 2007, pp. 1–8.

[18] M. Sierra, C. C. Coello, Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance, in: 3rd International Conference On Evolutionary multi-criterion optimization, Vol. 3410, 2005, pp. 505–519.

[19] P. A. N. Bosman, Learning and anticipation in online dynamic optimization, in: Evolutionary Computation in Dynamic and Uncertain Environments, 2007, pp. 129–152.

[20] R. Azzouz, S. Bechikh, L. B. Said, A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes, in: IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 3168–3175.

[21] S. Bechikh, Incorporating decision maker's preference information in evolutionary multi-objective optimization, Ph.D. thesis, University of Tunis, Tunisia (2013).

[22] S. Wessing, M. Preuss, H. Trautmann, Stopping criteria for multimodal optimization, in: Parallel Problem Solving from Nature PPSN XIII, Vol. 8672, 2014, pp. 141–150.

[23] S. Yang, X. Yao, Population-based incremental learning with associative memory for dynamic environments, IEEE Transactions on Evolutionary Computation 12 (5) (2008) 542–561.

[24] S. Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, L. Kang, A dynamic multi-objective evolutionary algorithm based on an orthogonal design, in: IEEE Congress on Evolutionary Computation, 2006, pp. 573–580.

[25] T. Ray, H. Singh, A. Isaacs, W. Smith, Infeasibility driven evolutionary algorithm for constrained optimization, in: Constraint-Handling in Evolutionary Optimization, Vol. 198, 2009, pp. 145–165.

[26] T. Ray, K. Tai, K. C. Seow, Multiobjective design optimization by an evolutionary algorithm, Engineering Optimization 33 (4) (2001) 399–424.

[27] Y. G. Woldesenbet, G. G. Yen, B. Tessema, Constraint handling in multi-objective evolutionary optimization, IEEE Transactions on Evolutionary Computation 13 (3) (2009) 514–525.

[28] Y. Wang, B. Li, Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization, Memetic Computing 2 (1) (2010) 3–24.

[29] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1) (1996) 1–32.

[30] Z. Zhang, S. Qian, Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems, Soft Computing 15 (7) (2011) 1333–1349.