Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 781–786

Fast Solutions Enhancing using a Copula-based EDA and SVM for many-objective problems

Abdelhakim Cheriet [*] **Foudil Cherif** [**] **Abdelmalik Taleb-Ahmed** [***]

[*] *LESIA Laboraroty,University of Biskra, Algeria (e-mail: ahcheriet@gmail.com).*
[**] *LESIA Laboraroty,University of Biskra, Algeria (e-mail: Foud_cherif@yahoo.fr)*
[***] *LAMIH, Electrical Engineering Department, Valenciennes (e-mail: Abdelmalik.Taleb-Ahmed@univ-valenciennes.fr)*

Abstract In this paper we propose a new Copula-based Estimation of Distribution Algorithm, to solve Many-objective optimization problem and to get new optimal solutions in very court time. Our algorithm uses the proprieties of Copula and exploits their statistical properties to make new solutions using the founded optimal solutions through the estimation of their distribution. The first step of the proposed Copula-based Estimation of Distribution Algorithm (CEDA-SVM) is taking initial solutions offered by any MOEA (Multi Objective Evolutionary Algorithm), and then creates Copulas to estimate their distribution, and we use Support Vector Machine (SVM) to learn the Pareto solutions model; those Copulas will be used to generate new solutions and SVM to avoid the expensive function evaluations. The idea of using the estimated distribution of the optimal solutions helps CEDA-SVM to avoid running the optimizer (MOEA) every time we need new alternatives solutions when the founded ones are not satisfactory. We tested CEDA-SVM on a set of many-objective benchmark problems traditionally used by the community, namely DTLZ (1, 2, 3, and 4) with different dimensions (3, 5, 8, 10, and 15). We used CEDA along with MOEA/D-Schy and MOEA/D-BI as two examples of MOEA thus resulting in two variants CEDA-MOAE/D-Scy and CEDA-MOEA/D-BI and compare them with MOEA/D-Schy and MOEA/D-BI. The results of our experiments show that, with both variants of CEDA-SVM, new solutions can be obtained in a very small time compared to the other algorithms.

Keywords: Multiobjective optimization, many-objective optimization, estimation of distribution algorithm, SVM, Copula.

1. INTRODUCTION

In many fields of science and engineering a Decision Maker (DM) have to pick a number of best options from a very high number of choices, those choices are solutions of a given problem which have a number of criteria or objectives to satisfy. The solutions should satisfy all objectives in same time, this condition results that a solution from the choice space can't get better without detriment of another objective, and by considering this we arrive at a very important consequence; in this type of problem there is no single optimal solution but rather a set of optimal solutions (called Pareto solutions PS).

The evolutionary algorithms (EAs) are commonly used to resolve this class of problems Zhou et al. (2011). The complexity of resolving those problems repose in the fact that for every objective function of the problem have to be evaluated in every generation of the EA, thus if the number of objectives increases the search process get more and more complex. By definition, if the number of the objectives of the problem are equal or more than four the multiobjective problem is called Many-objective problem, numerous multiobjective evolutionary algorithms (MOEA) was used to deal with many-objective problems Von Lücken et al. (2014).

The techniques used for solving a multiobjective problem have evolved over time and those based on Evolutionary Algorithms (EA) are among the best suited ones such as MOEA/D, NSGA2, SPEA2, etc. Among the EA techniques, Estimation of Distribution Algorithms (EDA) is a novel class which has good searching ability and explicitly model the statistical distribution of the solutions.

Where a probabilistic model linking the interactions between the solutions is constructed and then used to generate new solutions. Specifically, we base our solution on Copulas for their ability to generate new solutions without the need for running the solving algorithms each time a newer solution is needed.

This is so because typical algorithms used to solve the MOPs are based on evolutionary algorithms where finding a solution involves starting with a population and evaluating its closeness to Pareto front each step and run again until a satisfactory solution is found. Each step solutions are sorted and involves costly objective function evaluations and sorting.

Copula functions describe the dependence structure of two or more random variables associated by a joint probability distribution function. In other words, they provide a scale-free description of how a number of random variables are distributed. Once the copula is unveiled, the whole joint probability distribution function can be found. Moreover, we can use the copula function to generate new samples with such a joint probabil-

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.869

ity distribution. Briefly, this is achieved by the way CEDA (Copula-base EDA) operates, which starts by selecting the best individual using a MOEA (Multi Objective Evolutionary Algorithm) Deb et al. (2002); Zitzler et al. (2002) from a population generated randomly. Then, CEDA uses the selected individuals to estimate their distribution using a Copula. The constructed Copula is used to generate a new population. CEDA continues with generating and selecting the best individuals until the stop condition is met. When CEDA stops, the latest generated individuals are considered to be Pareto optimal solutions and the last Copulas can be used in a later calls of CEDA to generate alternative optimal solutions and using one-class SVM to classify the generated solutions, if those generated do not satisfy the needs of the Decision Maker. This design saves CEDA the need of running an MOEA every time alternative solutions are requested by a Decision Maker when the found solutions are not satisfactory.

The main contributions of this paper are the following:

- We devise a Copula-based EDA to increase the size, deliver high diversity, and achieve a quick convergence of Pareto optimal solutions for a many-objective optimization problem. We achieve this by exploiting the statistical properties of Copulas to produce new solutions from the existing ones through the estimation of their distribution and using SVM.
- We thoroughly test CEDA on a set of benchmark problems traditionally used by the community, namely DTLZ (1, 2, 3, and 4) with different dimensions (3,5,8,10,15), using MOEA/D with Bi (Boundary intersection) and Tchebycheff as decomposition methods Zhang and Li (2007) algorithms as candidates examples for MOEA selecting methods. We find that new Pareto optimal solutions can be generated in a significantly smaller time compared to those found by MOEA/D , without compromising the quality (convergence and diversity) of these solutions.

The rest of the paper is organized as follows. In Section 2 a definition of many-objective optimization problems is given. In Section 3, we present an overview on EDA and describe how they generally operate. In Section 4, we provide a mathematical definition of Copula and some of their features used in EDA. We present our contribution SVM-CEDA Copula-based EDA in Section 5 and evaluate its performance on various benchmark problems in Section 6. We conclude our paper in Section 7.

2. MANY-OBJECTIVE OPTIMIZATION

Many-objective optimization problem is a multi-objective problem with number of objective more or equal to four objectives Li et al. (2015) so a multi-objective optimization problem is an optimization problem with multiple objective functions Deb et al. (2005). In mathematical terms, a multi-objective optimization problem can be formulated as follows.

$$\begin{array}{ll} \min \mathbf{F}(\mathbf{x}) & \text{where } \mathbf{F} = (f_0, f_1, \ldots, f_m) \\ \text{subject to } \mathbf{G}(\mathbf{x}) \leq 0 & \end{array} \quad (1)$$

With $\mathbf{x} \in \mathbb{R}^n, \mathbf{F}(\mathbf{x}) \in \mathbb{R}^m, \mathbf{G}(\mathbf{x}) \in \mathbb{R}^p$ here we have m functions to optimize and p constraints to satisfy. The main goal of optimization methods is to find an optimal solution for the problem described in (1). Note that a multiobjective problem has many objectives to achieve which are usually mutually contradictory. Therefore, a relation for assessing the goodness of a solution compared to another one should be defined. Typically, the Pareto Dominance relation (see Definitions below) is used to achieve this end.

Definition 1. Considering a minimization problem, a decision vector **u weakly dominates v** ($\mathbf{u} \preceq \mathbf{v}$) iff $f_i(\mathbf{u}) \leq f_i(\mathbf{v}), \forall i \in \{0,1,\ldots,m\}$ and $\exists j \in \{0,1,\ldots,m\}, f_j(\mathbf{u}) < f_j(\mathbf{v})$.

Definition 2. Considering a minimization problem,a decision vector **u dominates v** ($\mathbf{u} \prec \mathbf{v}$) iff $f_i(\mathbf{u}) < f_i(\mathbf{v}), \forall i \in \{0,1,\ldots,m\}$.

Definition 3. A solution x^* is a **Pareto optimal** solution if and only if there is no other admissible solution x where $f(x)$ dominates $f(x^*)$. So the solution of a Multiobjective problem is a set of solutions which are not dominated by any other solution, we call this set the **Pareto Solutions PS**. The image of this set in the objective space form the **Pareto front PF**.

3. ESTIMATION OF DISTRIBUTION ALGORITHMS

Recently, a class of evolutionary Algorithms called Estimation of Distribution Algorithms is used to optimize many problems in literature. EDA is a population-based algorithm that starts with a population initially a random one, and then tries to choose the best solutions using a fitness function (for example in our experimentation, the best solution is the one that is not dominated by any another solution). The statistical properties of the selected solutions (individuals) are used to find a distribution or a kind of function or law representing all the selected solutions. The EDA algorithms try in every generation of the algorithm to estimate the distribution of the best solution in this generation. After finding or estimating the distribution of the best selected solutions, a number of new individuals are generated using the created function or law. In general, those new individuals have the same proprieties of the best solutions of the precedent generation. The algorithm runs many generations according to the steps described above until a criterion stop is achieved.

The general steps followed by an EDA algorithm are described in Algorithm 1:

Algorithm 1 Estimation of Distribution Algorithm

```
Initialisation
while Not termination criterion do
    Select best Solutions
    Estimate the best Solutions Distribution
    Generate a candidate Solutions
end while
```

The principal step in an EDA is the estimation of a distribution. There are many ways of estimation used by different EDAs (see Hauschild and Pelikan (2011) for a good description of the used methods of estimation). Recently, Copula theory has been increasingly used for the estimation of distribution by EDAs to solve optimization problems Salinas-Gutiérrez et al. (2011), Wang and Zeng (2010) Salinas-Gutiérrez et al. (2009), including multiobjective problems Gao et al. (2013), Gao et al. (2014). Many types of Copula have been used in the literature. In Gao et al. (2013), the authors used T-Copula, in Wang et al. (2009), Salinas-Gutiérrez et al. (2010), and Gao (2009), the authors used an Archimedean copula, in Wang et al. (2010a) the authors used Clayton Copula, and in Wang et al. (2012), Chang and Wang (2011), and Wang et al. (2010b), the authors combined more than one Copula to find the best estimation.

In this paper we will focus on Archimedean Copulas for their ability to model dependence in high dimensions with only one parameter which has the good effect of speeding up multiobjective optimization computation time.

4. MATHEMATICAL OVERVIEW ON ARCHIMEDEAN COPULAS

As defined in Nelsen (2006), Copulas are functions that join or couple multivariate distribution functions to their one-dimensional marginal distribution functions and as distribution functions whose one-dimensional margins are uniform.

Definition 4. A function C is called a *Copula* if only if is defined:

$$C : [0,1]^d \rightarrow [0,1]$$

It has the following characteristics:

$C(u_1,\cdots,u_d) = 0$ If one of its components u_i is equal to zero.
$C(1,\cdots,1,u_i,1,\cdots,1) = u_i$

In addition, C must be $d-increasing$. Example, for $d = 2$, we have:

$$C(u,v) : [0,1]^2 \rightarrow [0,1]$$

For any $0 \leq u \leq 1$ and $0 \leq v \leq 1$ we have the three following conditions:

$$C(0,v) = C(u,0) = 0$$
$$C(1,v) = v$$
$$C(u,1) = u$$

For any u and v, we define the $2-increasing$ propriety as:

$$C(u_1,v_1) - C(u_1,v_2) - C(u_2,v_1) + C(u_2,v_2) \geq 0$$

Definition 5. According to $Sklar's$ theorem, if C is a copula, and if $F_1,\cdots,F_d$ are a cumulative distribution functions (univariate), then:

$$F(x_1,\cdots,x_d) = C(F_1(x_1),\cdots,F_d(x_d))$$

is a cumulative distribution function with a dimension d, where the marginals are $F_1,\cdots,F_d$ exactly.

The converse is also true: if F is cumulative distribution function with d dimension, there is a C copula such that:

$$F(x_1,\cdots,x_d) = C(F_1(x_1),\cdots,F_d(x_d))$$

where all F_i are F marginals' laws.

5. CEDA-SVM: COPULA-BASED EDA WITH SVM ONE-CLASS CLASSIFIER

The goal of this proposal is two-fold:(1) to help the DM (decision maker) to find the solutions that are near to its interest, and (2) to reduce the number of function evaluations which decrease significantly the execution time. For this purpose, we use a two-phase algorithm composed of the *Optimization* phase, as it is proposed in Cheriet and Cherif (2015), that finds a number of the best solutions to a defined problem (see Section 5.1) and the *Update* phase which finds another set of the best solutions until the DM is satisfied using a SVM classifier (see Section 5.2).

5.1 Optimization Phase

In this step we use our proposed algorithm (Algorithm 2) with two steps (i) the *Selection* and (ii) the *Reproduction*. In the Selection step (performed by Function `SelectUsingMOEA`), we use the *MOEA/D* to select the best solutions that will be used in the Reproduction step where we make use of Copulas to estimate and regenerate new individuals.

Algorithm 2 Copula-based EDA

```
function CEDA
    $\mathbf{P_0} \leftarrow$ Initialization($m$)
    $\mathbf{P} \leftarrow$ SelectUsingMOEA($\mathbf{P_0}$)
    while Not termination criteria do            ▷ A maximum number of generations is used
        C $\leftarrow$ ConstructCopulas($\mathbf{P}$)
        $\mathbf{P'} \leftarrow$ GenerateSolutions(C)
        $\mathbf{P''} \leftarrow$ SelectUsingMOEA($[\mathbf{P'}\mathbf{P}]^T$)    ▷ $[\mathbf{P'}\mathbf{P}]^T$ takes all individuals of $\mathbf{P'}$ and $\mathbf{P}$
        $\mathbf{P} \leftarrow \mathbf{P''}$
    end while
    return $(\mathbf{P}, C)$
end function
```

Initially we put population $\mathbf{P_0} = [\mathbf{x_1},\ldots,\mathbf{x_m}]^T$ where $\mathbf{x_i}, i \in [1,m]$ are the individuals. Every individual $\mathbf{x_i} = [x_{i1},\ldots,x_{in}]$ where $x_{min} \leq x_{ij} \leq x_{max}$. Then the selection step use the algorithms MOEA/D - with Tchebycheff or Boundary Intersection (BI) as decompositon method - as a MOEA. This step return a set of individuals, those individuals will be used in the next step of the algorithm which is the reproduction step. We call $\mathbf{P}$ the matrix of the individuals resulting from the selection process operated on the precedent population. For the first generation, we use the initial population $\mathbf{P_0}$. We have $\mathbf{P}$ defined as the following:

$$\mathbf{P} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad (2)$$

For reproduction step, we create an Archimedean Copulas, represented by the vector $C = [C_1 \ldots C_j \ldots C_n]$, using the sub vectors $\mathbf{u_1},\ldots,\mathbf{u_n}$ and $\mathbf{v_1},\ldots,\mathbf{v_n}$, where each Copula $C_j, j \in [1,n]$ is constructed from the sub vectors $\mathbf{u_j}$ and $\mathbf{v_j}$ as shown in Algorithm 3.

Algorithm 3 Construct Copulas

```
function CONSTRUCTCOPULAS($\mathbf{P}$, $type$)        ▷ $type$ is the Copula: Frank, Clayton, Gumbel
    for all $\mathbf{w_j}$ a vector in $\mathbf{P^T}$ do
        $\mathbf{u_j} \leftarrow$ RandomPick($\mathbf{w_j}$)
        $\mathbf{v_j} \leftarrow$ Remainder($\mathbf{w_j}, \mathbf{u_j}$)
        $\mathbf{C_j} \leftarrow$ Copula($\mathbf{u_j}, \mathbf{v_j}, type$)
    end for
    return $C = [C_1 \ldots C_j \ldots C_n]$
end function
```

We use the constructed Copulas $C_1,\ldots,C_n$ to generate new individuals. The set of the new generated individuals $\mathbf{X'} = [\mathbf{w'_1},\ldots,\mathbf{w'_n}]$ where $\mathbf{w'_j}$ is the concatenation of $\mathbf{u'_j}$ and $\mathbf{v'_j}$ which are sampled using Copula C_j. Note that the vector $\mathbf{w'_j}$ (resulting from the concatenation of $\mathbf{u'_j}$ and $\mathbf{v'_j}$) is of size m' that is not necessarily the same of the initial population size m. The new individuals are therefore the vectors $\mathbf{x'_i}, i \in [1, m']$ where $\mathbf{X}' = [\mathbf{x'_1}, \dots, \mathbf{x'_{m'}}]^T$. Algorithms 4 summarizes theses steps.

Algorithm 4 Generate Solutions

```
function GenerateSolutions($(C, m')$)
    for all $C_j$ in $C$ do
        $(\mathbf{u'_j}, \mathbf{v'_j}) \leftarrow$ GenerateFromCopula$(C_j, m')$
        $\mathbf{w'_j} \leftarrow$ Concat$(\mathbf{u'_j}, \mathbf{v'_j})$
    end for
    return $\mathbf{X}' = [\mathbf{w'_1} \dots \mathbf{w'_j} \dots \mathbf{w'_n}]$
end function
```

5.2 Update Stage using one-class SVM

In general, the purpose of classification algorithms is to classify an unknown object from several predefined classes. In the problem of mono-class classification, it is assumed that only the data of the target class is available for the learning of the classifier, while the test set comprises positive examples (same class) and negative (classes).

Learning data used here is the Pareto Solution (PS) and all individuals in the PS are placed in one class (A), and all other individuals does not belong to this class. In the classification process we input the generated solutions using the copulas as an entry to the classifier (one-class SVM) and we assume that the individuals of the class (A), will be a Pareto Solutions set.

Algorithm 5 SVM classifier

```
function Classify($(NDset, P')$)
    Train-mono-class-SVM($(NDset)$)
    $P''$=Classify($(P')$)
    return $P''$
end function
```

As described in previous sections our proposal aims to give more alternative solutions to the DM in a very small time of execution, so in the optimization stage we calculate new solutions as shown in Algorithm 2. These solutions may not suit the needs of the DM this will need that another phase of new solutions generation is needed. The proposed Update phase in this paper (as shown in Algorithm 6) makes it possible for the decision maker to find other new solutions quickly by using the Copulas constructed in the Optimization stage and a mono-class SVM . Firstly, we call the Function `GenerateSolutions` with C (the Copulas constructed in the Optimization phase), and m'' the number of new solutions required, then we use a mono-class SVM to classify the generated individuals as shown in Algorithm 5 The output of the Update stage is the population $\mathbf{P}_{\text{update}}$.

Algorithm 6 Update Solutions

```
function UpdateSolutions($(C, m'')$)
    $\mathbf{P}_{\text{tmp}} \leftarrow$ GenerateSolutions$(C, m'')$
    $\mathbf{P}_{\text{update}} \leftarrow$ Classify(ParetoSolutions,$\mathbf{P}_{\text{tmp}}$)
    return $\mathbf{P}_{\text{update}}$
end function
```

6. EXPERIMENTATION

In this section, the proposed method will be compared with the well known algorithms: multi-objective evolutionary algorithm based on decomposition (MOEA/D) with two approaches of decomposition (Boundary Intersection (MOEA/D-BI) and Tchebycheff Approach (MOEA/D-Tchy)), through experiments. The experiments are conducted on four widely used and challenging enough many-objective benchmark problems.

6.1 Used Benchmark problems and metrics

To proof the efficiency of the proposed algorithm we have choice to test it using a set of test benchmark problems, and for the metric we calculate the IGD metric for all used benchmark, we used the DTLZ benchmark with different dimension that great or equal to 3, so the following test problems are used:

Each algorithm is run 20 times independently for each test instance, the parameter configuration in this paper is as follows:

Benchmark	Dimension	Generations
DTLZ(1,2,3,4)	3	250
DTLZ(1,2,3,4)	5	350
DTLZ(1,2,3,4)	8	500
DTLZ(1,2,3,4)	10	750
DTLZ(1,2,3,4)	15	1000

Dimension	Number of individuals
3	120
5	126
8	120
10	220
15	120

6.2 Simulation Results

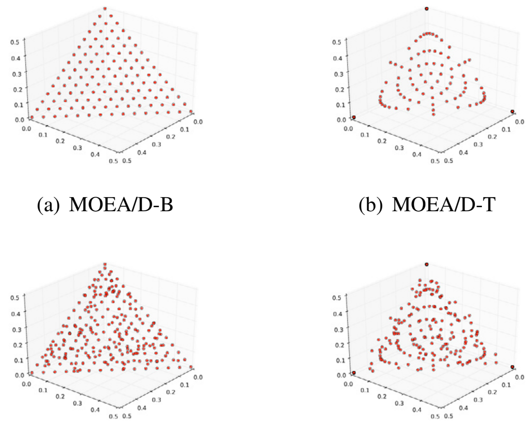
In this section we will show the results of the different simulations of the algorithm, in every step of simulation we execute the CEDA algorithm using the MOEA/D with Tchebycheff and Boundary Intersection on the different benchmarks. The presented Tables and figures in next sections will clarify the simulation results of the proposed algorithm.

The Figure 1 shows the result of the DTLZ1 benchmark with 3 dimensions using MOEA/D-BI and MOEA/D-T (Tchebycheff), and it shows that the update phase give more alternative solutions.

The Figure 2 shows the result of the DTLZ3 benchmark with 3 dimensions using MOEA/D-BI and MOEA/D-T (Tchebycheff), and it shows that the update phase give more alternative solutions.

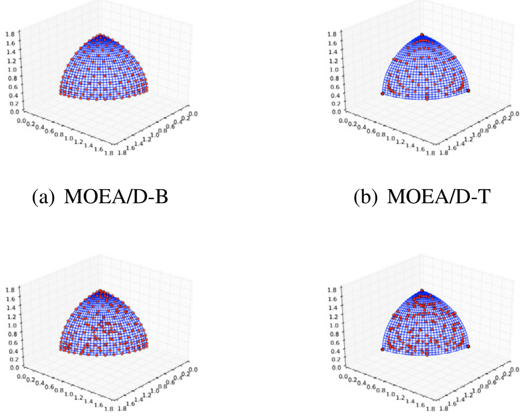
The Figures 3(a) and 3(c) shows the result of the DTLZ3 benchmark with 5 dimensions using MOEA/D-BI , and it shows that the update phase give more alternative solutions. Also, Figures 3(b) and 3(d) the result of the DTLZ3 benchmark with 8 dimensions which we use more objectives that the previous benchmark , and it shows that the use of CEDA update phase give more alternative solutions.

In Table 1 we present the values of the metric IGD used in our experimentation. IGD described in Tan et al. (2002) is used


(a) MOEA/D-B (b) MOEA/D-T

(c) CEDA-B (d) CEDA-T

Figure 1. Pareto front of the DTLZ1 problem with three objectives using the MOEA/D-B, MOEA/D-T, before and after updating the Pareto front using the Copula-based estimation of distribution algorithm.


(a) MOEA/D-B (b) MOEA/D-T

(c) CEDA-B (d) CEDA-T

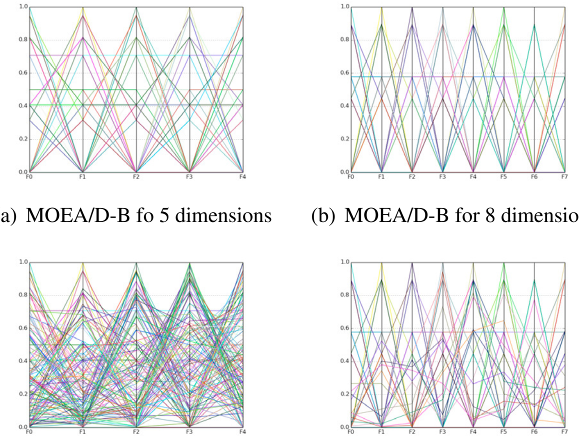
Figure 2. Pareto front of the DTLZ3 problem with three objectives using the MOEA/D-B, MOEA/D-T, before and after updating the Pareto front using the Copula-based estimation of distribution algorithm.

to measure the quality of the obtained PF in multi-objective optimization problems. The different IGD values of the used benchmarks show that the approximation of the obtained PF using the proposed SVM-CEDA are good or equal to the ones achieved with the classical MOEA/D for the two decomposition methods. This proof that the obtained new solutions using CEDA-SVM are also Pareto solutions. We can also see in all figures that the obtained new solutions are Pareto solutions.

Table 2 shows clearly that our proposition gets new solutions compared with the classical solutions. Also, as mentioned in the algorithm 5 our proposition selects new best solutions just by using the one-class SVM without need that the newly generated solutions to be evaluated.

7. CONCLUSION

In this paper, we have presented three main contributions: 1) the proposal of a new Estimation of distribution Algorithm.



(a) MOEA/D-B fo 5 dimensions (b) MOEA/D-B for 8 dimensions

(c) CEDA-B for 5 dimensions (d) CEDA-B 8 dimensions

Figure 3. Pareto front of the DTLZ3 problem with 5 and 8 objectives using the MOEA/D-B, before and after updating the Pareto front using the Copula-based estimation of distribution algorithm.

Table 1. The mean of IGD metric for the used benchmarks with the different dimensions

Benchmark	MOEA/D-B	CEDA-B	MOEA/D-T	CEDA-T
DTLZ1, 3D	$1.66e-02$	$1.38e-02$	$3.10e-02$	$2.04e-02$
DTLZ1, 5D	$1.82e+01$	$1.95e+01$	$2.13e+01$	$2.75e+01$
DTLZ1, 8D	$1.44e-01$	$1.22e-01$	$1.16e-01$	$9.45e-02$
DTLZ1, 10D	$3.77e-02$	$3.68e-02$	$2.35e-01$	$2.33e-01$
DTLZ1, 15D	$6.46e-02$	$6.32e-02$	$3.12e-01$	$3.13e-01$
DTLZ2, 3D	$4.33e-02$	$2.41e-02$	$7.45e-02$	$3.12e-02$
DTLZ2, 5D	$3.63e-01$	$2.82e-01$	$2.14e-01$	$1.51e-01$
DTLZ2, 8D	$4.43e-01$	$3.64e-01$	$4.70e-01$	$4.56e-01$
DTLZ2, 10D	$8.60e-02$	$8.34e-02$	$7.56e-01$	$7.33e-01$
DTLZ2, 15D	$8.67e-02$	$8.84e-02$	$9.43e-01$	$9.17e-01$
DTLZ3, 3D	$4.33e-02$	$3.90e-02$	$7.45e-02$	$5.99e-02$
DTLZ3, 5D	$1.51e+02$	$1.59e+02$	$1.20e+02$	$1.33e+02$
DTLZ3, 8D	$4.46e-01$	$4.25e-01$	$4.59e-01$	$4.46e-01$
DTLZ3, 10D	$1.03e-01$	$9.94e-02$	$8.10e-01$	$8.07e-01$
DTLZ3, 15D	$1.60e-01$	$1.59e-01$	$1.01e+00$	$1.01e+00$
DTLZ4, 3D	$5.12e-02$	$5.08e-02$	$7.45e-02$	$7.12e-02$
DTLZ4, 5D	$7.84e-01$	$7.60e-01$	$2.94e-01$	$2.15e-01$
DTLZ4, 8D	$5.66e-01$	$5.71e-01$	$5.18e-01$	$5.12e-01$
DTLZ4, 10D	$4.58e-01$	$4.53e-01$	$7.75e-01$	$7.76e-01$
DTLZ4, 15D	$4.76e-01$	$4.69e-01$	$9.18e-01$	$9.12e-01$

Table 2. The mean of the New obtained solution for the used benchmarks with the different dimensions.

Benchmark	MOEA/D-B	CEDA-B	MOEA/D-T	CEDA-T
DTLZ1, 3D	96.00	566.08	96.00	559.32
DTLZ1, 5D	96.00	4.16	96.00	11.80
DTLZ1, 8D	95.56	42.12	96.00	149.88
DTLZ1, 10D	173.36	5.12	175.80	4.08
DTLZ1, 15D	92.84	0.12	95.04	1.84
DTLZ2, 3D	96.00	526.04	96.00	555.52
DTLZ2, 5D	96.00	152.56	96.00	638.40
DTLZ2, 8D	96.00	50.72	96.00	68.80
DTLZ2, 10D	176.00	17.80	176.00	38.88
DTLZ2, 15D	96.00	1.68	96.00	37.76
DTLZ3, 3D	96.00	554.68	96.00	673.64
DTLZ3, 5D	96.00	2.08	96.00	6.32
DTLZ3, 8D	95.20	24.68	95.56	166.08
DTLZ3, 10D	172.08	6.00	174.04	27.08
DTLZ3, 15D	88.16	0.00	89.60	0.68
DTLZ4, 3D	96.00	19.08	96.00	34.48
DTLZ4, 5D	96.00	22.60	96.00	114.00
DTLZ4, 8D	96.00	74.08	96.00	28.76
DTLZ4, 10D	176.00	35.00	176.00	20.12
DTLZ4, 15D	96.00	28.72	96.00	7.68

The proposed algorithm uses a very useful estimation method in statistic which is the Copula and a very famous type which is the Archimedean one and 2) the application of the new Copula-based EDA algorithm to solve Many-objective Optimization Problems then 3) the use of the obtained Pareto Solutions Estimated Model to generate a new Pareto Solutions in a very efficiency manner using SVM. The new generated solution may enrich the Decision's choice space of a Decision Maker.

The proposed Copula-based EDA Algorithm uses in the phase of estimation the Archimedean Copula. This Copula is the Model used to describe the distribution and dependencies between the Pareto Solutions obtained in a generation of the algorithm. The Copula is created by considering that the Pareto Set have a kind of dependency. The Pareto Set is bisected randomly, and the two parts are used as an input for the Copula using the proposed algorithm. The new offsprings will be generated using the Copula Model, these offsprings are evaluated and the best are selected using the MOEA/D (Tchy or BI) algorithms. The proposed algorithm shown good results with the used four Many-objective benchmark problems. A new aspect was handled by the proposed algorithm, this aspect was the updating PS in a very negligible execution time, this aspect can help the DM maker to get new choices, all new choices are optimal choices because they are generated from the last Copula Model, which is the Model that describe the Pareto Optimal Solutions. This aspect is a new and original because all posteriori methods can't give the DM new PS without running their algorithm again.

The Copula Model can be viewed as a memory that conserve the characteristics of the PS, this vision has given us the motivation to use this algorithm in the Dynamic Multiobjective Algorithm, this work can be a perspective to new works by using the Copula Model as a memory, the memory based algorithm are a class of methods used in Multiobjective optimisation problems and that gives a good results.

REFERENCES

Chang, C. and Wang, L. (2011). A multi-population parallel estimation of distribution algorithms based on clayton and gumbel copulas. In *Artificial Intelligence and Computational Intelligence*, 634–643. Springer.

Cheriet, A. and Cherif, F. (2015). A Posteriori Pareto Front Diversification Using a Copula-Based Estimation of Distribution Algorithm. *International Journal of Advanced Computer Science and Applications*, 6(12).

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2), 182–197. doi:10.1109/4235.996017.

Deb, K., Thiele, L., and Laumanns, M. (2005). Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective*, (1990), 1–27. doi:10.1007/1-84628-137-7_6.

Gao, Y. (2009). Multivariate estimation of distribution algorithm with laplace transform archimedean copula. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, 1–5. IEEE.

Gao, Y., Peng, L., Li, F., Liu, M., and Hu, X. (2013). Eda-based multi-objective optimization using preference order ranking and multivariate gaussian copula. In C. Guo, Z.G. Hou, and Z. Zeng (eds.), *Advances in Neural Networks ISNN 2013*, volume 7952 of *Lecture Notes in Computer Science*, 341–350. Springer Berlin Heidelberg.

Gao, Y., Peng, L., Li, F., Liu, M., and Hu, X. (2014). Multi-objective estimation of distribution algorithms using multivariate archimedean copulas and average ranking. In *Foundations of Intelligent Systems*, 591–601. Springer.

Hauschild, M. and Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3), 111–128.

Li, B., Li, J., Tang, K., and Yao, X. (2015). Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1), 13.

Nelsen, R.B. (2006). *An introduction to copulas*. Springer.

Salinas-Gutiérrez, R., Hernández-Aguirre, A., and Villa-Diharce, E.R. (2009). Using copulas in estimation of distribution algorithms. In *MICAI 2009: Advances in Artificial Intelligence*, 658–668. Springer.

Salinas-Gutiérrez, R., Hernández-Aguirre, A., and Villa-Diharce, E.R. (2010). D-vine EDA: a new estimation of distribution algorithm based on regular vines. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 359–366. ACM.

Salinas-Gutiérrez, R., Hernández-Aguirre, A., and Villa-Diharce, E.R. (2011). Estimation of distribution algorithms based on copula functions. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '11, 795–798. ACM, New York, NY, USA.

Tan, K., Lee, T., and Khor, E. (2002). Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. *Artificial intelligence review*, 979–986.

Von Lücken, C., Barán, B., and Brizuela, C. (2014). A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3), 707–756. doi:10.1007/s10589-014-9644-1.

Wang, L., Wang, Y., Zeng, J., and Hong, Y. (2010a). An estimation of distribution algorithm based on clayton copula and empirical margins. In *Life System Modeling and Intelligent Computing*, 82–88. Springer.

Wang, L.F. and Zeng, J.C. (2010). Estimation of distribution algorithm based on copula theory. In *Exploitation of linkage learning in evolutionary algorithms*, 139–162. Springer.

Wang, L.F., Zeng, J.C., and Hong, Y. (2009). Estimation of distribution algorithm based on archimedean copulas. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 993–996. ACM.

Wang, L., Guo, X., Zeng, J., and Hong, Y. (2010b). Using gumbel copula and empirical marginal distribution in estimation of distribution algorithm. In *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on*, 583–587. IEEE.

Wang, X., Gao, H., and Zeng, J. (2012). Estimation of distribution algorithms based on two copula selection methods. *Int. J. Comput. Sci. Math.*, 3(4), 317–331.

Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. doi:10.1109/TEVC.2007.892759.

Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32–49.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the strength pareto evolutionary algorithm for multi-objective optimization. In *Evolutionary Methods for Design, Optimisation, and Control*, 95–100. CIMNE, Barcelona, Spain.