A Hybrid Evolutionary Approach with Search Strategy Adaptation for Mutiobjective Optimization

Ahmed Kafafy Stéphane Bonnevay Ahmed Bounekkar
Laboratoire ERIC - Université de Lyon
5 avenue Pierre Mendès-France
69676 Bron Cedex, France
ahmedkafafy80@gmail.com stephane.bonnevay@univ-lyon1.fr bounekkar@univ-lyon1.fr

ABSTRACT

Hybrid evolutionary algorithms have been successfully applied to solve numerous multiobjective optimization problems (MOP). In this paper, a new hybrid evolutionary approach based on search strategy adaptation (HESSA) is presented. In HESSA, the search process is carried out through adopting a pool of different search strategies, each of which has a specified success ratio. A new offspring is generated using a randomly selected strategy. Then, according to the success of the generated offspring to update the population or the archive, the success ratio of the selected strategy is adapted. This provides the ability for HESSA to adopt the appropriate search strategy according to the problem on hand. Furthermore, the cooperation among different strategies leads to improve the exploration and the exploitation of the search space. The proposed pool is combined to a suitable evolutionary framework for supporting the integration and cooperation. Moreover, the efficient solutions explored over the search are collected in an external repository to be used as global guides. The proposed HESSA is verified against some of the state of the art MOEAs using a set of test problems commonly used in the literature. The experimental results indicate that HESSA is highly competitive and can be considered as a viable alternative.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, simulated annealing.*
; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*heuristic methods.*

General Terms

Algorithms, Experimentation, Performance, Verification

Keywords

Multiobjective Optimization, Hybrid Evolutionary Algorithm, Search Strategy adaptation.

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.

1. INTRODUCTION

Many of the real-world problems can be modeled as multiple objective optimization problems (MOP), which are often characterized by their large size and the presence of multiple, conflicting objectives. In general, the basic task in multiple objective optimization is the identification of the set of Pareto optimal solutions or even a good approximation set to the Pareto Front (PF). Many of metaheuristics have been introduced in the last thirty years [4] such as Evolutionary Algorithms (EA), Evolutionary strategies (ES), Simulated Annealing (SA),Tabu Search (TS), Scatter Search (SS), Particle Swarm Optimization (PSO), Differential Evolution (DE). More details are found in [2].

Multiobjective Evolutionary Algorithms (MOEAs) are a very active and promising research area. They have recently received increased interest because they offer practical advantages in facing difficult optimization problems. Solving MOPs and their applications using evolutionary algorithms have been investigated by many authors [7, 10, 13, 27, 28]. NSGAII [7] and SEPA2 [27] are the most popular Pareto dominance based MOEAs that have been dominantly used. Based on many traditional mathematical programming methods for approximating the PF [22], the approximation of the PF can be decomposed into a number of single objective subproblems. Some of MOEAs adopt this idea such as MOGLS [12], MOEA/D [25]. Many of search algorithms attempt to obtain the best from a set of different metaheuristics that perform together, complement each other and augment their exploration capabilities. They are commonly called Hybrid Metaheuristics (HM). Diversification and intensification [2] are the two major issues when designing a global search method. Diversification refers to the ability to visit many different regions in the search space, whereas intensification refers to the ability to obtain high quality solutions within those regions. A search algorithm must balance between sometimes-conflicting two goals. The design of HM gives the ability to control this balance [20].

Motivated by the results achieved in [14, 15], this paper tries to extend this work to the continuous search domains. It studies the cooperation of different search operators and analyze its effect on handling MOPs. It develops a hybrid evolutionary approach (HESSA) which incorporates a pool of adaptive search strategies within the MOEA/D framework. The main goals are to capture the benefits of those strategies with providing cooperation and integration. Also, to make the approach capable of selecting the suitable search strategy according to the problem on hand. The remainder of this paper is organized as follows: section 2 presents some of the basic concepts and definitions. In section 3, some of the different search operators are overviewed. The proposed HESSA is presented in section 4. In additions, the experimental design and results are involved in sections 5 and 6 respectively. Finally, section 7 presents the conclusions and some further directions.

2. BASIC CONCEPTS AND DEFINITIONS

Without loss of generality, the MOP can be written as:

$$\begin{array}{c} Minimize\, F(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \\ Subject\, to:\; x \in \Omega \end{array} \quad (1)$$

where $F(x)$ is the m-dimensional objective vector, $f_i(x)$ is the i^{th} objective to be minimized, $x = (x_1, \cdots, x_n)^T$ is the n-dimensional decision vector and Ω is the feasible decision space.

Definition 1. A solution x dominates y (noted as: $x \preccurlyeq y$) if: $f_i(x) \leq f_i(y)$, $\forall i \in \{1, \cdots, m\}$ and $f_i(x) < f_i(y)$ for at least one i.

Definition 2. A solution x is said to ϵ-dominate a solution y for some $\epsilon > 0$ (noted as: $x \preccurlyeq_\epsilon y$) if and only if: $f_i(x) \leq (1+\epsilon) f_i(y)$, $\forall i \in \{1, \cdots, m\}$.

Definition 3. A solution x is called efficient ($Pareto\text{-}optimal$) if: $\nexists y \in \Omega$ such that $y \preccurlyeq x$.

Definition 4. The Pareto optimal set (P^*) is the set of all efficient solutions:

$$P^* = \{x \in \Omega \,|\, \nexists y \in \Omega\,,\, y \preccurlyeq x\}$$

Definition 5. The Pareto front (PF) is the image of P^* in the objective space:

$$PF = \{F(x) = (f_1(x), \cdots, f_m(x)) : x \in P^*\}$$

Definition 6. Given a reference point r^* and a weight vector $\Lambda = [\lambda_1, \cdots, \lambda_m]$ such that $\sum_{i=1}^m \lambda_i = 1$, $\lambda_i \geq 0, \forall i$, the *weighted sum* ($F^{ws}$) and the *weighted Tchebycheff* (F^{Tc}) scalarizing functions corresponding to (1) can be defined as:

$$F^{ws}(x, \Lambda) = \sum_{i=1}^{m} \lambda_i f_i(x) \quad (2)$$

$$F^{Tc}(x, \Lambda, r^*) = Max_{1 \leq i \leq m} \{\lambda_i (f_i(x) - r_i^*)\} \quad (3)$$

3. SEARCH OPERATORS

In this section, the components of the search strategies used in this research will be reviewed as follows:

3.1 Genetic operators

Crossover and mutation are the two most popular genetic operators. Crossover is the process of exchanging the genetic material of the parents to create new offspring. Whereas, the mutation operator is used to preserve the diversity of the population during generations. In the literature, various types of crossover and mutation were proposed. These types are successfully used to handle different types of optimization problems in the continuous search domains. In this context, the SBX crossover [5], Multiple parents crossover [9] and polynomial mutation [5] will be focused.

3.1.1 SBX crossover

The simulated binary crossover (SBX) is widely used in practice. It has been found to work well in many test problems that have a continuous search space. From a pair of parents x^a and x^b, the SBX crossover produces an offspring y as follows:

$$y = \begin{cases} \frac{1}{2}\left((1+\beta)x^a + (1-\beta)x^b\right) & \text{if: } p \leq 0.5 \\ \frac{1}{2}\left((1-\beta)x^a + (1+\beta)x^b\right) & \text{otherwise} \end{cases} \quad (4)$$

$$\beta = \begin{cases} (2u)^{1/(1+\eta_c)}, & u \leq 0.5 \\ (1/(2-2u))^{1/(1+\eta_c)}, & \text{otherwise} \end{cases} \quad (5)$$

where $p, u \in [0, 1]$ are two uniform random numbers and η_c is the distribution index.

3.1.2 Multi-parents crossover

various multi-parents crossover were proposed in the literature for continuous search domains such as simplex (SPX) [24], parents centric (PCX) [6], etc. However, the new multiple parents crossover (MPC) proposed in [9] will be used here. According to equation (6), the MPC crossover constructs a new offspring y, from three different randomly selected parents x^a, x^b and x^c as follows:

$$y = \begin{cases} x^a + \beta \times (x^b - x^c) & \text{if: } p \leq \frac{1}{3} \\ x^b + \beta \times (x^a - x^c) & \text{if: } \frac{1}{3} < p \leq \frac{2}{3} \\ x^c + \beta \times (x^a - x^b) & \text{otherwise} \end{cases} \quad (6)$$

where $\beta \sim N(\mu, \sigma)$ is a Gaussian random number and $p \in [0, 1]$ is a uniform random number.

3.1.3 Polynomial mutation

In polynomial mutation, the probability to produce a child near to the parent is greater than the probability to produce one distant it. The mutant offspring $\acute{x}$ can be produced as:

$$\acute{x}_j = \begin{cases} x_j + \delta_j \times (b_j - a_j) & \text{with probability } p_m \\ x_j & \text{with probability } 1 - p_m \end{cases} \quad (7)$$

$$\forall \delta_j = \begin{cases} (2u_j)^{1/(1+\eta_m)} - 1, & u_j \leq 0.5 \\ 1 - (2 - 2u_j)^{1/(1+\eta_m)} & \text{otherwise} \end{cases} \quad (8)$$

where $u_j \in [0, 1]$ is a random number. The distribution index η_m and the mutation rate p_m are two control parameters. a_j and b_j are the lower and the upper limits of x_j.

3.2 Differential Evolution operator

Differential evolution (DE) is a simple and efficient search operator to solve optimization problems mainly in continuous domains [3, 23]. DE's success relies on the *differential mutation*, that employs difference vectors built with pairs of candidate solutions in the search domain. Each difference vector is scaled and added to another candidate solution, producing the so-called mutant vector. Then, DE recombines the mutant vector with the parent solution to generate a new offspring. The offspring replaces the parent only if it has an equal or better fitness. There are different strategies to carry out this process such as "DE/rand/n/bin", "DE/best/n/exp", "DE/rand-to-best/n/bin", etc, where n is the number of difference vectors used [23]. In this work, the "DE/rand/1/bin" strategy is considered. DE has some control parameters as scaling factor F, that used to scale the difference vectors, and crossover rate CR. Given a population P of N individuals, the idea is to randomly select three distinct individuals x^a, x^b and x^c from P for each target individual $x^i \in P, \forall i \in \{1, \cdots, N\}$. The mutant individual v^i is produced according to (9). Then, the *binomial crossover* is applied on v^i and x^i to produce a new offspring u^i as:

$$v^i = x^a + F \times (x^b - x^c) \quad (9)$$

$$u^i_j = \begin{cases} v^i_j & \text{if } rnd \leq CR, \text{ or } j = j_{rnd}, \\ x^i_j & \text{otherwise}, \forall j = 1, \cdots, n. \end{cases} \quad (10)$$

where $rnd \in [0,1]$ and $j_{rnd} \in \{1, \cdots, n\}$ is a random chosen index to insure that at least one component of u^i is contributed by v^i, n is the individual length and $CR \in [0,1]$.

3.3 Particle swarm optimization

PSO is a population-based stochastic optimization technique that simulates the social behavior of bird flocking and fish schooling. It was originally proposed in [17]. PSO consists of a population of particles (solutions). Each particle i has its own position x^i and moves through the search space with an adaptable velocity v^i towards the best position that it has achieved x^i_{pb} and the overall best solution x^i_{gb}. For each i^{th} particle at generation t, the velocity and the new position for the next generation can be evaluated as follows:

$$v^i(t{+}1) = w \cdot v^i(t) + c_1 r_1 (x^i_{pb} - x^i(t)) + c_2 r_2 (x^i_{gb} - x^i(t)) \quad (11)$$

$$x^i(t{+}1) = x^i(t) + v^i(t{+}1) \quad (12)$$

where $w \geq 0$ represents the inertia weight, c_1 and c_2 are the acceleration coefficients and $r_1, r_2 \sim U(0,1)^n$. For each j, If $x^i_j(t+1)$ violates its domain $[a_j, b_j]$, it will be repaired and also its velocity $v^i(t+1)$ will be reset as follows:

$$x^i_j(t+1) = \begin{cases} a_j & \text{if } x^i_j(t+1) < a_j \\ b_j & \text{if } x^i_j(t+1) > b_j \end{cases}$$
$$v^i_j(t+1) = -\gamma v^i_j(t+1) \quad (13)$$

where a_j and b_j are lower and upper bounds of the j^{th} component respectively. Here, the parameter γ is set to 1.

3.4 Guided Mutation operator

Guided mutation proposed in [11] provides an integration between global and local search capabilities, through guiding the rotation of the evolutionary strategy (ES) mutation ellipses, for global search, and using the regular ES operation to conduct local search to find the promising solution. In guided mutation, the new solution y is generated from its parent x using the guided target solution t as follows:

$$y_j = \begin{cases} x_j + 0.5(t_j - x_j) \times r + R \times N(0,1) & \text{with } p_m \\ x_j + 0.5(t_j - x_j) \times r & \text{with } 1 - p_m \end{cases} \quad (14)$$

$$\forall R = \begin{cases} 0.1 \times |t - x| & \text{if } 0.1 \times |t - x| > \mu \\ \mu & \text{otherwise} \end{cases} \quad (15)$$

where $r \sim N(0,1)$ is a Gaussian number and p_m is the mutation rate. The new offspring y consists of the current position of its parent x, the guided vector derived from its target t and the mutation step R which specified by the distance $|t - x|$ and bounded by the control parameter μ.

4. THE PROPOSED HESSA

In this context, the proposed HESSA is presented in more details. In the research work in [14, 16, 15], the influence of incorporating different cooperative metaheuristics in MOEA/D framework was examined for discrete search domains. The achieved results motivate us to extend the

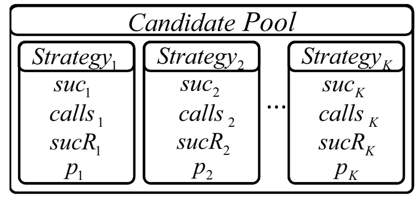

Figure 1: The structure of the candidate pool.

idea to the continuous case. However, an adaptive multiple search strategies are adopted for tackling continuous search domains. In the following subsections, the components of the proposed HESSA are discussed.

4.1 Multiple Search Strategies Adaptation

In this work, a pool of multiple search strategies is adopted to generate the new offspring solutions instead of using a single strategy as depicted by figure 1. To generate a new offspring, the candidate pool is accessed for selecting one search strategy for each target individual in the current population. During evolution, each certain number of consecutive pool's invokes is considered as a learning period (LP). The more successfully one strategy behaved in the previous learning period to generate promising solutions, the more probability it will be chosen in the current learning period to be used for generating the new offspring solutions. At each learning period, the probability of selecting each strategy from the candidate pool are summed to 1. These probabilities are adapted gradually during the evolution process. In the initial learning period, all strategies have the same chance to be selected, i.e., each strategy k has a probability $p_k = \frac{1}{K}$, where K is the total number of strategies in the candidate pool. During each learning period, each strategy k can be chosen to generate the new solution according to its probability p_k using the stochastic universal selection [1]. The number of selecting each strategy k is represented by $calls_k$. Each strategy is considered to achieve a success if it has the ability to generate an offspring capable of updating the current population. The number of successful calls for each strategy k is registered by suc_k. The number of invokes for the candidate pool is expressed as: $calls_{tot} = \sum_{l=1}^{L} \sum_{k=1}^{K} calls_{k,l}$, where, L is the total number of learning periods in the whole evolution. However, after each learning period l (when $calls_{tot} \% LP = 0$), the probability of selecting each strategy k for the next learning period $p_{k,l+1}$ will be adapted according to the following formulas:

$$p_{k,l+1} = \frac{sucR_{k,l}}{\sum_{k=1}^{K} sucR_{k,l}} \quad (16)$$

$$sucR_{k,l} = \begin{cases} \frac{suc_{k,l}}{calls_{k,l}} + \epsilon & \text{if } calls_{k,l} > 0, \forall k, l \\ \epsilon & \text{otherwise} \end{cases} \quad (17)$$

where $sucR_{k,l}$ is the success rate of the k^{th} strategy at the learning period l. The small constant value $\epsilon = 0.01$ is used to avoid the possible null success rates. Consequently, the strategies with null success rate have a chance to be chosen for generating offspring. Both $suc_{k,l}$ and $calls_{k,l}$ represent the number of successful invokes and the total number of invokes of the k^{th} strategy at the learning period l.

4.2 The HESSA framework

Like MOEA/D [25], the proposed approach uses a decomposition technique to convert the MOP in (1) into a

Table 1: Set of reproduction strategies used

Strategy	description
SBXPM	The SBX crossover is applied on two parents followed by polynomial mutation.
DEXPM	The differential evolution is applied on three selected parents followed by polynomial mutation.
MPCPM	The multiple parent crossover is applied on three selected parents followed by polynomial mutation.
GM	Guided mutation is used to produce an offspring from its parent and the global guide solution.
PSO	Particle swarm computes a new position from the current parent, its personal best and global guide.

set of single objective subproblems. The weighted Tchebycheff approach described in (3) is used in this study. However, if we have a set of N evenly distributed weight vectors $\{\Lambda^1, ..., \Lambda^N\}$, correspondingly after decomposition, we have N single objective subproblems. The proposed algorithm attempts to simultaneously optimize these subproblems. Each subproblem i has its own set of neighbors called B_i, which includes all the subproblems with the T closest weight vectors $\{\Lambda^{i1}, ..., \Lambda^{iT}\}$ to Λ^i in terms of Euclidean distance. The structure of the proposed framework is briefed as follows:

- A population P of N individuals, $P = \{x^1, ..., x^N\}$, where x^i represents the current solution of the i^{th} subproblem. Each individual x^i has its own velocity v^i, its personal best position x^i_{pb} and its age a_i.
- A set of N evenly distributed weight vectors $\{\Lambda^1,...,\Lambda^N\}$, correspond to the N subproblems. Each $\Lambda = [\lambda_1,..., \lambda_m]$ has m components correspond to m-objectives, such that: $\sum_{i=1}^{m} \lambda_i = 1$, $\forall \lambda_i \in \{0/H, 1/H, \cdots, H/H\}$ and $N = C^{H+m-1}_{m-1}$, $\forall H \in \mathbb{Z}^+$.
- A neighborhood B_i for each subproblem $i \in \{1, ..., N\}$, which includes all subproblems with the T closest weight vectors $\{\Lambda^{i1}, \cdots, \Lambda^{iT}\}$ to Λ^i.
- A set of adaptive reproduction strategies contained in a *Pool* for generating new solutions. Each strategy is selected according to its probability as mentioned above. Table (1) summarizes the set of adopted strategies.
- An external *archive* to collect efficient solutions explored over the search process. The archive also plays the role of global leaders repository.

After constructing the proposed framework, the proposed approach implements two basic phases. The first one is the *initialization* phase in which an initial population is randomly generated, whereas the second is the *main-loop* in which the search efforts are conducted to improve the initial population. The whole process is summarized in Alg.(1). Firstly, in lines (2-5), a set of N evenly distributed weight vectors is initialized. Then, the neighborhood structure B_i is constructed for each subproblem i by assigning all subproblems with the T closest weight vectors to Λ_i. The candidate *Pool* is also built using the adopted reproduction strategies. The archive and the evaluation counter are initialized. Secondly, the initial population is constructed in lines (6-12). For each subproblem i, the current solution x^i is randomly initialized. Then, x^i is evaluated and used to update the reference point r^* [25], the personal best x^i_{pb} and the *archive*. The velocity v^i and the age a_i are also initialized by 0. The i^{th} subproblem is appended to the population P. Now, the *main-loop* is executed until achieving the maximum evaluations $Mevals$ (lines 13-42). For each subproblem i, the mating/updating range M_i is chosen to be either the neighborhood B_i or the whole population. Then, three different parent solutions are randomly selected from M_i for reproduction. The global leader x^i_{gb} is randomly selected from the archive. A reproduction strategy S_k is also selected from the *Pool* for generating the new offspring y. According to the selected strategy S_k, the offspring y is generated. In case of using the guided mutation or the particle swarm, the age parameter a_i controls the generation process. In this case, if a_i exceeds the maximum allowable age T_a, a Gaussian value as: $N(\frac{1}{2}[x^i_{gb} - x^i_{pb}], |x^i_{gb} - x^i_{pb}|)$ is assigned to y. After that, the offspring y is evaluated and used to update the reference point r^*. The current population P is updated by invoking the UPDATESOLUTIONS module. The Archive is also updated by y according to the crowding distance. The evaluation counter is updated and checked. At the end of each learning period, the *Pool* is adapted by calculating the probability p_k for each strategy k according to (16). At the end of the evolution, the archive is returned.

In the UPDATESOLUTIONS module explained in Alg. (2), the offspring y updates the population P as follows: a random index j is selected from the updating range M_i. Then, the current solution of the j^{th} subproblem x^j is updated only if y achieves better scalar fitness according to (3). In this case, the success of the selected strategy Suc_k is increased. And the age a_j is reset. Also, the personal best x^j_{pb} is updated by the same manner. Finally the selected index j is eliminated from M_i. If the current solution x_j is not updated, its age a_j is increased. This process continues until updating t solution or M_i becomes empty.

5. EXPERIMENTAL DESIGN

In this paper, HESSA is verified using some of the state of the art MOEAs as: MOEA/D$_1$ [25], MOEA/D$_2$ [19] and dMOPSO [21]. A set of standard test problems which cover MOPs with different PFs' characteristics as convexity, concavity, disconnections and multifrontality is adopted. The test problems contain bi-objectives test MOPs including Fonseca, Kursawe [4], ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 proposed in [26]. They also contain three-objectives MOPs such as DTLZ2, DTLZ4, DTLZ6 and DTLZ7 proposed in [8]. Here, 30 decision variables are used for ZDT1, ZDT2 and ZDT3, whereas ZDT4 and ZDT6 are tested by 10 decision variables. In DTLZ2, DTLZ4 and DTLZ6, 12 decision variables are used, whereas DTLZ7 is tested by 22 decision variables. All experiments are performed on a PC with Intel Core i5-2400 CPU, 3.1 GHz and 4 GB of RAM.

5.1 Parameter settings

For each algorithm, the population size N and the maximum evaluations $Mevals$ are set to 100, 10000 for bi-objective problems and 300, 30000 for three-objective test problems respectively. The archive size and the learning period LP are set to N, 1000 respectively. In dMOPSO and HESSA, the inertia weight w and coefficients c_1, c_2 used in PSO are uniformly generated as $U(0.1, 0.5)$ for w and $U(1.2, 2)$ for c_1 and c_2 as used in [21]. For HESSA, the guided mutation parameter μ is set to 0.03 and the mutation rate P_m is set to $1/n$. In MPC crossover, β is set to $N(0.7, 0.1)$ as used in [9]. The other common parameters are depicted in table 2. Finally, the statistical analysis is applied on 30 independent runs for each test MOP.

Algorithm 1 :HESSA$(N, T, t, \delta, \eta_c, \eta_m, CR, F, T_a)$

Inputs:
N: *Population size or no. of subproblems*
T, t: *Min. neighborhood size, Max. replaced solutions*
δ: *prob. of selecting parents from neighborhood*

Begin:
$W_v \leftarrow \{\Lambda^1, \cdots, \Lambda^N\}$; ▷ *initialize a set of N evenly distributed weight vectors*
$B_i \leftarrow [i1, \cdots, iT]$; $\forall i = 1, ..., N$ ▷ *where* $\Lambda^{i1}, ..., \Lambda^{iT}$ *are T closest to* Λ^i
$Pool \leftarrow$ ConstructPool(sbxpm,dexpm,mpcpm,gm,pso); ▷ *5 strategies*
$Eval \leftarrow 0$; $Arch \leftarrow \varnothing$; ▷ *initialize Eval & Empty archive*
for $i \leftarrow 1$ to N **do**: ▷ *Initialization phase*
  $x^i_j \leftarrow U(a_j, b_j), \forall j = 1, ..., n.$ ▷ *get a uniform random* $x_j \in [a_j, b_j]$
  $r^* \leftarrow$ Evaluate&Update(x^i); ▷ *evaluate* x^i *and update ref. point* r^*
  $x^i_{pb} \leftarrow x^i$; $v^i \leftarrow 0$; $a_i \leftarrow 0$ ▷ *Initialize personal best,velocity & age*
  $P \leftarrow$ AddSubProblem$(x^i, \Lambda^i, v^i, x^i_{pb}, a_i)$; ▷ *add* i^{th} *subproblem*
  $Arch \leftarrow$ UpdateArchive(x^i); $Eval++$; ▷ *update Arch & Eval*
end for
while $(Eval < Mevals)$ **do**: ▷ *Main Loop*
  for $i \leftarrow 1$ to N **do**: ▷ *determine the mating/updating rang* M_i
    $M_i \leftarrow \begin{cases} B_i & \text{if}(rnd \in [0,1] < \delta) \\ 1, ..., N & \text{otherwise} \end{cases}$
    $x^a, x^b, x^c \leftarrow$ Selection(M_i, i); ▷ *Where:* $x^i \neq x^a \neq x^b \neq x^c$
    $x^i_{gb} \leftarrow$ SelectGlobalBest$(Arch)$; ▷ *randomly select Global guide*
    $S_k \leftarrow$ SelectStrategy$(Pool)$; ▷ *select a strategy* S_k *from Pool*
    $calls_k \leftarrow calls_k + 1$; ▷ *update # of calls for strategy* S_k
    if $(S_k =$ "sbxpm"$)$ **then**: ▷ *SBX crossover then Poly. mutation*
      $y \leftarrow$ Crossover(x^a, x^b);
      $y \leftarrow$ PolyMutation(y);
    else if $(S_k =$ "dexpm"$)$ **then**: ▷ *Diff. Evolution & Poly. mutation*
      $y \leftarrow$ DiffEvolution$(x^i, x^a, x^b, x^c, CR, F)$;
      $y \leftarrow$ PolyMutation(y);
    else if $(S_k =$ "mpcpm"$)$ **then**: ▷ *MP crossover & Poly. mutation*
      $y \leftarrow$ MPCrossover$(x^a, x^b, x^c, x^i_{gb})$;
      $y \leftarrow$ PolyMutation(y);
    else if $(S_k =$ "gm"$)$ **then**: ▷ *Apply Guided Mutation or reset y*
      $y \leftarrow \begin{cases} \text{GuidedMutation}(x^i, x^i_{gb}); & \text{if:} a_i < T_a \\ N(\frac{1}{2}[x^i_{gb} - x^i_{pb}], |x^i_{gb} - x^i_{pb}|) & \text{otherwise} \end{cases}$
    else if $(S_k =$ "pso"$)$ **then**: ▷ *Apply Particle Swarm or reset y*
      $y \leftarrow \begin{cases} \text{PSO}(x^i, x^i_{pb}, x^i_{gb}, v^i, a_i); & \text{if:} a_i < T_a \\ N(\frac{1}{2}[x^i_{gb} - x^i_{pb}], |x^i_{gb} - x^i_{pb}|) & \text{otherwise} \end{cases}$
    end if ▷ *The End of Reproduction*
    $r^* \leftarrow$ Evaluate&Update(y); ▷ *evaluate y and update ref. point* r^*
    $P \leftarrow$ UpdateSolutions(y, t, M_i, P, S_k, r^*);
    $Arch \leftarrow$ UpdateArchive(y, S_k); ▷ *crowding distance*
    $calls_{tot}++$;$Eval++$; ▷ *update total calls & evaluations*
    if $(calls_{tot} \% LP = 0)$ **then**: ▷ *The End of learning period*
      $Pool \leftarrow$ AdaptPool$(Pool)$; ▷ *recalculate* p_k *for each strategy k*
    end if ▷ *The End of Pool Adaptation*
  end for ▷ *The End of Generation*
end while
return $Arch$;
End

Algorithm 2 :UpdateSolutions(y, t, M_i, P, S_k, r^*)

Inputs:
y, t: *the new solution, Max. replaced solutions*
M_i, P: *the updating rang of subproblem i, the population*
S_k, r^*: *the selected reproduction strategy, reference. point*

Begin: $c \leftarrow 0$;
while $(c < t \, and \, M_i \neq \varnothing)$ **do**: ▷ *update Loop*
  $j \leftarrow$ SelectRandomIndex(M_i); ▷ *randomly select index j*
  if $(F^{Tc}(y, \Lambda^j, r^*) \leq F^{Tc}(x^j, \Lambda^j, r^*))$ **then**: ▷ *success case*
    $x^j \leftarrow y$; $c++$; $a_j \leftarrow 0$; ▷ *update* j_{th} *subproblem & reset age*
    $suc_k \leftarrow suc_k + 1$; ▷ *update # of success for strategy* S_k
    if $(F^{Tc}(x^j, \Lambda^j, r^*) \leq F^{Tc}(x^j_{pb}, \Lambda^j, r^*))$ **then**:
      $x^j_{pb} \leftarrow x^j$; ▷ *update the personal best*
    end if
    $M_i \leftarrow$ RemoveIndex(M_i, j); ▷ *exclude j from* M_i
  else: $a_j \leftarrow a_j + 1$; ▷ *update age* a_j
  end if
end while
return P; ▷ *return the updated population P*
End

Table 2: Set of common parameters

Parameters	MOEAD_1	MOEAD_2	dMOPSO	HESSA
Neighborhood size:T	30	30	-	30
Max Replaced Sols:t	-	2	-	2
Parents selection:δ	-	0.9	-	0.9
Crossover rate/index:p_c, η_c	1,20	-	-	1,20
Mutation rate/index:p_m, η_m	$1/n$, 20	$1/n$, 20	-	$1/n$, 20
DE parameters:CR, F	-	1,0.5	-	1,0.5
Age threshold:T_a	-	-	2	2
PBI penalty value:θ	-	-	5	-

5.2 Assessment Metrics

Let $A, B \subset \Re^m$ be two approximations to PF, $P^*, r^* \subset \Re^m$ be a reference set and a reference point respectively. The following metrics can be expressed as:

1. **The Set Coverage** (I_C)[28] is used to compare two approximation sets. The function I_C maps the ordered pair (A, B) to the interval $[0, 1]$ as:

$$I_C(A, B) = |\{u | u \in B, \exists v | v \in A : v \preccurlyeq u\}| / |B| \quad (18)$$

where $I_C(A, B)$ is the percentage of the solutions in B that are dominated by at least one solution from A. $I_C(B, A)$ is not necessarily equal to 1-$I_C(A, B)$. If $I_C(A, B)$ is large and $I_C(B, A)$ is small, then A is better than B in a sense.

2. **The Hypervolume** (I_H)[28] for a set A is defined as:

$$I_H(A) = \mathcal{L}(\cup_{u \in A} \{y | u \preccurlyeq y \preccurlyeq r^*\}) \quad (19)$$

where $\mathcal{L}$ is the Lebesgue measure of a set. $I_H(A)$ describes the size of the objective space that is dominated by A and dominates r^*. We use the referenced indicator such that: $I_{RH}(A) = I_H(P^*) - I_H(A)$ and r^* is the nadir vector included in P^*.

3. **The Generational** (I_{GD}) and Inverted Generational Distance (I_{IGD}) of a set A are defined as:

$$\begin{aligned} I_{GD}(A, P^*) &= \tfrac{1}{|A|} \textstyle\sum_{u \in A} \{min_{v \in P^*} d(u, v)\} \\ I_{IGD}(A, P^*) &= \tfrac{1}{|P^*|} \textstyle\sum_{u \in P^*} \{min_{v \in A} d(u, v)\} \end{aligned} \quad (20)$$

where $d(u, v)$ is the Euclidean distance between u, v in $\Re^m$. The $I_{GD}(A, P^*)$ measures the average distance from A to the nearest solution in P^* that reflects the closeness of A to P^*. In contrast, the $I_{IGD}(A, P^*)$ measures the average distance from P^* to the nearest solution in A that reflects the spread of A to a certain degree. The lower value of both $I_{GD}(A, P^*)$ and $I_{IGD}(A, P^*)$ means the better quality of A in terms of convergence and diversity respectively.

4. **The unary additive Epsilon** $(I_{\epsilon+})$ is defined as:

$$I_{\epsilon+}(A, P^*) = inf_{\epsilon \in \Re}\{\forall z^2 \in P^*, \exists z^1 \in A : z^1 \preccurlyeq_{\epsilon+} z^2\} \quad (21)$$

where $z^1 \preccurlyeq_{\epsilon+} z^2 \Leftrightarrow 1 \leq j \leq m : z^1_j \geq z^2_j - \epsilon$. it gives the minimum ϵ value by which each point in P^* can be decreased such that the resulting transformed approximation set is weakly dominated by A.

Here, the True Pareto front for each test problem is used as the reference set P^*.

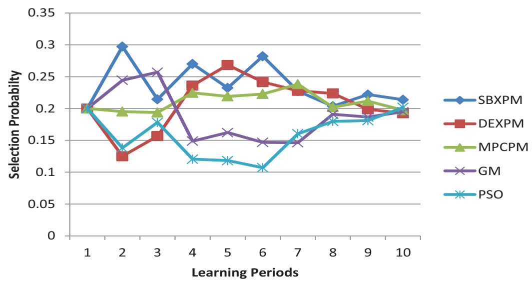

Figure 2: ZDT4 Search Strategy Adaptation

6. EXPERIMENTAL RESULTS

Here, the different simulation results are shown in details. Firstly, table 3 includes the results of the coverage I_C indicator. It contains the median values of I_C metric for the compared algorithms for each test MOPs used in this study. It is clear from the results that HESSA generally performs better than the other algorithms in all test MOPs except DTLZ2 and DTLZ4 with respect to $MOEAD_1$, and slightly have the same performance with dMOPSO in DTLZ6. The results depicted by table 4 express the average and the standard deviation of the referenced hypervolume I_{RH} indicator. The results indicate that HESSA is able to achieve better performance in all bi-objective test MOPs except Fonseca, and have the second best performance in three-objective problems except DTLZ4 in which it achieves the best performance. In table 5, the average and the standard deviation of the I_{GD} indicator are shown. It is clear that the results confirm the previous results of the I_{RH} indicator in most test problems. For DTLZ4, HESSA achieves the second best performance after $MOEAD_1$, whereas in DTLZ7, HESSA achieves the best performance followed by $MOEAD_1$. For the I_{IGD} indicator, the results is shown in table 6. According to these results, HESSA outperforms the other algorithms in all bi-objective test problems except Fonseca. It has also the second best performance in most three-objective test problems except DTLZ4 in which HESSA achieves the best performance. These results typically confirm the results of the I_{RH} indicator. Finally, table 7 shows the mean and the standard deviation of the epsilon $I_{\epsilon+}$ indicator. These results are nearly the same as those obtained by the previous indicators. HESSA achieves the best performance in all test problems except Fonseca, DTLZ2 and DTLZ4, in which $MOEAD_2$ has the best performance. Generally from the above results, HESSA achieves the best performance in most cases or at least the second best performance.

Figure 2 depicts the adaptation of different search strategies for ZDT4. Here, The run with the minimum I_{IGD} value is selected. It is clear that all search strategies begin with the same probability to be selected to launch the search process. During the evolutions, the performance of each search strategy is evaluated to adapt its probability of selection. As steady state is reached, all strategies go to nearly the same selection probability at the end of evolutions. This reflects the ability of HESSA to control the search process by launching the suitable search strategy at the appropriate time.

Here, the scatter plots presented in figures 3 and 4 contain the final Pareto fronts achieved by each algorithm for bi-objectives and three-objectives test problems respectively. The achieved final Pareto fronts is plotted versus the true Pareto front for each test problem. In these plots, The runs that achieve the minimum I_{IGD} values are considered.

Table 4: Results of I_{RH} indicator (Average,σ)

MOPs	$MOEAD_1$	$MOEAD_2$	dMOPSO	HESSA
Fonse	$5.03e-3_{6.5e-4}$	$3.64e-3_{4.3e-5}$	$1.19e-2_{1.4e-3}$	$4.08e-3_{1.1e-4}$
Kursa	$2.94e-1_{1.8e-2}$	$3.83e-1_{3.7e-2}$	$1.49e+0_{2.1e-1}$	$2.77e-1_{1.3e-2}$
ZDT1	$2.92e-2_{1.5e-2}$	$4.50e-1_{8.9e-2}$	$2.36e-2_{2.5e-3}$	$5.50e-3_{2.0e-4}$
ZDT2	$1.87e-1_{8.1e-2}$	$3.33e-1_{0.0e+0}$	$1.23e-1_{1.4e-1}$	$5.48e-3_{4.7e-4}$
ZDT3	$2.28e-2_{2.3e-2}$	$6.22e-1_{6.3e-2}$	$3.09e-2_{6.2e-3}$	$5.65e-3_{2.9e-4}$
ZDT4	$1.05e-1_{7.5e-2}$	$6.66e-1_{1.1e-8}$	$1.01e-1_{1.2e-1}$	$5.53e-3_{4.3e-4}$
ZDT6	$1.54e-2_{2.6e-3}$	$1.32e-1_{8.8e-2}$	$8.76e-3_{3.6e-3}$	$2.16e-4_{1.2e-5}$
DTLZ2	$4.61e-2_{1.3e-3}$	$4.84e-2_{1.1e-3}$	$1.21e-1_{6.7e-3}$	$4.64e-2_{1.1e-3}$
DTLZ4	$1.40e-1_{1.3e-1}$	$2.05e-2_{3.0e-2}$	$5.81e-2_{1.2e-2}$	$4.01e-4_{1.1e-3}$
DTLZ6	$1.11e-2_{6.9e-3}$	$2.67e-4_{7.6e-6}$	$7.71e-4_{1.6e-4}$	$3.11e-4_{3.4e-6}$
DTLZ7	$1.71e-1_{2.1e-2}$	$5.36e-1_{1.2e-1}$	$1.67e-1_{2.0e-2}$	$1.69e-1_{5.2e-3}$

Table 5: Results of I_{GD} indicator (Average,σ)

MOPs	$MOEAD_1$	$MOEAD_2$	dMOPSO	HESSA
Fonse	$1.56e-3_{2.4e-4}$	$9.96e-4_{3.3e-5}$	$4.49e-3_{5.1e-4}$	$1.14e-3_{5.8e-5}$
Kursa	$8.56e-3_{1.2e-3}$	$1.09e-2_{1.6e-3}$	$4.89e-2_{8.7e-3}$	$7.46e-3_{6.6e-4}$
ZDT1	$5.15e-3_{1.4e-3}$	$3.67e-1_{9.7e-2}$	$1.29e-2_{1.8e-3}$	$8.30e-4_{1.2e-4}$
ZDT2	$1.06e-3_{1.4e-3}$	$4.58e-1_{1.6e-1}$	$1.35e-2_{1.1e-2}$	$9.12e-4_{2.8e-4}$
ZDT3	$5.83e-3_{6.5e-3}$	$5.06e-1_{1.0e-1}$	$8.87e-3_{1.6e-3}$	$2.70e-3_{1.6e-4}$
ZDT4	$7.15e-2_{8.1e-2}$	$1.75e+1_{6.4e+0}$	$1.87e-3_{1.3e-3}$	$8.38e-4_{2.4e-4}$
ZDT6	$1.35e-2_{2.1e-3}$	$2.15e-1_{1.9e-1}$	$5.45e-3_{9.0e-3}$	$2.62e-3_{3.7e-5}$
DTLZ2	$5.85e-3_{1.1e-4}$	$7.85e-3_{2.4e-4}$	$6.85e-2_{5.0e-3}$	$6.33e-3_{1.7e-4}$
DTLZ4	$2.51e-2_{1.1e-2}$	$3.60e-2_{3.0e-3}$	$4.58e-2_{5.2e-3}$	$3.51e-2_{1.4e-3}$
DTLZ6	$4.19e-2_{2.8e-2}$	$3.72e-3_{8.4e-5}$	$4.42e-3_{2.3e-4}$	$3.85e-3_{6.8e-5}$
DTLZ7	$2.25e-2_{1.7e-3}$	$1.69e-1_{7.4e-2}$	$5.20e-2_{6.2e-3}$	$2.19e-2_{4.0e-4}$

Table 6: Results of I_{IGD} indicator (Average,σ)

MOPs	$MOEAD_1$	$MOEAD_2$	dMOPSO	HESSA
Fonse	$4.21e-3_{4.5e-4}$	$3.57e-3_{2.2e-5}$	$1.63e-2_{5.3e-3}$	$3.70e-3_{5.9e-5}$
Kursa	$4.24e-2_{1.2e-3}$	$4.42e-2_{1.2e-3}$	$1.06e-1_{2.1e-2}$	$4.20e-2_{5.2e-4}$
ZDT1	$3.87e-2_{3.2e-2}$	$3.90e-1_{9.6e-2}$	$1.48e-2_{1.5e-3}$	$4.05e-3_{7.1e-5}$
ZDT2	$2.46e-1_{1.2e-1}$	$8.98e-1_{1.5e-1}$	$1.95e-1_{2.7e-1}$	$4.00e-3_{1.2e-4}$
ZDT3	$2.67e-2_{2.7e-2}$	$4.66e-1_{7.6e-2}$	$1.75e-2_{2.4e-3}$	$1.06e-2_{1.1e-4}$
ZDT4	$1.05e-1_{6.8e-2}$	$6.48e+0_{2.8e+0}$	$1.60e-1_{1.8e-1}$	$4.11e-3_{1.3e-4}$
ZDT6	$1.93e-2_{3.4e-3}$	$1.63e-1_{2.2e-1}$	$3.63e-3_{1.1e-3}$	$1.89e-3_{1.6e-5}$
DTLZ2	$3.72e-2_{2.0e-4}$	$3.81e-2_{3.4e-4}$	$7.33e-2_{4.6e-3}$	$3.73e-2_{2.3e-4}$
DTLZ4	$1.69e-1_{1.4e-1}$	$2.99e-2_{5.2e-3}$	$4.41e-2_{6.5e-3}$	$2.98e-2_{2.0e-3}$
DTLZ6	$3.82e-2_{2.6e-2}$	$4.39e-3_{3.2e-5}$	$7.72e-3_{7.2e-4}$	$4.52e-3_{1.5e-5}$
DTLZ7	$2.24e-1_{1.5e-1}$	$3.76e-1_{1.8e-1}$	$8.94e-2_{5.1e-2}$	$1.15e-1_{3.5e-3}$

Table 7: Results of I_{ϵ_+} indicator (Average,σ)

MOPs	$MOEAD_1$	$MOEAD_2$	dMOPSO	HESSA
Fonse	$9.09e-0_{2.0e-0}$	$6.47e-3_{1.4e-4}$	$7.45e-2_{3.3e-2}$	$7.07e-3_{3.7e-4}$
Kursa	$8.79e-2_{1.6e-2}$	$1.09e-1_{1.4e-2}$	$4.57e-1_{1.9e-1}$	$8.24e-2_{9.0e-3}$
ZDT1	$1.10e-1_{7.0e-2}$	$4.88e-1_{1.2e-1}$	$2.88e-2_{3.7e-3}$	$8.25e-3_{4.1e-4}$
ZDT2	$7.21e-1_{2.6e-1}$	$1.42e+0_{1.7e-1}$	$3.26e-1_{4.4e-1}$	$7.44e-3_{4.5e-4}$
ZDT3	$1.27e-1_{1.8e-1}$	$8.66e-1_{1.7e-1}$	$4.64e-2_{1.2e-2}$	$1.51e-2_{4.2e-4}$
ZDT4	$2.13e-1_{1.0e-1}$	$6.80e+0_{2.8e+0}$	$2.53e-1_{2.3e-1}$	$8.91e-3_{8.4e-4}$
ZDT6	$2.85e-2_{4.9e-3}$	$3.09e-1_{2.4e-1}$	$2.99e-2_{8.6e-3}$	$5.03e-3_{2.3e-4}$
DTLZ2	$8.76e-2_{4.2e-3}$	$7.83e-2_{8.1e-3}$	$1.07e-1_{6.1e-3}$	$8.44e-2_{4.8e-3}$
DTLZ4	$4.33e-1_{3.3e-1}$	$8.24e-2_{5.7e-2}$	$1.38e-1_{1.8e-2}$	$6.39e-2_{1.0e-2}$
DTLZ6	$4.30e-2_{2.1e-2}$	$1.02e-2_{2.6e-4}$	$1.47e-2_{1.9e-3}$	$1.04e-2_{1.5e-4}$
DTLZ7	$6.16e-1_{6.3e-1}$	$9.65e-1_{6.0e-1}$	$2.49e-1_{2.1e-1}$	$1.68e-1_{3.9e-3}$

7. CONCLUSIONS

In this paper, a hybrid evolutionary approach with search strategy adaptation (HESSA) for handling multiobjective continuous problems was presented. In HESSA, the search process is controlled by adapting the search strategies used during the evolution process. HESSA was verified using a set of test MOPs commonly used in the literature. HESSA

Table 3: Results of the Coverage I_c indicator (Median)

$I_C(*,*)$	Fonseca	Kursawe	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ2	DTLZ4	DTLZ6	DTLZ7
(MOEAD1,HESSA)	$3.00e-02$	$5.81e-02$	$0.00e+00$	$1.00e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$6.30e-02$	$4.13e-02$	$0.00e+00$	$1.22e-01$
(HESSA,MOEAD1)	$1.80e-01$	$1.53e-01$	$6.67e-01$	$5.88e-02$	$4.60e-01$	$9.89e-01$	$9.90e-01$	$0.00e+00$	$0.00e+00$	$9.96e-01$	$1.73e-02$
(MOEAD2,HESSA)	$2.10e-01$	$6.90e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$8.37e-03$	$8.26e-03$	$2.05e-03$	$0.00e+00$
(HESSA,MOEAD2)	$3.00e-02$	$9.30e-02$	$1.00e+00$	$1.00e+00$	$9.83e-01$	$1.00e+00$	$6.72e-01$	$1.28e-01$	$1.58e-01$	$4.24e-03$	$9.06e-01$
(dMOPSO,HESSA)	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.00e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$6.54e-03$	$0.00e+00$
(HESSA,dMOPSO)	$4.70e-01$	$5.87e-01$	$8.74e-01$	$9.52e-01$	$8.28e-01$	$1.70e-01$	$9.09e-02$	$7.66e-01$	$1.18e-01$	$6.67e-03$	$5.19e-01$

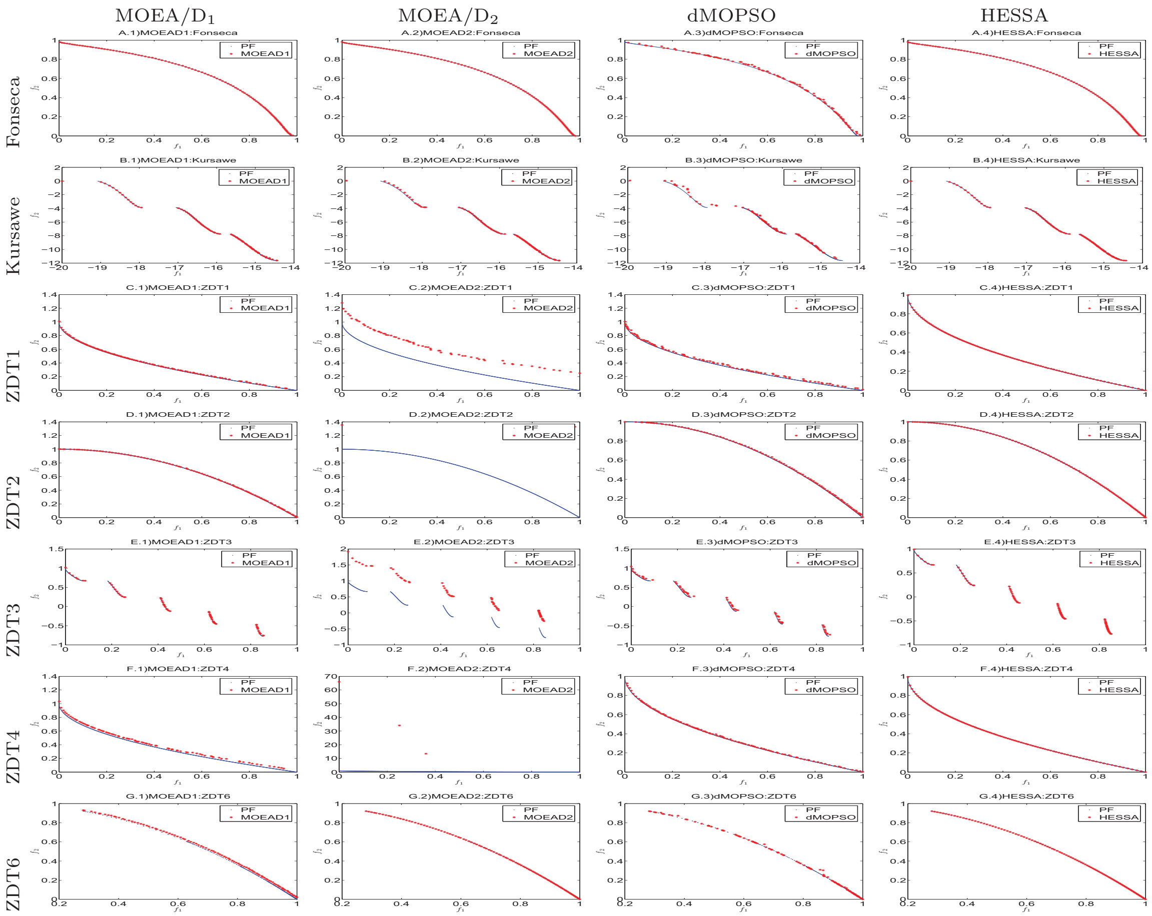

Figure 3: The Pareto fronts achieved for bi-objectives test problems

was also compared with three state of the art MOEAs. A set of quality indicators was also considered to evaluate the performance for all the compared MOEAs. The experimental results indicate the superiority of HESSA over both MOEA/D and dMOPSO on the most of test problems used. They also indicate that HESSA has an average performance highly competitive with respect to the compared MOEAs based on the assessment indicators used in this study. The contribution of HESSA is the combination among different cooperative search operators that intensify the search process to discover the promising regions in the search space and enhance the ability to explore good quality solutions. The second contribution is the ability to adapt the search process by using the suitable search operator to the problem on hand. In the future work, the tuning parameters of HESSA will be investigated as well as its convergence analysis. Additionally, HESSA will be extended to real applications and to include decision makers' preferences within the search.

8. REFERENCES

[1] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *the 2nd International Conference on Genetic algorithms and their application*, pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[3] U. Chakraborty. *Advances in Differential Evolution.* Studies in Computational Intelligence. Springer, 2010.

[4] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Sys.*, 9:115–148, 1995.

[6] K. Deb, A. Anand, and D. Joshi. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. *Evolu. Compu.*, 10(4):345–369, 2002.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast

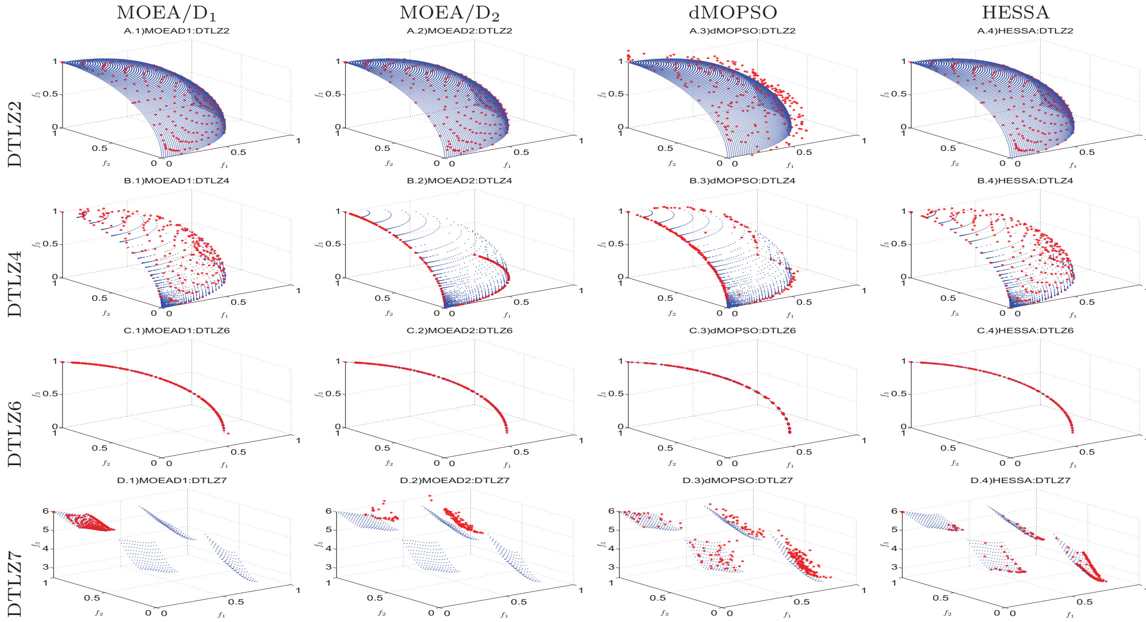

Figure 4: The Pareto fronts achieved for three-objectives test problems

Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6:182–197, 2000.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In A. Abraham, R. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005.

[9] S. M. Elsayed, R. A. Sarker, and D. Essam. GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. In *IEEE Congress on Evolutionary Computation*, pages 1034–1040. IEEE, 2011.

[10] M. Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Trans. Evolutionary Computation*, pages 425–442, 2004.

[11] C.-T. Hsieh, C.-M. Chen, and Y.-p. Chen. Particle swarm guided evolution strategy. In *the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 650–657, New York, NY, USA, 2007. ACM.

[12] A. Jaszkiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. *IEEE Trans. Evolutionary Computation*, 6(4):402–412, 2002.

[13] A. Jaszkiewicz. Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem. *IEEE Trans. Evolutionary Computation*, 7(2):133–143, 2003.

[14] A. Kafafy, A. Bounekkar, and S. Bonnevay. A hybrid evolutionary metaheuristics (HEMH) applied on 0/1 multiobjective knapsack problems. In Krasnogor and Lanzi [18], pages 497–504.

[15] A. Kafafy, A. Bounekkar, and S. Bonnevay. HEMH2: An Improved Hybrid Evolutionary Metaheuristics for 0/1 Multiobjective Knapsack Problems. In *the 9th International Conference (SEAL 2012), Hanoi, Vietnam, Dec. 16-19, 2012*, volume 7673 of *LNCS*, pages 104–116, New York, Germany, 2012.

[16] A. Kafafy, A. Bounekkar, and S. Bonnevay. Hybrid Metaheuristics based on MOEA/D for 0/1 multiobjective knapsack problems: A comparative study. In *IEEE Congress on Evol. Comp.*, pages 3616–3623. IEEE, 2012.

[17] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *IEEE international conference on neural networks IV*, pages 1942–1948, 1995.

[18] N. Krasnogor and P. L. Lanzi, editors. *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011.* ACM, 2011.

[19] H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *Trans. Evol. Comp*, 13(2):284–302, Apr. 2009.

[20] M. Lozano and C. García-Martínez. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & OR*, 37(3):481–497, 2010.

[21] S. Z. Martínez and C. A. C. Coello. A multi-objective particle swarm optimizer based on decomposition. In Krasnogor and Lanzi [18], pages 69–76.

[22] K. Miettinen. *Nonlinear multiobjective optimization.* Kluwer Academic Publishers, Boston, 1999.

[23] K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization* . Natural Computing Series. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[24] S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms. In *GECCO-99 Conference*, volume 1, pages 657–664, Orlando, Florida, USA, 13-17 July 1999.

[25] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation*, 11(6):712–731, 2007.

[26] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[27] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001.

[28] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol.Comp.*, 3(4):257–271, 1999.