A Hybrid Local Search Operator for Multiobjective Optimization

Alan Díaz-Manríquez, Gregorio Toscano-Pulido and Ricardo Landa-Becerra
Information Technology Laboratory, CINVESTAV-Tamaulipas
Parque Científico y Tecnológico TECNOTAM
Km. 5.5 carretera Cd. Victoria-Soto La Marina
Cd. Victoria, Tamaulipas 87130, MÉXICO
({adiazm,gtoscano}@tamps.cinvestav.mx)

Abstract—In recent years, the development of hybrid approaches to solve multiobjective optimization problems has become an important trend in the evolutionary computation community. Despite hybrid approaches of mathematical programming techniques with multiobjective evolutionary algorithms are not very popular, when both fields are successfully coupled, results are impressive. However, the main objective of this sort of hybridization relays on the needing of several executions of the mathematical approach in order to obtain a sample of the Pareto front, raising with this, the number of fitness function evaluations. However, the use of surrogate models has become a recurrent approach to diminish the number of function evaluations.

In this work, a hybrid operator that transforms the original multiobjective problem into a set of modified goal programming models is proposed. Furthermore, a local surrogate model is used instead of the real function in the hybrid operator. The goal programming model with the surrogate is optimized by a direct search method. Additionally, a standalone algorithm that uses the hybrid operator is here proposed. The new algorithm is validated using several test problems and performance measures commonly adopted in the specialized literature. Results indicate that the proposed operator gives rise to an effective algorithm, which produces results that are competitive with respect to those obtained by two well-known multiobjective evolutionary algorithms.

I. INTRODUCTION

The simultaneous satisfaction of a set of conflicting objective functions arises in many engineering an science areas. For this reason, the operation research community has developed several approaches to tackle this sort of problems [1]. These approaches can produce high quality solutions under certain conditions, even in some difficult problems [2, 3]. However, they also have some known issues e.g., they usually produce a single solution and therefore it is required to execute the algorithm several times in order to produce a desired number of trade-off solutions (requiring a high number of the objective function). Additionally, they usually assume certain conditions that are rarely available in nature. Evolutionary Algorithms (EAs) are population-based techniques that make a multidimensional search, finding more than one solution within an execution. Furthermore, EAs are able to locate solutions close to the global optimum even in highly rugged search spaces. Therefore, it was not coincidence that the EAs have been successfully applied to an important variety of difficult multiobjective optimization problems (MOPs). However, despite their success, when optimizing computationally expensive real-world optimization problems, even these approaches can fail when the number of fitness function evaluations required to give acceptable results exceeds the available computation time. Therefore, the EA community have proposed several ways to improve the efficiency of their existing approaches. Some of the major developments to achieve this goal are: hybrid algorithms [4]–[9], informed operators [10]–[14] and the use of surrogate models [15]–[19].

Among the hybrid approaches, well worth noting that the use of mathematical programming and MOEAs have produce successful approaches. For example, Landa et al. [20] could successfully deal with difficult bi-objective optimization problems by coupling an \( \epsilon \)-constraint approach with an evolutionary algorithm. Ranjithan et al. proposed CMEA [21]. This approach also hybridizes the \( \epsilon \)-constraint method with an evolutionary algorithm. However, in this approach, the authors try to reduce the computational cost of each independent optimization, recycling the final population of each optimization as the initial population of the next one. Baykasoglu et al. coupled a basic tabu search algorithm with a goal programming method [22]. The authors state that every optimization technique which works with more than one solution can effectively be used for solving MOPs. The results obtained by the authors were similar to previously reported results. Gen et al. showed that a genetic algorithm can be used for solving goal programming models [23].

One common issue of the previous approaches is the excessive number of evaluations needed to obtain a set of trade-off solutions. Therefore, it is possible to obtain better approaches if we adopt the use of surrogate models in order to diminish the number of evaluations functions needed to produce acceptable results.

In this work, we propose a hybrid local search approach that can be seen as an asexual hybrid operator that transforms the original multiobjective problem in a set of modified goal programming models. These models are then optimized using a numerical method (Nelder-Mead). In this regard, the use of Nelder-Mead will increase drastically the number of function evaluations. Therefore, we decided to replace the real function with local surrogate models in order to avoid the excessive number of function evaluations. Finally, to validate the hybrid operator, a standalone algorithm is proposed.

The remainder of this work is organized as follows. Details of the hybrid operator and the standalone algorithm are given in Section II. A comparative study with respect to other...
algorithms is presented in Section III. Finally, conclusions and future work are given in Section IV.

II. HYBRID OPERATOR

The hybrid operator here proposed can be an alternative or a complement to the traditional genetic variation operators. Additionally, we say that our approach is a local search approach since our original motivation was to develop an algorithm that after accepting a single solution, it could generate new non-dominated solutions. Therefore, the single solution enters in an improvement cycle that is conducted by the use of a modified goal programming technique.

A. Modified goal programming

In the original goal programming (GP) method was expressly created to handle multiobjective problems where the Decision Maker (DM) specifies optimistic aspiration levels (goals) for each objective function that he/she wishes to attain. Thus, the goal programming formulation tries to minimize the absolute deviation from the goals to the objectives. The simplest form of this method may be formulated as follows:

$$\min \sum_{i=1}^{k} |f_i(\bar{x}) - G_i|, \quad \text{subject to } \bar{x} \in Q$$  \hspace{1cm} (1)

where $G_i$ denotes the goal set by the decision maker for the $i$th objective function $f_i(\bar{x})$, the goal vector is denoted by $G = [G_1, G_2, \ldots, G_k]^T$, and $Q$ represents the feasible region. Then, the criterion is to minimize the sum of the absolute values of the differences between goals values and currently achieved values.

However, we would like to direct the search to specific regions of the search space. Therefore, a modified goal programming (modified-GP) method is here proposed. In the modified-GP, the DM assigns a goal that can be seen as a direction in which any solution represents an improvement with respect to convergence and/or diversity in the original multiobjective problem. Figure 1 shows the difference between the GP and the modified-GP. The goal vector restricts the search to a defined location (goal) in the objective space, while the goal direction allows the approach to search for a solution in a suitable direction.

In the modified-GP, the goal direction can be seen as a straight line. Therefore, the objective function tries to minimize the distance to the goal direction. However, this distance needs to be penalized if the obtained solution is in an opposite location to the desired direction (defined). For example, Figure 2 supposes that the solution to be improved is $A$ and the goal direction is $d_1$, if $B$ is evaluated, this solution will not be penalized because it is located in the same side of the search direction, so that, the evaluation of $B$ will be only the distance to the search direction $\text{dist}_B$. Conversely, if the solution to evaluate is $C$, this solution need to be penalized because it is located in the opposite direction, and its evaluation will be the distance to the search direction $\text{dist}_C$ and the distance to the perpendicular direction of the search direction $\text{dist}_C'$, so that, $F(C) = \text{dist}_C + \text{dist}_C'$.

The key feature required for the hybrid operator is the improvement of the existing solutions. To achieve this purpose, three goals for the classical GP are proposed. The three proposed goals are shown in Figure 3. It is obvious that goals $G_2$ and $G_3$ improve the convergence, whilst goals $G_1$ and $G_4$ can improve the diversity.

However, we would like to direct the search to specific directions. Therefore, we propose to replace the goal vector with goal directions. Figure 4 shows the four directions proposed in the proposed modified-GP.

The goal directions $d_2$ and $d_3$ were proposed in order to surpass $G_2$ and $G_3$, respectively. It is worth noticing, however, that we are not interested in accomplish $G_2$ and $G_3$, but to use their directions and go beyond their positions. Therefore, these directions emphasize the search for solutions in order to improve the algorithm's convergence. However, it is certainly more difficult to create directions that passes through $G_1$ and $G_4$, because it is necessary the degree of inclination of each direction. In order to avoid this inclination parameter, the up
(d₁) and reverse (d₄) directions were proposed. Then, the goal G₁ can be achieved with the collaboration of the up direction (d₁) and d₂. Moreover, the goal G₄ can be achieved with the collaboration of the reverse direction (d₄) and d₃. In Figure 5, we show how a combination of d₄ and d₃ can achieve the G₄ goal.

Since the directions are seen as straight lines and each of them is represented with an equation of the form \( Ax + By + C = 0 \). Hence, we have:

\[
d₁ = d₃ = y - f₂(\bar{x})
\]  

(2)

where \( f₁(\bar{X}) \) and \( f₂(\bar{X}) \) are the objective values of the solution that is desired to be improved. Since \( d₃ \) is in the same direction than \( d₁ \), and similarly \( d₂ \) is in the same direction than \( d₄ \), then only two equations are used. However, modified-GP can penalize solutions according to their search direction.

With the above procedure the hybrid operator will create four new solutions on each execution.

C. Optimization of the modified-GP with Nelder-Mead

EAs has been successfully applied to solve difficult optimization problems. However, since we intend to use the proposed operator as a local search operator, then an additional search approach to achieve the aspiration points is needed. In this work, we decided to use Nelder-Mead [24], since it is a reliable and efficient direct search method [25]. However, the main disadvantage on using this technique is the number of evaluations performed on each execution. For this sake, we replaced the original function \( f(\bar{x}) \) with a local surrogate model \( s(\bar{x}) \), and only the new solutions resulting from the optimization process (conducted with Nelder-Mead) will be evaluated in the original function \( f(\bar{x}) \). To create the local surrogate model is necessary to maintain a repository of previously evaluated solutions \( R \). Therefore, in order to select the solutions to create the surrogate model, we use a simple clustering algorithm (k-means) in \( R \), such that, we can find the solutions which belong to the same cluster (region) of the parent solution (\( \bar{x} \)). Algorithm 1 shows the procedure for the hybrid operator.

Díaz et. al [26] performed a comparative study of several surrogate models, according their results, the Radial Basis Function (RBF) is a surrogate model with a robust behavior, therefore, we decided to adopt it to perform the experimentation of this article.

Algorithm 1 Hybrid-Operator(\( \bar{x}, R \))

Input: Solution \( \bar{x} \), Repository \( R \)
Output: New solutions \( NS \)
\[
NS = \emptyset \\
s(\bar{x})=\text{Local-Surrogate-Model}(\bar{x}, R) \\
d =\text{Create search directions} \\
\text{for each Search direction } dᵢ \text{ do} \\
\quad \bar{Y} \leftarrow \text{Solve optimization problem for } dᵢ \text{ in } s(\bar{x}) \\
\quad NS \leftarrow NS \cup \bar{Y} \quad \text{(Add the solution to the new solutions created)} \\
\text{end for}
\]

D. A standalone algorithm for the hybrid operator

In order to evaluate the effectiveness of the proposed operator, a simple stand-alone algorithm based on the hybrid operator was proposed. The proposed algorithm works as follows: first an initial population of solutions \( P₀ \) of size \(|P|\) is created using a latin hypercube\(^2\) [27]. Then, the main procedure of the algorithm begins with the selection of the solutions that

\(^1\)The clustering algorithm is performed in the variable space
\(^2\)A latin hypercube is a statistical method for a stratified sampling that can be applied to multiple variables
our hybrid operator will try to improve. Therefore, $\beta$ solutions are selected in $P$ according to their dominance rank and their crowding distance. Then the hybrid operator is applied to those solutions. The resulting solutions from this procedure will be stored in $P$. Since $P$ is a fixed-size archive, then dominance rank and crowding distance will be applied again in order to maintain the file size. The Standalone algorithm pseudocode is shown in Algorithm 2.

**Algorithm 2 Standalone Algorithm**

| Input: $|P|$ (size of population), $t_{\text{max}}$ (number of iterations), $\beta$ (percentage of solutions to apply the operator) |
| Output: $P_{t_{\text{max}}}$ (Final population) |

1: $P_0 \leftarrow \text{GenerateRandomPopulation}()$
2: for $t \leftarrow 0$ to $t_{\text{max}}$ do
3:   $S \leftarrow \text{Choose } \beta \text{ solutions of } P_t \text{ (with dominance rank and crowding distance)}$
4:   for $j \leftarrow 0$ to $|S|$ do
5:     $\text{NewSolutions} \leftarrow \text{Hybrid-Operator}(S_j, P_t)$
6:     $NS \leftarrow NS \cup \text{NewSolutions}$
7: end for
8: $P_{t+1} \leftarrow P_t \cup NS$
9: $F \leftarrow \text{fast-non-dominated-sorting}(P_t)$
10: $P_{t+1} = \emptyset$
11: repeat
12: Assign crowding distance on $F_i$
13: $P_{t+1} \leftarrow P_{t+1} \cup F_i$
14: until $|P_{t+1}| + |F_i| \leq |P_t|$ |
15: Crowding distance on $F_i$
16: $P_{t+1} \leftarrow P_{t+1} \cup \text{(Choose the first } |P| - |P_{t+1}| \text{ individuals of } F_i)$
17: end for

### III. DESIGN OF EXPERIMENTS AND COMPARISON OF RESULTS

The proposed approach was evaluated using 6 test functions, the well-known ZDT test suite [28] and the Kursawe test problem [29].

In order to assess the performance of the proposed approaches, we decided to take three performance measures from the specialized literature.

1) The Averaged Hausdorff Distance ($\Delta_p$) [30] between the obtained Pareto front ($\mathcal{P} \mathcal{F}$) and the true Pareto front $\mathcal{P} \mathcal{F}^*$ is composed of slight modifications of generational distance and the inverted generational distance (called, $GD_p$ and $IGD_p$), which leads to a more fair indicator. The $GD_p$ performance indicator is stated as follow:

$$GD_p = \left( \frac{1}{n} \sum_{i=1}^{n} d_i^p \right)^{\frac{1}{p}}$$  \hspace{1cm} (4)

The ($IGD_p$) is similar to the $GD_p$ but uses the true Pareto front as reference; therefore, it compares each of the $\mathcal{P} \mathcal{F}$ elements with respect to the front produced by an algorithm. Finally, the $\Delta_p$ indicator refers to the maximum of $GD_p$ and $IGD_p$.

$$\Delta_p = \max(GD_p, IGD_p)$$  \hspace{1cm} (5)

where $p$ is the key to handle the outlier trade-off: the smaller $p$, the higher the averaging effect and the lower the influence of single outliers. On the other hand, if $p$ is increased, outliers influence the value of $\Delta_p$. In this work, $p = 2$ was used. In this metric, we will prefer smaller values of $\Delta_p$, since this represents a closer result to the true Pareto front.

2) Hypervolume ($HV$). The $HV$ computes the area covered for all the solutions in the approximated Pareto front $Q$ with the help of a reference point $W$. Equation (6) shows the mathematical definition of $HV$.

$$HV = \text{volume}(\bigcup_{i=1}^{Q} v_i)$$  \hspace{1cm} (6)

where, for each solution $i \in Q$, a hypercube $v_i$ is constructed using the reference point $W$. Therefore, $HV$ is the union of the volume of all the hypercubes.

3) Spacing ($S$). The spacing is computed with the relative distance between two consecutive solutions in the approximated Pareto front ($Q$). Equation 7 refers to the mathematical definition of the spacing.

$$S = \sqrt{\frac{1}{Q} \sum_{i=1}^{Q} (d_i - \bar{d})^2}$$  \hspace{1cm} (7)

where $d_i = \min_{j \in Q \land j \neq i} \sum_{m=1}^{k} |f_{i,m} - f_{j,m}|$ is the minimal distance between two solutions and $\bar{d}$ is the mean of the previously measured distances. Small values are preferred for this indicator.

The parameters used for the standalone algorithm were hand-tuned using a trial-and-error approach. The selected parameters are an initial population of $|P| = 100$, the number of solutions to apply the operator $\beta = 5$, the number of clusters created with k-means are 4.

In order to validate our proposal, we chose the two following approaches: 1) NSGA-II which is, by far, the most popular Pareto-based MOEA, and 2) MOEA/D, a more recent MOEA, based on decomposition, who has been found to be very effective. The parameters used in NSGA-II and MOEA/D were taken from their original proposals. Both approaches use a population size of 100, a mutation rate of $1/N$, where $N$ is the number of variables, a crossover rate of 0.8, a crossover index of 15, and a mutation index of 20. Additionally, MOEA/D uses $T = 20$ as the size of the neighborhood.

Since we would like to investigate the behavior of our proposal with respect to the compared approaches, we decided to measure the online convergence. For this sake, we performed 31 independent executions, and we measured $\Delta_p$ every 100 evaluations during 2,000 function evaluations.

Figure 6 shows the results obtained by the compared approaches in all the adopted problems. In this figures, x-axis shows the number of evaluations whilst y-axis shows the average performance according to $\Delta_p$. In problems ZDT1, ZDT2, ZDT3 and ZDT4, we can be observed that our proposed standalone algorithm outperformed the NSGA-II and MOEA/D approaches. From this figure, it is also clear that our standalone algorithm required approximately 400 evaluations in most of the problems to obtain a good results, while NSGA-II and MOEA/D failed to approximate the true Pareto front within
TABLE I
HYPERVOLUME (HV) OF THE 3 MOEAS IN THE 6 TEST FUNCTIONS

<table>
<thead>
<tr>
<th>Problem</th>
<th>NSGA-II</th>
<th>MOEA/D</th>
<th>Standalone</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>3.44 ±0.25</td>
<td>1.06 ±0.02</td>
<td>2.63 ±0.02</td>
</tr>
<tr>
<td>ZDT2</td>
<td>4.62 ±0.34</td>
<td>1.08 ±0.03</td>
<td>3.04 ±0.03</td>
</tr>
<tr>
<td>ZDT3</td>
<td>7.81 ±0.43</td>
<td>1.02 ±0.05</td>
<td>3.44 ±0.04</td>
</tr>
<tr>
<td>ZDT4</td>
<td>1.94 ±0.14</td>
<td>3.41 ±0.45</td>
<td>1.13 ±0.02</td>
</tr>
<tr>
<td>ZDT5</td>
<td>120.51 ±1.62</td>
<td>114.78 ±1.62</td>
<td>120.59 ±1.62</td>
</tr>
<tr>
<td>ZDT6</td>
<td>72.81 ±0.34</td>
<td>102.82 ±0.40</td>
<td>65.90 ±0.34</td>
</tr>
<tr>
<td>KURSAWE</td>
<td>113.79 ±1.33</td>
<td>107.88 ±2.78</td>
<td>120.59 ±0.04</td>
</tr>
</tbody>
</table>

TABLE II
SPACING (S) OF THE 3 MOEAS IN THE 6 TEST FUNCTIONS

<table>
<thead>
<tr>
<th>Problem</th>
<th>NSGA-II</th>
<th>MOEA/D</th>
<th>Standalone</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>0.05 ±0.00</td>
<td>0.07 ±0.03</td>
<td>0.02 ±0.00</td>
</tr>
<tr>
<td>ZDT2</td>
<td>0.08 ±0.04</td>
<td>0.03 ±0.05</td>
<td>0.06 ±0.07</td>
</tr>
<tr>
<td>ZDT3</td>
<td>0.06 ±0.02</td>
<td>0.07 ±0.04</td>
<td>0.04 ±0.02</td>
</tr>
<tr>
<td>ZDT4</td>
<td>4.62 ±0.01</td>
<td>4.16 ±2.12</td>
<td>0.17 ±0.63</td>
</tr>
<tr>
<td>ZDT5</td>
<td>72.81 ±0.34</td>
<td>102.82 ±0.40</td>
<td>65.90 ±0.34</td>
</tr>
<tr>
<td>ZDT6</td>
<td>120.51 ±1.62</td>
<td>114.78 ±1.62</td>
<td>120.59 ±1.62</td>
</tr>
</tbody>
</table>

2,000 evaluations. Moreover, in ZDT6, the performance of MOEA/D was slightly better than the standalone algorithm. However, it can be seen that the proposed approach had a faster convergence that the MOEA/D. Finally, in Kursawe the performance of all the algorithms were similar.

Additionally, the three approaches were executed for 2,000 evaluations and it was measured the hypervolume. In order to have more confident results, each MOEA was executed 31 times.

The application of the Hypervolume performance measure to the results obtained by the three approaches is shown in Table I. From these results, it is easy to see that the proposed approach outperformed the others in all the test problems. It is important to note, that despite ZDT4 test problem is a difficult problem the proposed approach could manage to outperform the other algorithms.

We decided to adopt a third performance measure so that we could be able to reach more general conclusions. Spacing (S) is a performance indicator that has been widely used in the specialized literature. Table II shows the results of the applications of this indicator to the algorithms’ outputs. Results show that the dispersion was competitive with respect to the obtained by other approaches in most of the problems. In ZDT6 the spacing reached for our proposal was the worst. However, it is necessary indicate that the proposed algorithm was the algorithm that achieved the best convergence with respect to hypervolume (see Table I).

Since performing a visual comparison can also be sometimes helpful, we decided to plot the median of the executions with respect to the Hypervolume performance measure. The obtained Pareto fronts are shown in Figure 7. From this figure, it is easy to see that the proposed approach could successfully achieve the true Pareto front. Additionally, despite ZDT4 was a very difficult test function for MOEA/D and NSGA-II, our proposal was capable to converge to its true Pareto front using only 2,000 evaluations. Finally, the standalone algorithm behaved well for the rest of the problems (including Kursawe). For this reasons, it can be said that the proposed hybrid operator helped to the standalone algorithm to accelerate its convergence and also to explore different regions of the search space.

A. Statistical analysis

Additionally, a statistical analysis was performed in order to verify the statistical differences among the proposed approach, the NSGA-II and MOEA/D. The comparison methodology is as follows: first, the Shapiro-Wilk test was performed to verify the normality of the data distributions. If both samples are normal distributed, the homogeneity of their variances is verified with a Bartlett’s test. If both variances are homogeneous then an ANOVA test is performed to verify the statistical significance, otherwise a Welch’s test is used. For non-normally distributed data, the Kruskal-Wallis test was performed to verify the statistical significance. In all the test a significance level of 0.05 was used. The results are presented in Tables III and IV. The values marked with “+” suggest a statistically significant difference in performance in favor of our proposal, while the values marked with “-” suggest a difference in favor of the state-of-the-art algorithms. Additionally, the values in blank suggest a similar performance. The obtained results confirm the above discussion. The proposed approach showed a good general performance for all the instances, when both, HV and Δp, performance indicators were measured. However, in Kursawe, the NSGA-II and MOEA/D outperformed our proposal in Δp with few evaluations, this is because Kursawe can be solved easily in 2,000 evaluations by these algorithms.

IV. Conclusions

The purpose of this paper was to propose a hybrid operator that uses a mathematical programming technique with a classical numerical method. The adopted mathematical programming technique was the goal programming method. However, to control the search direction a modified-GP was proposed. This modified-GP was solved using the Nelder-Mead algorithm. In order to decrease the number of evaluations needed by the Nelder-Mead algorithm, the real function was replaced with the use of local surrogate models.

The proposed operator was embedded into a simple standalone algorithm in order to evaluate its performance. The algorithm was compared in six test problems borrowed from the literature. Results showed that the new operator was able to guide the search to the true Pareto front in all the adopted problems. Moreover, the proposed algorithm was compared with respect to NSGA-II and MOEA/D (two popular MOEAs), results showed that while the state-of-the-art algorithms could not convert using 2,000 evaluations, the standalone algorithm reached the Pareto fronts of most of the adopted problems within only approximately 400 evaluations. As future work, we would like to explore the hybridization of the hybrid operator here proposed with an evolutionary algorithm.
Fig. 6. $\Delta_p$ of NSGA-II, MOEA/D and standalone algorithm in the six test problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
<th>KURSAWE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluations</td>
<td>0 500 1000 1500 2000</td>
<td>0 500 1000 1500 2000</td>
<td>0 500 1000 1500 2000</td>
<td>0 500 1000 1500 2000</td>
<td>0 500 1000 1500 2000</td>
<td>0 500 1000 1500 2000</td>
</tr>
</tbody>
</table>

TABLE III
HV: STATISTICAL ANALYSIS

<table>
<thead>
<tr>
<th>Statistic</th>
<th>NSGA-II</th>
<th>MOEA-D</th>
<th>Standalone algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
<tr>
<td>ZDT2</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
<tr>
<td>ZDT3</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
<tr>
<td>ZDT4</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
<tr>
<td>ZDT6</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
<tr>
<td>KURSAWE</td>
<td>+ + + +</td>
<td>+ + + +</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
</tbody>
</table>

Standalone algorithm/NSGA-II
Standalone algorithm/MOEA-D

- Table III: HV: Statistical Analysis
Fig. 7. Pareto fronts for (a): ZDT1, (b): ZDT2, (c): ZDT3, (d): ZDT4, (e): ZDT6, (f): Kursawe using 2,000 evaluations
REFERENCES


TABLE IV

Δp: Statistical Analysis

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
<th>KUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MOEA/D</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
<th>KUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>800</td>
<td>1200</td>
<td>1600</td>
<td>2000</td>
<td>400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delta</th>
<th>Table IV</th>
<th>Statistical Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δp</td>
<td>ZDT1</td>
<td>ZDT2</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>