PHC-NSGA-II: A Novel Multi-objective Memetic Algorithm for Continuous Optimization

Slim Bechikh, Nabil Belgasmi, Lamjed Ben Said, and Khaled Ghédira
Higher Institute of Management of Tunis
Intelligent Information Engineering Laboratory
Tunis, Tunisia
{slim.bechikh, nabil.belgasmi}@gmail.com, {lamjed.bensaid, khaled.ghedira}@isg.rnu.tn

Abstract

We introduce in this paper a new multi-objective memetic algorithm. This algorithm is a result of hybridization of the NSGA-II algorithm with a new designed local search procedure that we named Pareto Hill Climbing. Verification of our novel algorithm is carried out by testing it on two sets of multi-objective test problems and comparing it to other multi-objective evolutionary algorithms (MOEAs) and other multi-criterion memetic algorithms (MMAs). Simulation results show the algorithm ability in tackling continuous multi-objective problems in terms of convergence and diversity. Our hybrid algorithm (1) outperforms pure MOEAs, (2) is competent with other gradient based MMAs, and (3) can solve non differentiable problems.

1. Introduction

Multi-objective optimisation, also known as multi-criteria optimization [1], consists in simultaneously optimizing two or more conflicting objectives under certain constraints. Real world optimization problems deal with competing incommensurable objectives in a complex search space. If a multi-objective problem is well-formed, there should not be a single solution that simultaneously minimizes each objective to its fullest, but we are looking for a set of compromise solutions. Evolutionary algorithms are population based metaheuristics that operate on a set of individuals in order to find trade-off solutions as most as possible. This discernible characteristic make them the most adapted to solve multi-objective optimization problems. Several works have been done in this field. For a comprehensive survey, refer to [2]. In most cases, as set out in [3], genetic algorithms are defined to be not enough effective because the crossover and mutation operators do not allow to intensify the search sufficiently. The mutation operator is typically expected to make a slight modification to an individual. Its role is to encourage the diversification of individuals while the selection is charged to conserve the best of them. Evolutionary algorithms researchers have suggested several approaches to overcome the weakness of these search methods and improve their performance by increasing their convergence rate and solutions diversity. One promising approach is hybridization [4] consisting in the merge between evolutionary algorithms and other search methods. The most common and effective technique is to incorporate local search (LS) into evolutionary algorithms. The local search operator replaces or follows the mutation operator, and then helps to intensify the research in various areas pointed by the genetic mechanisms: selection and crossover, we call this type of hybrid algorithms Memetic Algorithms (MAs) [4] or more marginally genetic local search algorithms [5]. We can classify MAs on the basis of the used LS method type:

- *Gradient based schemes*: This type of MAs incorporates LS methods that exploit gradient information. For example, two versions of the NSGA-II [15] algorithm were hybridized with the sequential quadratic programming (SQP): SBX-NSGA-II [7] and PCX-NSGA-II [6]. Another work [8] addresses two hybrid algorithms, named NSGA-SQP and SPEA-SQP, incorporating also the SQP procedure in the two MOEAs NSGA-II and SPEA2 [25] by the means of a modified version of the ε-constraint method. All these works have shown good results in terms of convergence and CPU time.

1082-3409/08 $25.00 © 2008 IEEE
DOI 10.1109/ICTAI.2008.87

- *Neighbourhood based schemes*: Here, MAs integrate LS methods that explore solutions neighbourhoods without using gradient information such that MOGLS [9], PMA [10], IMMOGLS [11] and M-PAES [12].

When designing multi-objective MAs, we are face to face to several design issues. Most of them are evoked in single objective case [13]. These issues can be summarized as follows:

- How to incorporate LS method into MOEAs?
- How to generate neighbourhood?
- How much time does the LS take?
- How often LS should be performed?
- How to select solutions for LS?
- What is the replacement strategy?
- How to maintain population diversity?

For an attractive survey of multi-objective MAs detailing these design problems, refer to [14].

In this work, we introduce a new multi-objective memetic algorithm, PHC-NSGA-II, taking its origin from the hybridization of the NSGA-II algorithm with our new designed LS method that we called *Pareto Hill Climbing (PHC)*. Then, we show our algorithm ability to tackle continuous multi-objective problems using some well known benchmarks [20]-[23]. The remainder of this paper is organised as follows. In section II, we describe the design of our PHC-NSGA-II algorithm. Section III is devoted to experiments and comparisons. Finally, in section IV, we state our concluding remarks and future works.

2. PHC-NSGA-II description

PHC-NSGA-II is a multi-objective genetic local search algorithm integrating the so-called PHC procedure into the well known multi-criterion genetic algorithm NSGA-II [15]. We first describe the NSGA-II algorithm. Second, we present our new local search procedure PHC, and finally we illustrate our new hybrid algorithm PHC-NSGA-II.

2.1. NSGA-II

NSGA-II is an improved version of the non-dominated sorting genetic algorithm NSGA [16]. Initially, a parent population P_0 is created randomly. Besides, this population is sorted based on the non-domination[1] concept. For each individual, fitness is assigned and it is equal to its non-domination rank. After that, a child population Q_0 is created from P_0 by performing binary tournament selection, crossover and mutation operations. From the first generation award, the procedure is different. First, the two populations P_t and Q_t are combined to form a population R_t of size 2N ($|P_t| = |Q_t| = N$). Second, a non-dominated sorting is used to classify the entire population R_t. Once, the non-dominated sorting is over, the population R_t becomes subdivided in several categories. All elements of the same category have the same non-domination rank. Each category is called a non-dominated front. After that, the new parent population P_{t+1} is filled with individuals of the best non-dominated fronts, one at a time. Since the overall population size is 2N, not all fronts may be accommodated in N slots available in the new population P_{t+1}. When the last allowed front is being considered, it may contain more solutions then the remaining available slots in P_{t+1}. Instead of discarding arbitrary some elements from the last front, it will be judicious to use niching strategy to choose individuals from the last front which reside in the least crowded region in this front. At this point, it may be practical to discuss the crowding distance (CD) assignment and the crowded tournament selection operator used for the NSGA-II selection. But, we will not plunge into too many details on that. More on this can be obtained from [17]–[15]. Figure 1 illustrates the above mentioned mechanism.

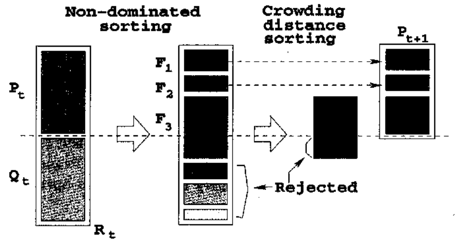

Figure 1. NSGA-II basic iteration [17].

2.2. Pareto Hill Climbing (PHC)

The main issue when designing a local search procedure for the multi-objective case is how to adapt it to optimize more than one objective simultaneously. The basic idea of our hill climbing method is taking advantages of Pareto dominance concept. First a neighbourhood is generated, then our procedure iterates on each one. Only dominated neighbours are rejected. The second crucial issue is how constructing neighbours set when dealing with continuous search space. Unlike the discrete case, it is impossible to explore the overall solution neighbourhood. Hence,

[1] A solution s_i is said to dominate another solution s_j, if and only if s_i is no worse than s_j with respect to all objectives and s_i is strictly better than s_j in at least one objective. If none of the two solutions dominates the other one, s_i and s_j are said non-dominated solutions.

researchers have proposed various methods for sampling the continuous search space, e.g. the n-balls method [26] and the hyper-rectangles methods [27]. To avoid additional computational cost, we use polynomial mutation [18] to generate neighbourhood. Since a set of mutated individuals are similar, so they build a neighbourhood. The PHC algorithm is presented in figure 2. First, the neighbours set of the current solution s (which is subject to local search) is generated by the mean of the mutation operator. Then, we iterate on each individual. An individual replaces current solution unless it is dominated. This process is repeated for a number of iterations *N_MAX_ITERATIONS* defined by the user (We suggest 5 ≤ *N_MAX_ITERATIONS* ≤ 25).

```
Input:
 s: solution subject to PHC.
N_MAX_ITERATIONS: number of iterations to be preformed.
Neighbourhood_size: neighbours number of solution s to generate.
Output:
s': solution obtained by PHC.
Begin
    Repeat
      N(s) ← Neighbourhood_generating (s, Neighbourhood_size)
      For each neighbour ∈ N(s) do
            If s does not dominate neighbour Then
                  s ← neighbour
            EndIf
      EndFor
    Until (N_MAX_ITERATIONS)
      s' ← s
End
```

Figure 2. PHC algorithm.

2.3. PHC-NSGA-II

In this subsection, we describe the design of our new algorithm by answering design issues described in the first section. For a generation t ≥ 1, the PHC-NSGA-II main iteration algorithm is presented in figure 3. As we have mentioned, the PHC-NSGA-II algorithm is a modified version of NSGA-II which integrates the new local search procedure PHC. The main idea of our algorithm is the following: once the parent population P_{t+1} is created, it is subject to the PHC procedure. This procedure is applied one time to each front F_i (i=1, 2,…, fronts_number, fronts_number corresponds to the last rank) from P_{t+1}. The solutions selected for LS are the least crowded solutions of front F_i in objective space. In fact, PHC-NSGA-II selects the *n* solutions having the greatest crowding distance values in front F_i to perform LS in order to explore the neighbourhood of each one (n=2M+1, M is the number of objectives). This set of solutions is called *PHC_set* in our algorithm depicted in figure 3. Afterward, each solution from *PHC_set* performs PHC and the resultant solutions set *Improved_set* is added to F_i. Once, the front F_i is updated, the replacement strategy must take care of population diversification. For these reason, we introduce a new distance measure which estimates a solution crowding distance in the decision space and not in the objective space. Similarly to that one in NSGA-II, the distance of a solution x_i is computed as a function of the decision variables of the nearest neighbours x_{i+1} and x_{i-1}. Hence, we term it as *decision crowding distance (DCD)*. Figure 4 illustrates the DCD assignment procedure.

```
Input:
P_t: parent population.
Q_t: offspring population.
N: parent population size.
M: objectives number.
V: decision variables number.
Output:
P_t: updated parent population.
Q_t: updated offspring population.
Begin
     R_t ← P_t ∪ Q_t
     F ← Fast_non-dominated_sort (R_t, M)
     P_{t+1} ← ϕ
     i ← 1
     While | P_{t+1}| + |F_i| ≤ N do
          Crowding_distance_assignment (F_i, M)
       P_{t+1} ← P_{t+1} ∪ F_i
       i ← i + 1
     EndWhile
     Sort (F_i, <_n)
     P_{t+1} ← P_{t+1} ∪ F_i [1: (N-| P_{t+1}|)]
    For each front F_i ∈ P_{t+1} do
          PHC_set ← Local_search_selection (F_i)
        Improved_set ←ϕ
        For each individual ∈ PHC_set do
              Improved_set ← Improved_set ∪ PHC (individual)
      EndFor
       F_i ← F_i ∪ Improved_set
       Decision_crowding_distance_assignment (F_i, V)
       Sort (F_i, DCD,'acsend')
       F_i ←Replace (F_i)
     EndFor
     Q_{t+1} ← make_new_pop (P_{t+1})
     t ← t + 1
End
```

Figure 3. PHC-NSGA-II basic iteration algorithm.

After assigning DCDs to each solution of front F_i, PHC-NSGA-II sorts F_i on the basis of DCDs in an ascend order and selects the 2M+1 most crowded solutions in the decision space. In other words, the solutions to be replaced are the 2M+1 solutions having the lowest DCD values. PHC-NSGA-II selects the most crowded individuals in the decision space because these individuals can be represented each one by one of its neighbours. So solutions to be replaced are considered to be the less important for the decision maker since each one can be represented by its nearest neighbour in the genotypes space. After updating the parent population P_{t+1}, we apply genetic operators to create the offspring population Q_{t+1}. As we design our PHC-NSGA-II algorithm to tackle continuous optimization problems, we use the Simulated Binary Crossover (SBX) operator [19] and polynomial mutation operator [18] for recombination. These genetic operators are those used in the original version of the NSGA-II algorithm. The SBX operator simulates the binary crossover mechanism to create two offspring solutions by using a specified probability distribution around two selected parent solutions. A higher value of the distribution index η, which controls the children distribution among parents, allows creating near parent solutions and vice versa. The polynomial mutation operator calculates a perturbance factor δ by using a polynomial probability distribution to generate a mutated individual from a parent one.

```
Input:
I: front without DCDs.
V: decision variables number.
Output:
I: front updated with DCDs.
Begin
    N ← |I| // number of individuals of front I
    For i=1 to N do
        I[i].distance ← 0
    EndFor
    For v=1 to V do
        I ← sort (I, D(v), 'ascend')
        I[1].distance ← ∞
        I[N].distance ← ∞
        For i=2 to (N-1) do
            I[i].distance ← I[i].distance + (I [i+1].D(v) - I [i-1].D(v))
// I[i].D(v) represents the v^th decision variable value of individual i in
// the front I.
        EndFor
    EndFor
End
```

Figure 4. Decision crowding distance assignment.

For a better comprehension of the search process introduced in PHC-NSGA-II, we illustrate an example of front before and after performing PHC. According to figure 5, it is interesting to explore the neighbourhood of individual C (pointed with a dashed line circle) in order to find better solutions (e.g. solution G). In order to preserve the genotypes diversity, we do not replace the most crowded individuals in the objective space by the improved solutions. In fact, these ones may be not crowded in the decision space and its elimination could deteriorate the population diversity. The solutions to be replaced are the most crowded elements in decision space (e.g. solution B, designed by triangle) since its deletion does not ruin the population diversity compared to the random elimination of other solutions.

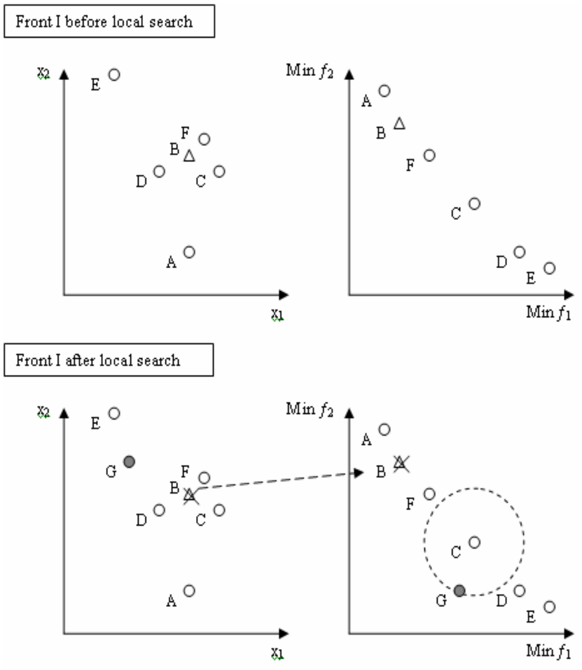

Figure 5. Selection and replacement strategies (case of only one selected solution).

Now, let us explain why we select 2M+1 solutions. In fact, the original NSGA-II assigns infinite crowding distances to solutions having the maximum and/or minimum value(s) on one or more objectives. Thus, the maximum number of solutions having infinite crowding distances is 2M. Selecting the 2M least crowded individuals for PHC allows the algorithm to explore the edges of Pareto region. We select another 1 solution in addition to the 2M solutions in order to assure exploration of the neighbourhood(s) of one or more solution(s) within the Pareto region. In summary, the goals in evolutionary multi-objective optimization are (1) increasing the convergence rate to optimum while (2) preserving a good diversity of solution. These two aims are conflicting and may form a multi-objective optimization problem. Our hybrid algorithm takes into account these goals. First, it tries to improve the convergence by intensifying the search with the

PHC procedure. Second, it tries to preserve population diversity by the CD based selection strategy and the DCD based replacement strategy. This latter can utilise the CD directly, but the use of DCD avoids the case of eliminating solutions that are crowded in the objective space and not crowded in the decision space at the same time.

3. Computational studies

This section is devoted to assess the performance of our hybrid algorithm PHC-NSGA-II on two sets of test problems: 2-dimentional and 3-dimentional test functions. On the first subsection, we compare our algorithm to two other hybrid algorithms that are NSGA-SQP and SPEA-SQP in addition to NSGA-II and SPEA2. On the second subsection that concerns 3-dimentional benchmarks, we make a comparison only between PHC-NSGA-II and NSGA-II due to the data sets availability. It should be noted that parameters values are determined experimentally.

3.1. First set of test problems

In this sub-section, we test the PHC-NSGA-II performance by comparing it to NSGA-II, SPEA2, NSGA-SQP and SPEA-SQP on three bi-objective test functions ZDT1, ZDT2 and ZDT6 [20], which are among the most used in quality measure of multi-objective evolutionary optimizers. The three problems have the same structure:

Min $f_1(x) = x_1$

Min $f_2(x) = g(x)h(f_1(x), g(x))$

ZDT1:

$$f_1(x) = x_1$$

$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$

$$h(f_1, g) = 1 - \sqrt{f_1 / g}$$

where n=30 and all $x_i \in [0,1]$.

The Pareto-optimal front is convex and is formed with $g(x) = 1$. In other words, the Pareto-optimal set corresponds to solutions verifying $0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,...,30$.

ZDT2:

$$f_1(x) = x_1$$

$$g(x) = 1 + \frac{1}{n-1}\sum_{i=2}^{n} x_i$$

$$h(f_1, g) = 1 - \left(f_1 / g \right)^2$$

where n=30 and all $x_i \in [0,1]$.

The Pareto-optimal front is non-convex and is formed with $g(x) = 1$. In other words, the Pareto-optimal set corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,...,30$.

ZDT6:

$$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$$

$$g(x) = 1 + 9\left(\left(\sum_{i=2}^{n} x_i\right) / (n-1) \right)^{0.25}$$

$$h(f_1, g) = 1 - \left(f_1 / g \right)^2$$

where n=10 and all $x_i \in [0,1]$.

The Pareto-optimal front is non-convex and is formed with $g(x) = 1$. Differently speaking, the Pareto-optimal set corresponds to $0 \le x_1^* \le 1$ and $x_i^* = 0$ for $i = 2,3,...,10$. The test function includes two difficulties caused by the non-uniformity of the search space [20]. First, the Pareto-optimal solutions are non-uniformly distributed along the global Pareto front (the front is biased for solutions for which $f_1(x)$ is near one). Second, the solutions density is lower near the Pareto-optimal front and higher away from this front.

Parameters settings of NSGA-II, SPEA2, NSGA-SQP and SPEA-SQP are detailed in [8]. For PHC-NSGA-II algorithm, we use the following parameters values:

- Population size (*pop*) = 50.
- Number of generations (*gen*) = 100 for ZDT1 and 50 for both ZDT2 and ZDT6.
- Neighbourhood size (*ns*) = 12.
- LS iterations number (*N_MAX_ITERATIONS*) = 15.
- Crossover rate = 0.9.
- Mutation rate = 0.1.

In all studies we compare the approximation sets obtained by each algorithm.

Figure 6 and figure 7 illustrates clearly that hybrid MOEAs have better convergence to the Pareto optimal front than the original MOEAs for ZDT1 and ZDT2. This is also true for ZDT6 from figure 8 and the results obtained in [8]. It should be noted that figure 3 illustrates only PHC-NSGA-II, NSGA-SQP and SPEA-SQP non-dominated solutions sets view to data sets availability. It can be observed that solutions of SPEA-SQP represent better diversity than the pure SPEA2 for all problems. And since approximation sets of PHC-NSGA-II are very similar to SPEA-SQP ones in terms of convergence and diversity, it yields that PHC-NSGA-II outperforms SPEA2 for all problems. In addition, it should be noted that ZDT6 does not block PHC-NSGA-II algorithm ability in terms of solutions diversification since it is designed to cause non-uniform density difficulty across the optimal front. From figure 8, it is evident that PHC-NSGA-II solutions have better distribution than SPEA-SQP across the Pareto optimal front.

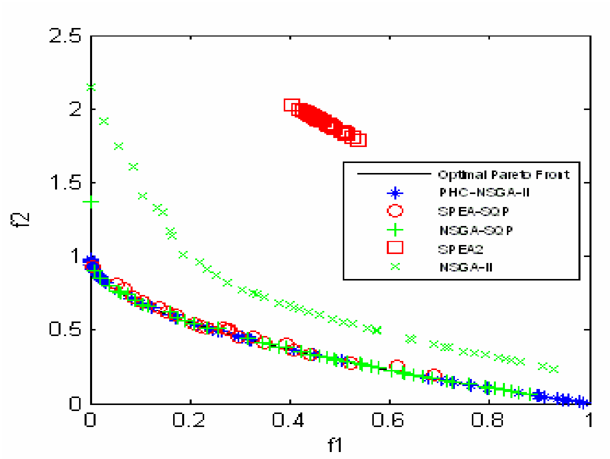

Figure 6. Non-dominated solutions of ZDT1.

To emphasize our claims we choose to use the additive binary epsilon indicator as a performance measure since it is complete and compatible [21]. This metric takes a pair of non-dominated solutions sets X and Y as inputs and returns a pair of numbers as outputs. (I_X, I_Y) such that:

$$I_X = I_{\varepsilon+}(X,Y) = \underset{\varepsilon\in\Re}{Inf}\left\{\forall z^2 \in Y, \exists z^1 \in X : z^1 \preceq_{\varepsilon+} z^2\right\}$$

$$I_Y = I_{\varepsilon+}(Y,X) = \underset{\varepsilon\in\Re}{Inf}\left\{\forall z^2 \in X, \exists z^1 \in Y : z^1 \preceq_{\varepsilon+} z^2\right\}$$

where, for a given $\varepsilon > 0$, $z^1 \preceq_{\varepsilon+} z^2$ if and only if $\forall i \in \{1,\ldots,M\} : z^1{}_i \leq \varepsilon + z^2{}_i$ (M is the number of objectives, $z^1{}_i$ is the ith objective value of solution z^1), assuming minimization (We read z^1 epsilon-dominates z^2). A pair of numbers $(I_X \leq 0, I_Y > 0)$ indicates that X is strictly better than Y [21], while a pair of numbers $(I_X > 0, I_Y > 0)$ means that X and Y are incomparable. Nevertheless, if IX is less than I_Y, then, in a weaker sense, we can say that X is better than Y because the minimum ε value needed to that X ε-dominates Y is smaller than the ε value needed to so that Y ε-dominates X. Table 1 shows different values of the additive binary epsilon indicator for the three problems (assuming the following notation: N = NSGA-II approximation set, S = SPEA2 approximation set, M = PHC-NSGA-II approximation set, NSQP = NSGA-SQP approximation set and SSQP = SPEA-SQP approximation set).

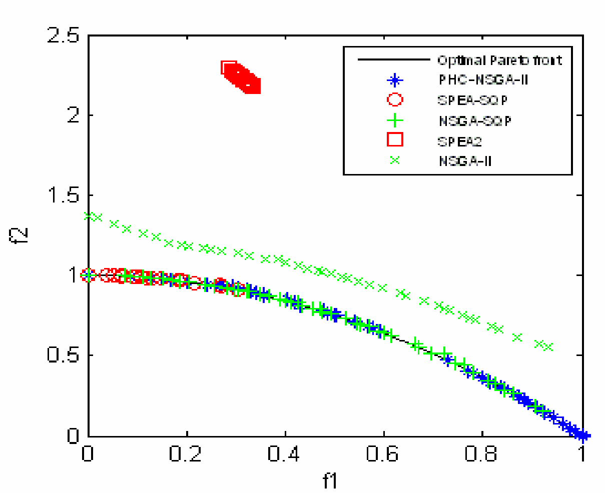

Figure 7. Non-dominated solutions of ZDT2.

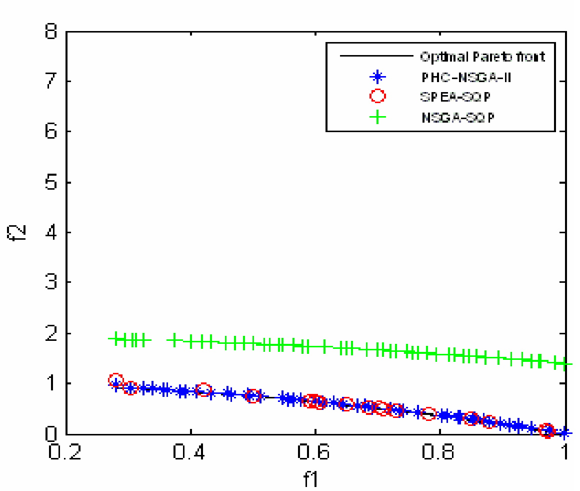

Figure 8. Non-dominated solutions of ZDT6.

From this table, we can state that PHC-NSGA-II outperforms NSGA-II and SPEA2 in all test problems.

Also, PHC-NSGA-II approximation sets are strictly better than NSGA-SQP one on ZDT6 problem and SPEA-SQP one on ZDT2. For positive values of pairs, we remark that on ¾ of pairs, I_M is the least value. Consequently, we can make a weak statement that PHC-NSGA-II outperforms the two other hybrid algorithms.

To make a more accurate comparison between the hybrid algorithms, we use other performance indices [22]: number of solutions obtained by the first technique which dominate solutions obtained by the second technique and number of solutions obtained by the first technique which are dominated by the solutions obtained by the second technique. For two solutions sets *X* and *Y*, which are compared, indices are denoted by *Dominate(X, Y)* and *Dominated(X, Y)*. Table 2 demonstrates obtained results of these indices. It is obvious that PHC-NSGA-II non inferior solutions dominate all NSGA-SQP ones on ZDT6. Besides, there is no solution obtained by SPEA-SQP that dominates any solution from PHC-NSGA-II approximation set on ZDT1 and ZDT2. In addition, no solution from M is dominated by any solution from SSQP on ZDT1 and ZDT2. The superiority of our memetic MOEA may be explained by the performance of our *Pareto Hill Climbing* heuristic in locating promising regions and finding fitter individuals.

3.2. Second set of test problems

This section is devoted to test PHC-NSGA-II ability to tackle tri-objective test problems. We make a comparison between PHC-NSGA-II and NSGA-II metaheuristics on three tri-criterion test functions DTLZ2a, DTLZ4a and DTLZ7a [23] which are modified versions of DTLZ2, DTLZ4 and DTLZ7 [24] for the three dimensional case.

DTLZ 2a and DTLZ4a

The two problems have three objective functions and eight decision variables.

$$\text{Min } f_1 = (1+g)\cos(x_1^{\alpha}\pi/2)\cos(x_2^{\alpha}\pi/2)$$

$$\text{Min } f_2 = (1+g)\cos(\cos(x_1^{\alpha}\pi/2)\sin(x_2^{\alpha}\pi/2)$$

$$\text{Min } f_3 = (1+g)\sin(x_1^{\alpha}\pi/2)$$

$$g = \sum_{i\in\{3,\dots,8\}} (x_i - 0.5)^2$$

$$x_i \in [0,1], i \in \{1,\dots,n\}, n = 8$$

$$\alpha = \begin{cases} 1, \text{if function is DTLZ2a} \\ 100, \text{if function is DTLZ4a} \end{cases}$$

The Pareto front is one eighth of a sphere of radius 1, centred on (0, 0, 0). The Pareto optimal set consists of all solutions where all but the first decision variables are equal to 0.5, and the first decision variable may take any value in [0, 1]. The effect of setting α = 100 is to severely bias the density distributions of the solutions towards the $f_3 - f_1$ and $f_2 - f_1$ planes.

DTLZ7a

$$\text{Min } f_1 = x_1$$

$$\text{Min } f_2 = x_2$$

$$\text{Min } f_3 = (1+g)h$$

$$g = 1 + 9/6 \sum_{i\in\{3,\dots,8\}} x_i$$

$$h = 3 - \sum_{i\in\{1,2\}} \left[\frac{f_i}{1+g}(1+\sin(3\pi f_i)) \right]$$

$$x_i \in [0,1], i \in [1,\dots,n], n = 8$$

The optimal Pareto region has four disconnected surfaces in objective space.

Parameters settings of the two algorithms are summarized in table 3. The parameters values are organised as follows: ***(pop, gen, ns, N_MAX_ITEARATIONS) / maximum number of functions evaluations***. It should be noted that crossover and mutation rates are set to 0.9 and 0.1 respectively for the two algorithms.

It is obvious from different plots (Figure 9, figure 10 and figure 11) that PHC-NSGA-II non-dominated solutions are nearer to the optimal Pareto region than NSGA-II ones on all the three problems. As mentioned above, DTLZ7a problem has four disconnected region in criterion space. Figure 11 shows the ability of our algorithm to find solutions in the four regions while NSGA-II found only solutions in three regions. Besides, non-inferior solutions of PHC-NSGA-II are closer to Pareto surfaces than NSGA-II solutions which proves the contribution and the positive effect of incorporating local search within the evolutionary search process. Table 4 presents the additive binary epsilon indicator values for DTLZ problems. With respect to these values, we can only say in a weak sense, that PHC-NSGA-II approximation set is better than NSGA-II one because the minimum ε value needed to so that PHC-NSGA-II non-dominated solutions set ε-dominates NSGA-II one is smaller than the ε value needed to so that NSGA-II approximation set ε-dominates PHC-NSGA-II one.

To more confronting PHC-NSGA-II and NSGA-II algorithm, we illustrate in table 5 results of the

Dominate and *Dominated* indices. We notice, from this table, that the number of solutions of NSGA-II approximation set dominated by PHC-NSGA-II set is so greater than number of solutions of NSGA-II non-dominated solutions set dominated by NSGA-II set. In addition, the number of solutions obtained by PHC-NSGA-II dominating solutions from NSGA-II approximation set is greater than the number of solutions obtained by NSGA-II dominating solutions from PHC-NSGA-II approximation set on all problems especially for DTLZ4a and DTLZ7a. Differently speaking, number of solutions from N that are dominated by solutions from M represents more than half of the population for both problems: DTLZ4a and DTLZ7a.

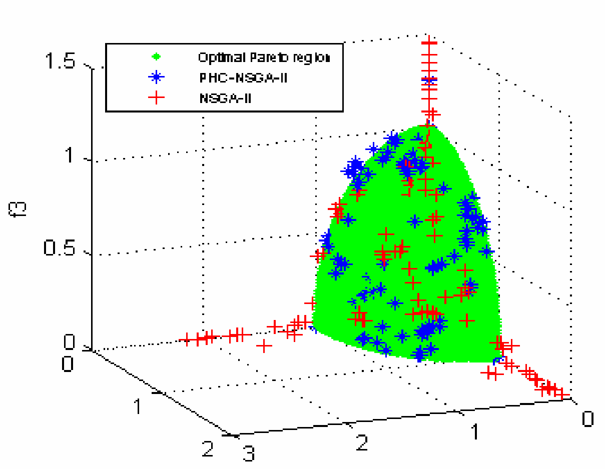

Figure 9. Non-dominated solutions of DTLZ2a.

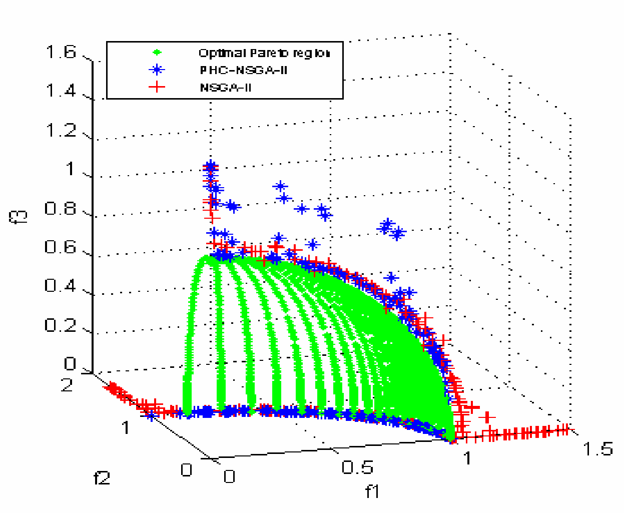

Figure 10. Non-dominated solutions of DTLZ4a.

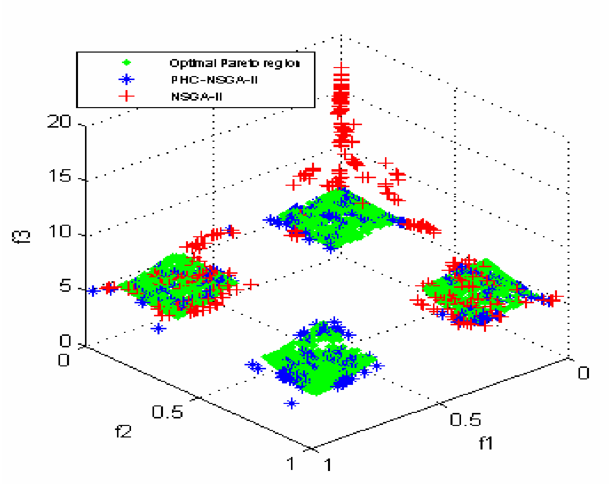

Figure 11. Non-dominated solutions of DTLZ7a.

4. Conclusions

In this study, a new local search method, named Pareto Hill Climbing, was designed and incorporated into the NSGA-II evolutionary algorithm. New strategies for local search selection and individuals' replacement have been presented. A new multi-objective memetic algorithm was introduced. Some conclusions could be summarized as follows. First, if there are local traps, the hybrid algorithm is more able than the original MOEA to identify the true optimal Pareto region. Second, the hybridization method does not ruin solutions diversity and for some problems it can enhance the extent of the Pareto region. Third, PHC-NSGA-II is competitive with NSGA-SQP and SPEA-SQP which are gradient based hybrid schemes. In addition, our algorithm can tackle non differentiable problems where gradient based algorithm can not be applied. Hence, our new MMA could be easily extended to solve discrete and/or constrained multi-objective optimization problems. Our future research will focus on (1) applying our hybrid algorithm to solve real world problems such as the multi-location transshipment problem [28] and (2) incorporating an adaptive parameters tuning into it.

References

[1] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley and Sons, New York, 1986.

[2] C.A. Coello, ‘‘An updated survey of GA-based multiobjective optimization techniques’’, Technical

report, Laboratorio Nacional de Informatica Avanzada (LANIA), Xalapa, Veracruz, Mexico, 1998. Technical Report Lania-RD-98-08.

[3] H. H. Hoos, and T. Stützle, *Stochastic local search: Foundations and Applications*, Morgan Kaufmann Publishers, 2005.

[4] P. Moscato, ‘‘On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms’’, Caltech Concurrent Computation Program, C3P Report 826, (1989).

[5] P. Merz, “Memetic algorithms for combinatorial optimization problems: fitness landscapes and effective search strategies’’, Ph. D. Thesis, department of Electrical Engineering and Computer Science, University of Siegen, Germany 2000.

[6] D. Sharma, A. Kumar, K. Deb, and K. Sindhya, ‘‘Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems’’, *Special Session & Competition on Performance Assessment of Multi-Objective Optimization Algorithms*, *CEC-07*, Singapore, 25-28 September 2007.

[7] A. Kumar, D. Sharma, K. Deb, ‘‘A hybrid multi-Objective optimization procedure using PCX based NSGA-II and sequential quadratic programming’’, *Special Session & Competition on Performance Assessment of Multi-Objective Optimization Algorithms*, *CEC-07*, Singapore, 25-28, September 2007.

[8] X. Hu, Z.Huang, and Z. Wang, ‘‘Hybridization of the multiobjective evolutionary algorithms and the gradient-based algorithms’’, *Evolutionary Computation*, 2003, vol. 2, pp. 870- 877.

[9] A. Jaszkiewicz, ‘‘Genetic local search for multiple objective combinatorial optimization’’, Technical Report RA-014/98, Institute of Computing Science, Poznan University of Technology, 1998.

[10] A. Jaszkiewicz, ‘‘Do multiple-objective metaheuristics deliver on their promises? a computational experiment on the set covering problem’’, *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp.133-143, April 2003.

[11] H. Ishibuchi, and T. Murata, ‘‘Multi-Objective Genetic Local Search Algorithm’’, *In Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 119–124, Nagoya, Japan, 1996.

[12] J. Knowles, and D. Corne, ‘‘M-PAES: A memetic algorithm for multiobjective optimization’’, *In 2000 Congress on Evolutionary Computation*, vol. 1, pp. 325–332, Piscataway, New Jersey, July 2000.

[13] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby, ‘‘Hybrid genetic algorithms: A review’’, *Engineering Letters*, vol. 3, no. 2, pp. 124-137, 2006.

[14] J. Knowles, and D. Corne, ‘‘Memetic algorithms for multiobjective optimization: issues, methods and prospects’’, *In Recent Advances in Memetic Algorithms*, pp. 313–352, Springer-Verlag, 2004.

[15] K. Deb, S. Argawal, A. Pratap, and T. Meyarivan, ‘‘A fast and elitist multi-objective genetic algorithm: NSGA-II’’, *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.

[16] N. Srinivas, and K. Deb, ‘‘Multiobjective optimization using non-dominated sorting in genetic algorithms’’, *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, fall 1994.

[17] K. Deb, *Multi-objective optimization using evolutionary algorithms*, John Wiley and Sons, Ltd, 2002, pp. 245-253.

[18] M. M. Raghuwanshi, and O. G. Kakde, ‘‘Survey on multiobjective evolutionary and real coded genetic algorithms’’, *In Proceeding of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, vol. 11, pp. 150-161, 2004.

[19] K. Deb, and R. B. Argawal, ‘‘Simulated binary crossover for continuous search space’’, *Complex Systems*, vol. 9, no. 2, pp. 115-148, April 1995.

[20] E. Zitzler, K. Deb, and L. Thiele, ‘‘Comparison of multiobjective evolutionary algorithms: Empirical results’’, *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[21] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca, ‘‘Performance assessment of multiobjective optimizers: An analysis and review’’, *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[22] C. Grosan, and A. Abraham, ‘‘Hybrid line search for multiobjective optimization’’, *In Proceedings of the Third International Conference on High Performance Computing and Communications (HPCC 2007)*, vol. LNCS 4782, Springer-Verlag, Germany, 2007.

[23] J. Knowles, ‘‘ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems’’, *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50-66, Feb 2006.

[24] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, ‘‘Scalable multi-objective optimization test problems’’, *Evolutionary Computation*, 2002, vol. 2, pp. 825-830.

[25] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm”, Comput. Eng. Networks Lab. (TIK), Swiss Fed. Inst. Technol. (ETH), Zurich, Switzerland, Tech. Rep. 103, May 2001.

[26] R. Chelouah, and P. Siarry, ‘‘Tabu search applied to global optimization’’, *European Journal of Operational Research*, vol. 123, no. 2, pp.256-270, 2000.

[27] Y.S. The, and G.P. Rangaiah, ‘‘Tabu search for global optimization of continuous functions with application to phase equilibrium calculations’’, *Computers & Chemical Engineering*, vol. 27, no. 11, pp. 1665-1679, 2003.

[28] N. Belgasmi, L. Ben Said, and K. Ghédira, ‘‘Evolutionary multiobjective optimization of the multi-location transhipment problem’’, *Operational Research*, vol. 8, no. 2, Springer-Verlag, 2008.

Table 1
Additive binary epsilon indicator values for ZDT1, ZDT2 and ZDT6.

	ZDT1	ZDT2	ZDT6
(I(M, N), I(N, M))	(-0.0028, 0.2307)	(-2.1783e-004, 0.5506)	▨
(I(M, S), I(S, M))	(-0.4032, 1.7789)	(-0.0104, 0.9081)	▨
(I(M, NSQP), I(NSQP, M))	(0.0612, 0.0549)	(0.0653, 0.1495)	(-8.4952e-006, 1.3936)
(I(M, SSQP), I(SSQP, M))	(0.0345, 0.1926)	(-0.2855, 2.1801)	(0.0215, 0.0656)

Table 2
Dominate and Dominated indicators values for ZDT1, ZDT2 and ZDT6 (only for PHC-NSGA-II, NSGA-SQP and SPEA-SQP).

	ZDT1	ZDT2	ZDT6
Dominate (M, NSQP) / Dominated (M, NSQP)	4/1	4/11	50/0
Dominate (NSQP, M) / Dominated (NSQP, M)	1/3	11/5	0/50
Dominate (M,SSQP) / Dominated (M,SSQP)	12/0	13/0	8/5
Dominate (SSQP, M) / Dominated (SSQP, M)	0/13	0/39	5/4

Table 3
Parameters settings for DTLZ test problems.

	DTLZ2a	DTLZ4a	DTLZ7a
PHC-NSGA-II	(100, 50, 5, 10) / 26200	(200, 40, 5, 5) / 25912	(300, 50, 5, 10) / 55200
NSGA-II	(100, 262) / 26200	(200, 130) / 25912	(300, 200) / 55200

Table 4
Additive binary epsilon indicator values for DTLZ2a, DTLZ4a AND DTLZ 7a.

	DTLZ2a	DTLZ4a	DTLZ7a
(I(M, N), I(N,M))	(0.1018, 0.2426)	(0.0288, 0.0580)	(0.0412, 1.1791)

Table 5
Dominate and Dominated indicators values for DTLZ2a, DTLZ4a AND DTLZ7a.

	DTLZ2a	DTLZ4a	ZDT7a
Dominate (M, N) / Dominated (M, N)	10/8	57/37	113/2
Dominate (N, M) / Dominated (N, M)	8/46	41/124	41/163